

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ciencias y Humanidades

TECLADO ESPECIAL PARA MINUSVÁLIDOS

JOSÉ FRANCISCO ARÉVALO LIÉVANO

Guatemala

2001

TECLADO ESPECIAL PARA MINUSVÁLIDOS

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ciencias y Humanidades


TECLADO ESPECIAL PARA MINUSVÁLIDOS

Trabajo de investigación presentado por José Francisco
Arevalo Liévano para optar al grado académico de
Licenciatura en Ingeniería Electrónica

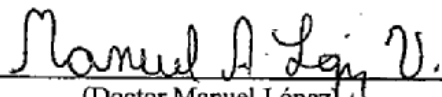
Guatemala

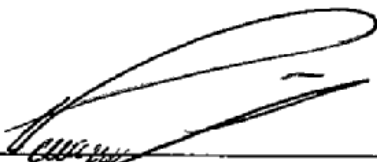
2001

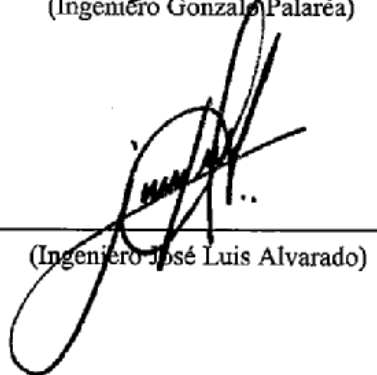
Vo. Bo. :

(f) 
(Ingeniero Gonzalo Palaréa)

Tribunal Examinador

(f) 
(Doctor Manuel López)

(f) 
(Ingeniero Gonzalo Palaréa)

(f) 
(Ingeniero José Luis Alvarado)

Fecha de aprobación: Guatemala 1 de octubre de 2001

A mi Madre y
a todas las personas que
colaboraron con este trabajo

CONTENIDO

	Páginas
Lista de tablas	vii
Lista de figuras	viii
Resumen	ix
I. INTRODUCCIÓN	1
II. OBJETIVOS	2
III. ANTECEDENTES	4
IV. MÉTODO DE TRABAJO	6
A. Marco teórico	7
1. Bases del funcionamiento de un teclado convencional	7
2. El microcontrolador PIC 16F84	19
3. Módulo de pantalla de cristal líquido	22
B. Marco experimental	27
1. Verificación de la trama de datos y prototipos	27
2. Diseño del circuito	31
3. Diseño del programa	34
V. CONCLUSIONES	43
VI. RECOMEDACIONES	44
VII. BIBLIOGRAFÍA	45
APÉNDICES	
A. Diagrama del teclado especial	46
B. Flip-Flops en cascada	47
C. Programa	48
D. Manual del usuario	69
E. Diagrama de tiempo controlador de pantalla de cristal líquido	78

LISTA DE TABLAS

Tabla	Página
1. Códigos para teclados AT	11
2. Códigos para teclas XT	12
3. Códigos de espera	16
4. Códigos de velocidad de repetición	17
5. Códigos indicadores del teclado	18
6. Descripción de pines del 16F84	20
7. Descripción de pines del módulo de pantalla de cristal líquido	22
8. Uso de los pines del controlador	33
9. Variables en el programa	42

LISTA DE ILUSTRACIONES

Figuras	Páginas
1. Conectores DIN 180 y mini din, PS2	14
2. Adaptadores DIN 180 a PS2	14
3. Circuito de alimentación del módulo de pantalla de cristal líquido	24

RESUMEN

En este trabajo se expone el desarrollo de un proyecto para proporcionar a personas discapacitadas un Teclado Especial en que con solamente tres botones de gran tamaño pueda escribir en una computadora personal, se investigó si había algo similar y lo único que se encontró fue programas de ayuda y no así un teclado como el que se expone, además es posible hacer modificaciones en los tres botones o cambiarlos por sensores adaptando a la necesidad del discapacitado, el proyecto se diseño para que pueda ser utilizado en cualquier computadora de escritorio o portátil no importando así el tipo de teclado que tenga (conector DIN 180, PS2, USB) . El teclado también puede ser utilizado con cualquier programa de computadora sin la necesidad de instalar manejadores (drivers) adicionales a los que ya posee para el teclado convencional.

I. INTRODUCCIÓN

Observaciones detenidas sobre las dificultades que algunas personas discapacitadas tienen en el manejo de una computadora personal, especialmente las que sufren por problemas de motricidad en sus manos, motivaron este estudio para ofrecerles alguna solución que al mismo tiempo fuera sencilla en su construcción y manejo, tres botones grandes y de fácil acceso; además, que fuera de bajo costo y se prestara para la capacitación del que la opera.

Una revisión de las distintas soluciones que pueden ser útiles en la actualidad demostró que aunque existen varios programas de software estos no ofrecen una solución física (hardware) para estas personas.

Ante esto se diseñó un teclado especial de uso sencillo en el que, con solamente tres botones grandes, se pueda seleccionar cualquier letra, número o símbolo especial que aparece en un teclado común.

Este teclado, a la vez que desarrolla sus habilidades motrices, ayuda al niño a través de programas a aprender progresivamente las letras del alfabeto, formar palabras, hasta llegar a escribir como cualquier persona con la única limitante de la velocidad con que podrá ir seleccionando las letras.

La base del teclado es un microcontrolador PIC de la marca “Microchip”; estos controladores son cada vez más comunes en el mercado como dispositivos de bajo costo, con su propio procesador y periféricos encapsulados en el mismo chip.

Dentro de este trabajo está incluido un pequeño manual (apéndice D) con el objetivo que las personas que harán uso del teclado puedan tener acceso con facilidad sin que tener que comprender aspectos técnicos. Además un apéndice donde está el programa principal del microcontrolador (apéndice C) para que cualquier persona interesada pueda revisar cada una de las rutinas.

También se incluyen dentro del mismo los diagramas de los circuitos utilizados y el circuito del teclado propiamente (apéndices B y A respectivamente).

II. OBJETIVOS

A. Objetivo general

El objetivo de este trabajo es desarrollar e implementar un teclado funcional formado por una pantalla de corrimiento y tres botones que permitan seleccionar cualquier letra, número o símbolo especial tal como aparecen en un teclado común de una computadora personal.

Los botones están diseñados para que la persona discapacitada o con problemas de coordinación motriz, pueda presionarlas con facilidad de la siguiente forma: 1) Botón de la derecha desplaza los caracteres de izquierda a derecha según el orden de un teclado convencional, 2) Botón de la izquierda los desplaza en sentido inverso y 3) con el botón del centro el usuario enviará el carácter seleccionado a la unidad central de procesamiento de la computadora personal.

El teclado de 3 botones deberá permitir al usuario seleccionar su nivel de complejidad y velocidad según su capacidad inicial.

B. Objetivos específicos

1. Desarrollar un teclado que pueda ser conectado en paralelo con un teclado convencional en cualquier computadora personal (PC) y que pueda ser utilizado en cualquier aplicación de software.
2. Concluir con el desarrollo de un teclado de tres botones grandes que, además de su tamaño, los mismos se presten para ser operados con facilidad por la persona discapacitada.
3. Implementar una pantalla de cristal líquido donde se desplieguen de una en una y en secuencia las letras, números y símbolos de un teclado normal.
4. Diseñar un programa del microcontrolador que permita la visualización y el control del despliegue, al igual que el envío de letras y símbolos.
5. Establecer dos diferentes modalidades del uso del teclado: el primero en el que se desplieguen únicamente letras y números (modo simplificado) y el segundo todos los símbolos que posee un teclado convencional (modo completo).

6. Permitir al usuario establecer diferentes velocidades de despliegue de las letras en la pantalla de cristal líquido una vez conectado el teclado.
7. Permitir al usuario de una forma versátil el cambio de velocidad establecido en un inicio hacia delante y hacia atrás una vez iniciado el despliegue de los caracteres.

III. ANTECEDENTES

Revisión realizada:

Microsoft ha desarrollado algunas facilidades que permiten el acceso de personas discapacitadas a sus productos y servicios tecnológicos. A continuación se presentan algunas facilidades que Windows ofrece siendo el sistema operativo más utilizado.

Windows incluye numerosas características para facilitar el acceso a usuarios con discapacidades de movimiento o de audición. Estas características permiten cambiar las funciones de presentación del ratón y del teclado así como utilizar sonidos para ayudar a los usuarios a trabajar con Windows de manera más eficaz. Por ejemplo:

- “MOUSE KEYS”: si tiene dificultades para utilizar el ratón podrá recurrir a esta característica que permite emplear el teclado numérico para mover el puntero del ratón.
- “STICKY KEYS”: permite habilitar las funciones de las teclas CTRL, ALT o SHIFT presionándolas y teniéndolas activas hasta la siguiente vez que se presione otra tecla que no sea CTRL, ALT o SHIFT. Esto es útil para personas que tengan la dificultad para presionar dos teclas simultáneamente.
- “FILTER KEYS”: al activar esta opción se le dan instrucciones al teclado de ignorar teclas que han sido presionadas accidental o repetidamente. También se puede ajustar la velocidad de repetición de una tecla cuando se le deja presionada.
- “TOGGLE KEYS”: al activar esta opción se le da instrucciones a la computadora para que emita un sonido agudo cuando se presiona las teclas “CAPS LOCK”, “SCROLL LOCK” o “NUM LOCK”; y para que emita un sonido grave cuando se presiona dichas teclas para desactivarlas.
- “SOUND ENTRY”: esta opción da instrucciones para que parpadee una parte de la computadora cada vez que la bocina del sistema emita algún sonido. Es posible especificar qué parte de la pantalla parpadee.

- “SHOW SOUNDS”: esta opción le da orden a los programas que muestren informaciones por sonido, para que también lo hagan mediante información visual, como por ejemplo: desplegar letreros o íconos informativos.
- “HIGH CONTRAST”: esta opción le da instrucciones a los programas para cambiar la gama de colores para una mejor legibilidad cuando sea posible.

Algunos programas despliegan en la pantalla ayudas extras de cómo usar el teclado si se activa la función “Mostrar ayuda extra en programas”.

Estas son algunas de las formas de acceso que ofrecen los programas de uso normal.

Sobre la base de las investigaciones realizadas se concluyó que existen varios programas (software) que ofrecen ayuda a discapacitados, sin embargo son limitados ya que la persona debe seguir utilizando los dispositivos convencionales de la computadora tales como: ratón, teclado, pantallas sensibles al tacto etc. Actualmente no existe un dispositivo físico (hardware) que ofrezca una verdadera ayuda al discapacitado, que pueda ser utilizado en cualquier computadora y en cualquier programa.

IV. MÉTODO DE TRABAJO

La investigación del funcionamiento de un teclado convencional, la experimentación del funcionamiento del teclado mediante el diseño de prototipos, para luego diseñar e implementar un teclado especial definitivo que cumpliera las especificaciones requeridas.

Se presentan los siguientes títulos y el desglose de los mismos:

A. Marco teórico:

1. Bases del funcionamiento de un teclado convencional.
2. Bases del funcionamiento de un microcontrolador.
3. Bases del funcionamiento de un módulo de pantalla de cristal líquido.

B. Marco experimental:

1. Verificación de trama de datos y prototipos.
2. Diseño del circuito.
3. Diseño del programa.

C. Materiales empleados:

1. Microcontrolador PIC
2. Base de un teclado en mal estado como soporte, distintas partes de juguetes de plástico y otros desechos.
3. Dispositivos electrónicos de uso común.

A. Marco teórico

1. Bases de funcionamiento de un teclado convencional. El teclado es una unidad independiente en un sistema de una computadora personal (PC) y posee su propio microprocesador y memoria. El procesador informa al sistema cuando una tecla es presionada o levantada. Si el usuario presiona una tecla en el teclado se genera un impulso eléctrico que define la posición de la misma. Esta señal la maneja el microprocesador del teclado, el cual se encuentra de dentro del mismo. Generalmente ese microprocesador es un Intel 8048 o un equivalente. En computadoras de clase AT la comunicación la maneja un Intel 8042. Con este microprocesador las computadoras de clase AT son capaces de una comunicación bidireccional entre el teclado y el CPU. Las primeras computadoras no tienen la capacidad de una comunicación bidireccional.

El procesador del teclado convierte los impulsos eléctricos indicando la posición de la tecla a un código llamado en inglés "SCAN CODE". No hay relación entre este código y el carácter impreso en la tecla que fue presionada o la función que la tecla representa en el programa en uso.

El procesador del teclado envía el código a la computadora. En una computadora AT, el controlador de teclado acepta el código. Esta transferencia es serial dado que el cable que conecta el teclado y la computadora tiene solamente una línea de datos. Esta comunicación es sincrónica, en contraposición con la comunicación asincrónica en el puerto serial de la PC.

La comunicación sincrónica se lleva a cabo mediante una línea de datos y una de reloj. La línea de reloj transmite una señal temporizada cambiando continuamente de alto a bajo (1 a 0). La transmisión de los bits individuales de los códigos son sincronizados por este pulso.

Si varias teclas se presionan simultáneamente, el procesador de teclado lo guarda en una pila interna. Esta usualmente tiene suficiente espacio para presionar diez teclas. El usuario no debe preocuparse que esta pila se llene porque los datos son transmitidos al CPU mucho más rápido que lo que pueda escribir.

Se generan códigos de teclado ("SCAN CODES") cuando el usuario suelta una tecla. Esto le informa al sistema cuando una tecla sigue presionada o ha sido liberada. Lo cual es muy importante por ser la única forma en que la computadora puede interpretar correctamente la situación cuando más de una tecla ha sido presionada. Sin esta capacidad el usuario no podría ejecutar ciertas tareas como letras mayúsculas o reiniciar la computadora con CTRL + ALT +DEL.

El sistema utiliza códigos de tecla presionada y tecla liberada. En sistemas XT la única diferencia entre ambos es el séptimo bit del código de la tecla liberada. Es de notar dos importantes consecuencias: a) Los códigos de tecla liberada son siempre mayores de 128 y los códigos de tecla presionada son menores que 128, considerando que los bits son enviados del bit menos significativo al más significativo. b) un teclado de computadora no puede tener más de 128 teclas, de otra forma los códigos de tecla presionada y liberada se traslaparían.

El ejemplo más obvio en el que más de una tecla ha sido presionada simultáneamente es el procedimiento para escribir una mayúscula. Por ejemplo, para escribir “A”, el usuario presiona y deja presionada la tecla SHIFT, después presiona la tecla A. El procesador de teclado envía el código de tecla presionada para SHIFT (36H)¹, después el código de tecla presionada de la tecla A (1EH)². Como el sistema no ha recibido un código para tecla liberada de SHIFT aún, él reconoce que dos teclas están siendo presionadas simultáneamente y genera el código de mayúscula en vez de minúscula.

El controlador del teclado (dentro del CPU) recibe estos códigos mediante la interrupción IRQ1, la cual se ejecuta cada vez que el teclado envía un código de tecla presionada o liberada. Esta entrada llama a la interrupción 09H. Esta rutina controladora de teclado recibe los códigos de tecla presionada y liberada y los convierte a sus correspondientes caracteres ASCII, que pueden ser leídos en la aplicación de uso.

Como se ha visto en el ejemplo de la letra mayúscula “A”, no toda tecla presionada se convierte en caracteres visibles en la pantalla. El controlador de teclado genera caracteres solamente después de recibir el código de tecla presionada “A”. Un carácter no se genera cuando se recibe el código de tecla presionada de SHIFT. Cuando se introduce un código ASCII por medio de la tecla ALT y el teclado numérico, es necesario que varias teclas se presionen antes que un carácter aparezca en la pantalla.

Así, los códigos del teclado son convertidos en códigos ASCII, los cuales son estándar en todas las computadoras. Aún cuando el conjunto de códigos ASCII consiste en 128 caracteres, las computadoras personales utilizan un conjunto de códigos ASCII extendidos, el cual contiene 256 caracteres.

La conversión a caracteres ASCII no pasa directo a la aplicación, primero se guarda en una pila, y el trabajo del controlador de teclado ha finalizado. La aplicación podrá leer después los caracteres de la pila

¹ Ver tabla 4.2

² Ver tabla 4.2

del controlador de teclado y procesarlo. La interrupción 16H en el "ROM-BIOS" tiene diferentes funciones disponibles para este propósito. DOS puede reemplazar este controlador con otro programa. El controlador de teclado en el ROM-BIOS se configura como teclado norteamericano normalmente. Para usar caracteres en otros idiomas, es posible instalar otro controlador de teclado en el archivo AUTOEXEC. BAT; lo que probará que, a pesar de los caracteres impresos en el teclado, el programa convierte los códigos de teclado en ASCII, lo que determinará cuáles caracteres aparecerán en la pantalla.

La pila está configurada en anillo. Esta clase de pila se utiliza cuando los caracteres son escritos dentro de ella y leídos del mismo asincrónicamente, es decir sin un espaciamiento de tiempo específico. Así la pila en anillo utiliza dos punteros: un puntero indica la posición del nuevo carácter a ser leído y el otro indica la posición donde el nuevo carácter será escrito.

Con este método, los punteros son movidos y no todo el contenido de la pila. Inicialmente, los dos punteros apuntan al principio de la misma. Éstos se mueven hacia el final de la pila con cada lectura y escritura; cuando los punteros alcanzan el fin de la pila, son reiniciados al principio de esta. El nombre de pila en anillo viene del movimiento circular de los punteros.

Suponiendo que el puntero de escritura se encuentra en una posición específica y el usuario presiona otra tecla, esta será guardada en esa posición. El puntero después se mueve dos bytes hacia el fin de la pila. Cada carácter en la pila del teclado requiere dos bytes: uno para el código ASCII y el otro para el código del teclado ("Scan Code"). Si el código de la tecla se guardó en la última palabra, el puntero se reinicia al principio de la pila.

De la misma forma una vez el carácter es leído, el puntero de lectura se mueve dos bytes hacia el fin de la pila; si la última casilla ha sido leída, el puntero de lectura se reinicia al principio de la pila.

La relación entre los dos punteros indica el estado de la pila. Por eso dos condiciones son importantes:

- Si los dos punteros tienen el mismo valor, indican que la pila de teclado está vacío.
- Si el puntero del final trata de ocupar el mismo espacio que es puntero de inicio, esto indica que la pila del teclado está llena.

La pila del BIOS del teclado puede manejar 32 bytes. Como cada carácter requiere de dos bytes, la pila puede manejar 16 caracteres.

a. **Teclados de computadoras personales:** Varios tipos de teclados están disponibles para las computadoras personales (PC), pero realmente hay tres tipos estándar. Estos teclados fueron originalmente introducidos por IBM; y a pesar de ser estándar su presentación puede variar dependiendo de los lenguajes, así las teclas pueden localizarse en diferentes lugares y contener diferentes símbolos. A pesar de lo anterior, el número de teclas y los códigos que produce son estándar.

b. **Teclados PC/XT y AT:** Las computadoras personales fueron introducidas al mercado con un teclado PC/XT, el cual tenía 83 teclas. El diseño de este teclado era con las teclas de ENTER y SHIFT pequeñas y difíciles de usar.

El teclado AT, con 84 teclas, resolvió dichos problemas. Las teclas de ENTER Y SHIFT fueron mayores, lo que hace fácil su localización y escritura rápidamente. Sin embargo, el teclado en sí no fue mayor, algunas otras teclas tuvieron que ser menores. Las teclas de NUM LOCK, SCROLL LOCK y + en el teclado numérico fueron hechas de menor tamaño, dado que no se usan tan frecuente como las teclas de SHIFT.

El teclado MF-II, después del MF-I, fue desarrollado para PC y XT, sin embargo no fue muy popular. Algunas de las características del MF-II se convirtieron en estándares de teclado:

- El grupo dedicado a teclas de cursores está separado de teclado numérico
- Teclas de funciones en la parte de arriba del teclado
- Teclas de funciones F1 y F2
- Teclas de ALT
- Tres LEDs indican el estado de las teclas NUM LOCK, CAPS LOCK Y SCROLL LOCK

Existen dos versiones del teclado MF-II. a) la versión estadounidense con 101 teclas y b) la europea con 102; esta última permite que una tecla con una letra adicional pueda ser agregada después de la tecla SHIFT izquierda.

Programas como, manejadores de teclado en particular, pueden reconocer el teclado MF-II. Cuando son revisados los teclados MF-II responden con su código de identificación correspondiente. Los otros dos teclados no tienen esta capacidad.

Con la evolución de las computadoras portátiles, se han diseñado varios tipos de teclados. Estos teclados utilizan usualmente sistemas más complicados simulando alguno de los tres estándares de teclados para computadoras personales previamente descritos.

c. **Códigos del teclado:** Aún cuando el conjunto de caracteres ASCII son estándar, los códigos varían dependiendo del teclado. Los tres tipos de teclado estándar descritos con anterioridad trabajan con diferentes conjuntos de códigos de teclado. Las siguientes tablas muestran la lista de códigos para teclados de tipo PC/XT y AT.

Tabla 1
Códigos para teclados AT.

Carácter	Código	Carácter	Código
F1	05]	5B
F2	06	a	1C
F3	04	s	1B
F4	0C	d	23
F5	03	f	2B
F6	0B	g	34
F7	83	h	33
F8	0A	j	3B
F9	01	k	42
F10	09	l	4B
F11	78	;	4C
F12	07	'	52
ESC	76	ENTER	5A
1	16	SHIFT	12
2	1E	z	1A
3	26	x	22
4	25	c	21
5	2E	v	2A
6	36	b	32
7	3D	n	31
8	46	m	3A
9	45	,	41
0	4E	.	49
-	55	/	4A
=	5D	CLTR	14
BACKSPACE	66	SPC	29
TAB	0D	ALT	11

Continuación Tabla 1

Carácter	Código	Carácter	Código
q	15	Home	6C
w	1D	PgUp	7D
e	24	End	69
r	2D	PgDw	7A
t	2C	Ins	70
y	35	Del	71
u	3C	Up	75
i	43	LEFT	6B
o	44	Dn	72
p	4D	RIGHT	74
[54		

Tabla 2

Códigos para teclados XT

Carácter	Código	Carácter	Código
F1	3B	a	1E
F2	3C	s	1F
F3	3D	d	20
F4	3E	f	21
F5	3F	g	22
F6	40	h	23
F7	41	j	24
F8	42	k	25
F9	43	l	26
F10	44	;	27
ESC	01	'	28
1	02	ENTER	1C
2	03	SHIFT	36
3	04	z	2C
4	05	x	2D
5	06	c	2E
6	07	v	2F
7	08	b	30
8	09	n	31
9	0A	m	32
0	0B	,	33
-	0C	.	34
=	0D	/	35
BACKSPACE	0E	CLTR	1D
TAB	0F	SPC	39
Q	10	ALT	38
W	11	Home	47

Continuación Tabla 2

Carácter	Código	Carácter	Código
E	12	PgUp	49
R	13	End	4F
T	14	PgDw	51
Y	15	Ins	52
U	16	Del	53
I	17	Up	48
O	18	LEFT	4B
P	19	Dn	50
[1A	RIGHT	4D
]	1B		

En los teclados XT se utilizan dos códigos: el primero³ correspondiente a la tecla presionada y el segundo de tecla liberada, la diferencia ente estos códigos es únicamente el bit más significativo el cual se pone a 1 cuando se libera la tecla. Así estos últimos son derivados de los primeros sumando 80 h.

Para teclados AT se utilizan tres códigos: el primero⁴ Corresponde al código de tecla presionada y para tecla liberada se utilizan dos más F0 y luego se vuelve a mandar el código correspondiente al carácter o tecla presionada.

d. El procesador del teclado: Como se ha mencionado el teclado es una unidad independiente del sistema de una computadora personal por tener su propio microprocesador y memoria.

El procesador informa al sistema cuando una tecla se presiona o se libera. Esto lo hace por medio de códigos especiales (“scan codes”) indicando al sistema la posición de la tecla que se presiona o libera y no tiene relación con los códigos ASCII, hasta que el sistema los convierte.

El teclado se conecta al sistema por medio de un cable de cinco líneas siendo las siguientes:

- Línea de reloj (pin #1).
- Línea de datos (pin #2).
- Línea de iniciación (reset) en el pin # 3
- Línea de tierra (pin #4)

³ Ver tabla 4.2

- Línea de 5 voltios (pin # 5).

Existen también dos tipos de conectores para teclados:

Figura 1



DIN 180



PS2

El primero (izquierda) es un conector de tipo DIN 180 de 5 pines, y el segundo a MINI DIN o más conocido como PS2 de 6 pines, este último es totalmente compatible con el primero, ya que el 6 pin no se utiliza, existen adaptadores de un tipo a conector al otro como los que se muestran a continuación:

Figura 2



Adaptadores DIN 180 a PS2

La comunicación con el sistema se efectúa por medio de las líneas #1 y #2 de forma sincrónica. Además de la línea de datos que transmite los bits individualmente (línea # 1), existe una línea de reloj que sincroniza la transmisión periódica de las señales (línea # 2). Ambas líneas son de colector abierto, lo que significa que están conectadas por medio de resistencias a la fuente de poder positiva (vcc) y pueden ser conectadas a

⁴ Ver tabla 4.1

tierra en cualquier extremo de la línea de comunicación en cualquier dirección. La transmisión se efectúa en incrementos de un bit, donde un bit de inicio se transmite primero (start bit), seguido por ocho bits de datos, comenzando por el bit menos significativo. Un bit de paridad, calculado usando paridad impar, sigue los ocho bits de datos. La transmisión concluye con un bit de final (stop bit) el cual forma el undécimo bit de la trama. En ambos lados de la línea de comunicación existen dispositivos que convierten las señales en la línea de datos en bytes y viceversa tanto en la computadora personal como en el propio teclado.

Aún cuando todos los tipos de computadoras personales utilizan este protocolo de comunicación, existen dos tipos de modelos PC/XT Y AT. Estos sistemas utilizan diferentes microprocesadores: a) el Intel 8048 usado en teclados PC y XT es un dispositivo no muy inteligente, sólo envía códigos de teclado al sistema. b) el microprocesador Intel 8042 puede hacer mucho más que eso haciendo la comunicación entre el sistema y el teclado más complicada donde el sistema también puede controlar partes del teclado.

En un teclado XT el sistema de colector abierto es un protocolo en el cual cada extremo de la línea se observa, el teclado revisa el estado de las líneas de datos y de reloj antes de enviar información a la computadora. Se tienen entonces dos posibilidades: a) si la línea del reloj está en bajo (lógica TTL), el teclado se deshabilita y no enviará la información. b) si la línea del reloj está en alto (cerca de 5 voltios pero no más alto) y la línea de datos en alto también el teclado podrá entonces comenzar a enviar su información. Estando las dos líneas en alto significa que la computadora no intenta enviar información o hacer que el teclado espere para mandar datos. Cuando las dos líneas están en alto, siendo éstas de colector abierto, el teclado puede tomarlas y comenzar a enviar el reloj, poner la línea de datos en bajo y empezar a enviar la información. La frecuencia del reloj es aproximadamente de 10 KHz.

Para el teclado AT, la comunicación básicamente se representa por un registro de estatus y pilas de entrada y salida. Estas pilas transmiten lo siguiente:

- Códigos de teclado de presionado y de liberación de tecla correspondientemente.
- Datos que el sistema requiere del teclado.

Estas pilas pueden utilizarse por medio del puerto 60H en sistemas AT.

La pila de entrada puede ser escrita tanto en el puerto 60H como en 64H.

El puerto que se usa depende del tipo de información que se transfiere. Si el sistema quiere enviar un código de un comando, el código deberá ser enviado al puerto 60H, con su código de datos correspondiente

por el puerto 64H, ambos llegarán a la pila de entrada del teclado, pero una bandera en el registro de estatus indicará cuándo es un comando y cuándo un dato.

Además con esta bandera, los bits 0 y 1 en el registro de estatus del teclado son especialmente importantes para la comunicación con el teclado. El bit 0 indica el estado de la pila de salida. Si este bit es 1, la pila de salida del teclado contiene información que aun no ha sido leída por el puerto 60H. La lectura en este puerto automáticamente pondrá de regreso a 0 este bit, indicando que no hay más caracteres en la pila de salida.

El bit 1 en el registro de estatus es siempre puesto a 1 cuando el sistema ha puesto un carácter en la pila de entrada antes que este carácter sea procesado por el teclado. Nada deberá ser escrito a la pila de entrada del teclado, hasta que este bit sea igual a 0, el cual indica que la pila de entrada está vacía.

e. **Velocidad de escritura:** Esta velocidad es el número de códigos por segundo que el teclado puede enviar al sistema cuando una tecla se mantiene presionada. Esta puede ser entre dos y treinta códigos por segundo. Para prevenir la repetición de caracteres sin intención, esta función no empieza hasta después del tiempo establecido. Este tiempo de espera lo establece el usuario y se codifica en binario de la siguiente forma:

Tabla 3
Códigos de espera

Código	tiempo	Código	tiempo
00b	¼ segundo	01b	½ segundo
10b	¼ segundo	11b	1 segundo

El teclado conservara la cantidad de tiempo con una tolerancia de 20%

El tiempo de repetición de los caracteres también es codificado en binario. La siguiente tabla muestra la relación entre el código de repetición y el número de repeticiones por segundo.

Tabla 4
Códigos de velocidad de repetición

CÓDIGO	RPS	CÓDIGO	RPS
11111b	2.0	01111b	8.00
11110b	2.1	01110b	8.60
11101b	2.3	01101b	9.20
11100b	2.5	01100b	10.0
11011b	2.7	01011b	10.9
11010b	3.0	01010b	12.0
11001b	3.3	01001b	13.3
11000b	3.7	01000b	15.0
10111b	4.0	00111b	16.0
10110b	4.3	00110b	17.1
10101b	4.6	00101b	18.5
10100b	5.0	00100b	20.0
10011b	5.5	00011b	21.8
10010b	6.0	00010b	24.0
10001b	6.7	00001b	26.7
10000b	7.5	00000b	30.0

El tiempo de espera codificado en binario y el código de repetición se combinan en un byte colocando los cinco dígitos enfrente a los tres de tiempo de espera.

No se puede enviar solamente este valor directo al teclado. Primero es necesario enviar el código apropiado (34H) y luego los parámetros de repetición. Los dos bytes son enviados al puerto 60H.

Es necesario el uso de un protocolo de transmisión que incluye la lectura de la palabra de estatus de teclado.

Para enviar datos al teclado el sistema primero leerá la palabra de estatus en una secuencia iterativa hasta que la pila de entrada del teclado esté libre, esta secuencia iterativa evita que se escriba algo en la pila del teclado cuando todavía no ha sido leído por el mismo.

Luego puede ser enviado el carácter por el puerto 60H. Para asegurarse que el carácter ha sido recibido el teclado envía otro de aceptación, para asegurarse que no ha existido un error en la comunicación (un error de paridad por ejemplo).

f. **Indicadores en el teclado:** El sistema utiliza el mismo método para cambiara los “LED’S” en el teclado AT de encendido a apagado y viceversa. El código de instrucción correspondiente es 0EDH y es nombrado instrucción indicadora de encendido / apagado.

Después que este comando ha sido transmitido con éxito, el teclado espera por el byte indicador del estado de los “LED’S” resumidos en la tabla siguiente:

Tabla 5
Códigos indicadores de teclado

NÚMERO DE BIT	LED
0	SCROLL LOCK
1	NUM LOCK
2	CAPS LOCK
3-7	SIN USO

Un bit en este byte representa uno de los tres “LED’S”, el cual es encendido cuando el bit correspondiente se activa. Activar y desactivar dichos “LED’S” no es labor del teclado sino del BIOS. Por ejemplo el teclado no convierte automáticamente las letras a mayúsculas cuando la tecla CAPS LOCK esta activada. El teclado solamente puede asociar una tecla como su número virtual en vez de un carácter específico. Este carácter es convertido a un código ASCII por el BIOS. Obviamente también se aplica para

la tecla NUM LOCK, donde el teclado simplemente envía el código correspondiente a la computadora cuando se presiona. El BIOS asigna la función de la tecla NUM LOCK escribiendo un uno (1) a la bandera interna, esta indica que está activa la función, luego envía la instrucción indicadora activar-desactivar al teclado (OEDH) seguido de la palabra de estatus al teclado para encender el “LED” correspondiente.

Incluso cuando dichas modalidades del teclado son habilitadas y deshabilitadas por el usuario, al presionar las teclas correspondientes, podría ser útil hacerlo desde un programa. Esto se aplica a aquellos que tienen teclas separadas para cursores y teclado especial para números; por ejemplo: en algunos teclados solamente es posible ingresar números cuando la tecla “NUM LOCK” está activa, tiene sentido activarla automáticamente cuando el sistema está encendido.

Para hacer esto simplemente se debe activar la bandera adecuada en el BIOS y después encender el “LED” correspondiente en el teclado para informarle al usuario que dicha tecla está activada.

En la práctica un programa debe activar el modo del BIOS correspondiente dado que el BIOS automáticamente controla los “LED’S”.

2. El Microcontrolador PIC 16F84: Este microcontrolador de 8 bits puede utilizarse en gran variedad de aplicaciones ya que posee una memoria de programación EEPROM, de 1K X 14, una memoria de datos de 64 bytes y 36 registros de propósito general. Además posee sus propios periféricos dentro de los cuales están:

- Un controlador / temporizador RTCC de propósito general de ocho bits y una escala programable de ocho bits, haciéndolo un temporizador de 16 bits. En modo temporizador el RTCC será incrementado cada ciclo instrucción sin preescala, este modo se selecciona poniendo un 0 en el bit TOCS del registro OPTION_REG. El asignamiento de la preescala se controla por software por el bit de control PSA en el registro OPTION_REG.

- Oscilador Interno seleccionable
 - ❖ Oscilador RC.
 - ❖ Cristal/ Resonador XT utilizado en el teclado especial, el circuito para este consiste únicamente de un cristal y dos capacitores como se muestra en el diagrama del circuito (apéndice A).

- ❖ Cristal / resonador de alta velocidad HS.
- ❖ Cristal de bajo consumo baja frecuencia.
- Trece pines de entrada o salida I/O: organizado como dos puertos, Puerto A de 5 bits y Puerto B de ocho bits. Cada uno de los pines puede ser programado como entrada o salida a través de los registros internos del microcontrolador TRISA y TRISB, también posee dos registros internos para almacenar el estado de los puertos PORTA y PORTB, respectivamente.

a. Descripción de los pines:

Tabla 6
Descripción de pines 16F84

Nombre de pin	tipo	Operación normal	Modo de programación
Vdd	P	Energía	Energía
Vss	P	Tierra	Tierra
OSC1/CLKIN	I	Entrada de reloj Entrada de oscilador	
OSC2/CLKOUT	I/O	Salida de reloj en RC/ Conexión de reloj en los otros modos	
MCLR/Vpp	I/P	Entrada externa de Restauración. Negada	Se le aplica alto voltaje para programación
RA0	I/O	Pin bidireccional. Nivel TTL	
RA1	I/O	Pin bidireccional. Nivel TTL	
RA2	I/O	Pin bidireccional / Nivel TTL	
RA3	I/O	Pin bidireccional / Nivel TTL	
RB0/ INT	I/O	Pin bidireccional / Inter. Externa	
RB1	I/O	Pin bidireccional / Nivel TTL	
RB2	I/O	Pin bidireccional / Nivel TTL	
RB3	I/O	Pin bidireccional / Nivel TTL	

Continuación Tabla 6

Nombre de pin	tipo	Operación normal	Modo de programación
RB4	I/O	Pin bidireccional / Nivel TTL	
RB5	I/O	Pin bidireccional / Nivel TTL	
RB6	I/O	Pin bidireccional / Nivel TTL	Entrada de reloj
RB7	I/O	Pin bidireccional / Nivel TTL	Datos entrada / salida

El microcontrolador 16F84 posee también cuatro niveles de interrupción:

- Pin de interrupción externo RB0: El cambio de estado de este pin provoca una interrupción a la posición de memoria de programa 0004. En el teclado especial cada vez que se presiona un botón provoca un cambio en RB0 y a partir de la posición de memoria de programación 0004 se lee qué botón fue presionado para establecer la rutina a seguir (ver diseño del programa).
- Desbordamiento del temporizador TRC: Una interrupción se genera cada vez que exista un desbordamiento del temporizador.
- Cambio de los pines 7-4 Puerto B, una interrupción se genera también cuando los valores de estos pines son cambiados.

El PIC 16F84 se escogió para el teclado especial por sus características dentro de las cuales cabe mencionar: posee la cantidad suficiente de pines de entrada y salida tanto para manejar los tres botones del teclado como las líneas de control y datos de la pantalla de cristal líquido; su tamaño lo hace perfecto para esta aplicación; la memoria es suficiente para almacenar el programa; el circuito externo del reloj ocupa muy poco espacio y el temporizador es ideal para su uso en las rutinas de espera.

Además el microcontrolador utiliza muy poca energía, una característica ideal para la aplicación del teclado, teniendo en cuenta que, toda alimentación del circuito del teclado especial proviene de la línea de 5 voltios del teclado convencional.

Debido al bajo consumo de energía y tomando en cuenta que máxima corriente de salida es de 25 mA. Se reducen los manejadores externos y por lo tanto una reducción de costos.

El PIC16F84 utiliza lenguaje de programación assembler, un simulador y un programador. El sistema de programación utiliza solamente dos pines para la transferencia de datos y ofrece flexibilidad para adiestramiento del producto después de completar el ensamblaje y revisión. Estas herramientas pueden usarse en computadoras IBM y máquinas compatibles.

3. Módulo de pantalla de cristal líquido: El LC-II (HD44780) es un controlador de pantalla líquida de matriz de punto el cual despliega caracteres alfanuméricos y símbolos. Puede manejar la pantalla de cristal líquido mediante 4 u 8 bits controlados por un microprocesador. Todas las funciones internas requeridas para el manejo de la pantalla son proveídas internamente por un chip.

a. Descripción de los pines:

Tabla 7

Descripción pines módulo de pantalla de cristal líquido

Nombre	Tipo de pin	Descripción
E	C	Habilitadora
R/W	C	Lectura o escritura
RS	C	Registro de instrucción o datos
DB0	I/O	Pin bidireccional. Nivel TTL
DB1	I/O	Pin bidireccional. Nivel TTL
DB2	I/O	Pin bidireccional. Nivel TTL
DB3	I/O	Pin bidireccional. Nivel TTL
DB4	I/O	Pin bidireccional. Nivel TTL
DB5	I/O	Pin bidireccional. Nivel TTL
DB6	I/O	Pin bidireccional. Nivel TTL
DB7	I/O	Pin bidireccional. Nivel TTL

Continuación Tabla 7

Nombre	Tipo de pin	Descripción
Vdd	P	Voltaje de alimentación 5 V
Vo	P	Voltaje variable
Vss	P	Tierra.

b. Características:

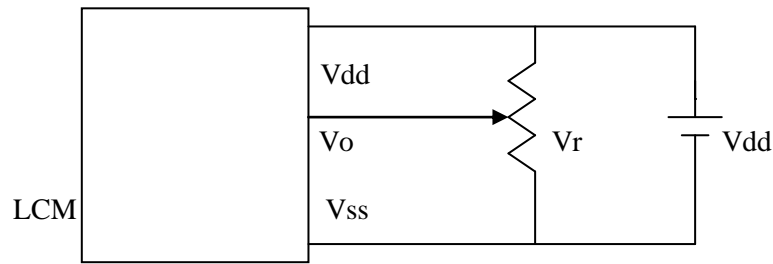
- Capaz de manejar interfaces de 4 u o 8 bits.
- Memoria RAM de la pantalla (80 X 8 bits) hasta 80 caracteres.
- Generador de caracteres ROM.
- Ambas memorias pueden ser leídas por el microprocesador.
- Amplio rango de instrucciones de funciones.
- Restauración automática al ser encendido.

El pin RS selecciona el tipo de registro a leer o escribir, A=0 selecciona un registro de instrucción y A=1 selecciona un registro de datos.

Los pines DB0 – DB7 pueden manejar una capacitancia de 130 pf. Las terminales del bus de datos son de tres estados y permanecen en alta impedancia mientras la señal de habilitadora (E) esté en 0.

También posee tres terminales de poder Vdd Vo y Vss. El módulo de la pantalla de cristal líquido es manejado por el voltaje que es igual a $V_{dd} - V_o$. Como la temperatura podría afectar la fuente de voltaje que alimenta la pantalla de cristal es posible variar el voltaje de V_o mediante un potenciómetro como se muestra en la figura 3, de esta forma se puede ajustar el voltaje de la pantalla.

Figura 3



Circuito de alimentación del módulo de pantalla de cristal líquido

c. **Interfase con el microprocesador:** Cuando la interfase de datos es de cuatro bits, estos son transferidos usando un bus de cuatro líneas DB4 – DB7. Las líneas DB0- DB3 no se utilizan. Primero son transferidos los 4 bits más significativos (DB4 – DB7) y luego los 4 bits menos significativos (DB0 – DB3).

La bandera de ocupado puede utilizarse después de transferir tanto la parte alta como la baja para poder enviar el siguiente dato, esta bandera puede revisarse fácilmente leyendo el pin DB7 del bus de datos.

Cuando la interfase de datos es de ocho bits, estos son transferidos en los pines DB0 – DB7. A través del bus de datos pueden enviarse al módulo de la pantalla dos tipos de información:

- **Instrucción:** cuando se pone un cero en la línea de RS se accede al registro de instrucciones, no son datos a desplegar a la pantalla sino instrucciones específicas del manejo del mismo.
- **Datos:** Son códigos de los caracteres que serán desplegados a la pantalla directamente, se accede al registro de datos poniendo un 1 en la línea RS.

Solamente estos dos registros pueden ser manejados por un controlador externo. Antes que la operación interna comience, la información de control se almacena temporalmente en estos registros, para permitir una interfase entre HD44780 y varios tipos de microprocesadores, los cuales operan a diferentes velocidades o para permitir una interfase entre el control de sus periféricos. La operación interna del HD44780 se determina por señales enviadas desde el microprocesador. Dentro de estas señales se incluye registros de selección (RS) señales de lectura y escritura (R/W) y un bus de datos (DB0 – DB7).

Existen cuatro tipos de instrucciones:

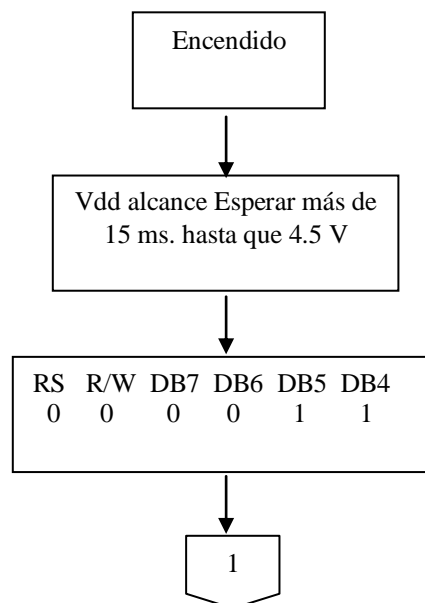
- Funciones específicas de HD44780 como formato del despliegue, longitud de datos etc.

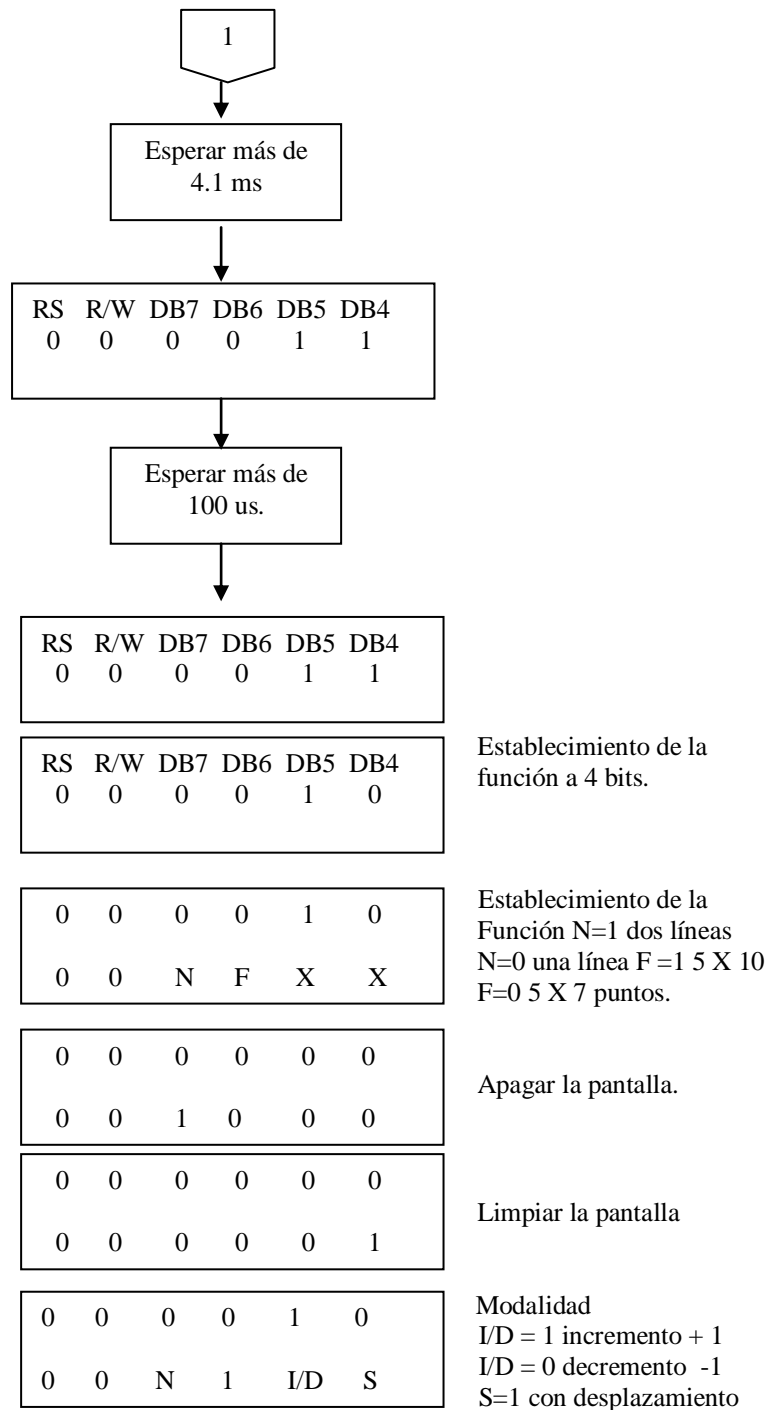
- Dar una dirección interna a la memoria RAM.
- Realizar transferencia de datos dentro de la memoria RAM.
- Otro tipo.

En uso normal las instrucciones del numeral 3 son las más utilizadas, incluso cuando existe un incremento automático de uno (o decremento en uno) de la dirección de la memoria, cada escritura de un dato disminuye la carga de programa del microcontrolador. El desplazamiento de la pantalla es especialmente útil para realizar lecturas concurrentes en el módulo. Cuando una instrucción se ejecuta durante una operación interna, no deberá realizar otra instrucción que la lectura de la bandera de ocupado. Como la bandera de ocupado se pone en 1 cuando una instrucción ha sido ejecutada, dicha bandera debe estar en 1 antes de enviar una instrucción desde el microcontrolador.

Nota 1: Asegúrese que el HD44780 no esté en estado de ocupado (BF=0) antes de enviar una instrucción desde el microcontrolador hacia el HD44780. Si la instrucción se envía sin revisar la bandera de ocupado, el tiempo entre la primera y la siguiente instrucción deberá ser mayor que el tiempo de la instrucción.

d. Iniciación:





e. Diagrama de tiempos: El diagrama de tiempos (ver apéndice E) muestra la temporización entre cada una de las líneas de control y las de datos, estos tiempos máximos y mínimos están resumidos en la

tabla. Dichos tiempos deberán ser tomados en cuenta a la hora de enviar información del microcontrolador a la pantalla.

B. Marco experimental:

1. Verificación de la trama de datos y prototipos: Con base en la investigación bibliográfica y por Internet acerca del funcionamiento del teclado, con el objeto de conocer sobre los componentes del mismo, las diferencias entre teclados AT y XT, así como cada una de las líneas de comunicación entre el teclado y la computadora se encontró que dicha información no establecía claramente cómo es la trama de datos.

El primer método para establecer dicha trama fue conectando la línea de datos y reloj a dos entradas del osciloscopio; con este primer método no se pudo establecer claramente la línea de datos ya que no son transmitidos continuamente, aún dejando presionada la tecla para que el teclado envíe códigos continuamente. Como se sabía por la investigación del teclado, los tiempos entre transmisión y transmisión no permitieron observar en la pantalla la trama de manera continua, lo único que se pudo determinar fue la frecuencia del reloj la cual efectivamente estaba alrededor de los 10 KHz.

El segundo método para establecer la trama fue el diseño del circuito que se describe a continuación (ver apéndice B).

El circuito consta de 6 registros de corrimiento de ocho bits 74LS164, conectados en cascada; estos chips operan de la siguiente forma: poseen dos entradas A y B en los pines 1 y 2 respectivamente, si una de las dos entradas está en 0 inhibe la entrada de datos a los registros y restablece el primer “flip-flop” a bajo nivel al siguiente pulso de reloj. De esta manera se conectó el pin B a Vcc, la línea de datos del teclado a la entrada A del primer 74LS164 y el último bit del registro conectado al pin de entrada A del siguiente registro, la línea de reloj al pin 8 de cada uno de los registros y un bus conectado a cada uno de los pines de CLR, para poder borrar todos los registros de una vez conectándolo a 0.

Con el circuito conectado de esta forma se tiene 48 “flip-flop” que se utilizaron como registros para almacenar cada uno de los “SCAN CODES”. Así con un teclado convencional conectado a una fuente de poder, la línea de reloj del teclado conectado al bus del reloj de los registros y la línea de datos conectada a la entrada del primer registro, con sólo presionar una tecla se puede entonces tener toda la trama de datos (6

bytes) almacenada en estos registros; de esta forma al presionar cada una de las teclas se obtiene el código de cada una de ellas.

Al seguir este procedimiento se obtienen los “SCAN CODES” correctos, además de la certeza de la trama de datos que corresponden a cada uno de los códigos, ya que dentro de la información obtenida por Internet no están bien definidos dichos códigos, de esta forma se obtuvo los “SCAN CODES” para las teclas F11 y F12.

De las bases del funcionamiento del teclado se sabe que la trama de datos está compuesta por un bit de inicio, ocho de datos comenzando por el menos significativo, un bit de paridad y uno de parada, pero la información recabada no indica si estos bits son unos o ceros, tampoco especifica si la línea de datos es de flanco positivo o negativo; es decir, ¿la computadora lee el bus de datos cuando cambia de uno a cero? ¿O de cero a uno? Todas estas interrogantes se resuelven mediante este método de trabajo: primero se conectó el circuito en un “protoboard” y se comenzó a almacenar datos en estos registros de corrimiento presionando cada uno de los botones, por ejemplo al presionar y liberar W se obtiene la siguiente trama:

	STOP BIT		SCAN CODE	
	↑		┌──────────┐	
1	1	0001	1101	0
1	1	1111	0000	0
	↓		PARITY	START BIT

El primer bit, de izquierda a derecha, es el bit de inicio o “start bit”, los siguientes ocho bits son el código de la tecla; luego un bit de paridad y uno más de parada o “stop bit”. Luego comienza nuevamente con otro bit de comienzo y el código F0 con sus respectivos bits de paridad y parada. Después se repite nuevamente la secuencia con el código de la tecla.

Estos son solamente los 33 bits que se almacenaron al presionar la tecla W y liberarla, comparando el código de esta tecla con la documentación obtenida (ver tabla 4.1) se puede determinar que efectivamente es 1D en hexadecimal, se puede entonces asumir lo siguiente: 1) el start bit es un cero y 2) la paridad se calcula de la siguiente forma: si el número de unos es par el bit de paridad es uno y si es impar es un 0 y el

bit de parada es un uno. También es posible confirmar que al liberar la tecla este manda el código F0 y luego el “SCAN CODE” de la tecla nuevamente.

De esta forma se ratifica los códigos de las teclas ya conocidas y se investiga las que no, por ejemplo F11 y F12 cuyos códigos son 78 y 07 respectivamente.

Una vez establecido cada uno de los bits de la trama de datos, aún existía el problema de los tiempos de mínimos y máximos entre el bus de datos y el de reloj; al carecer un diagrama de tiempos como el del módulo de pantalla de cristal líquido, se optó entonces por crear programas preliminares para enviar un código de forma consecutiva, como el que se muestra a continuación:

```

INCLUDE C:\PIC\P16C84.INC
LIST P=16C84

DIGITO EQU H'000C'
CONT EQU DIGITO+1
TEMP EQU DIGITO+2
    CLRF DIGITO
    CLRF CONT
    CLRF TEMP
    BSF STATUS,RP0
    BCF OPTION_REG,T0CS ; HABILITAR EL TIMER
    CLRF TRISB ; PUERTO B COMO SALIDA
    CLRF TRISA ; PUERTO A COMO SALIDA
    BCF STATUS,RP0
    CLRF TMR0
    MOVLW B'11111111'
    MOVWF PORTB
    MOVWF PORTA ; SE PONE EN 1 AMBAS LINEAS
COM
    MOVLW H'23'
    MOVWF DIGITO ; SCAN CODE A TRASMITIRSE
INICIO
    CALL DELAY
    INCF CONT,F
    MOVLW B'00000000'
    MOVWF PORTB
    MOVF CONT,W
    SUBLW .1
    MOVLW H'00'
    BTFSC STATUS,Z
    MOVWF PORTA
    MOVF CONT,W
    SUBLW .9
    MOVLW B'11111111'
    MOVWF TEMP
    BTFSS STATUS,C
    GOTO SEGUIR
    MOVF DIGITO,W
    BCF STATUS,C

```

```

RRF DIGITO,F
MOVWF TEMP
SEGUIR
CALL DELAY
MOVLW B'1111111'
MOVWF PORTB
MOVF TEMP,W
MOVWF PORTA
MOVF CONT,W
SUBLW .11
BTSS STATUS,Z
GOTO INICIO
CLRF CONT
LOOP
CALL DELAY
INCF CONT,F
MOVF CONT,W
SUBLW H'FF'
BTSS STATUS,Z
GOTO LOOP
CLRF TMR0
CLRF CONT
GOTO COM
DELAY
NOP
MOVF TMR0,W
SUBLW .15
BTSS STATUS,C
GOTO DELAY
CLRF TMR0
RETURN
END

```

El programa que se describe es muy parecido a la rutina serial de programa definitivo (ver apéndice C), pero en él no se hace el cálculo de paridad ya que únicamente envía el código de la letra D, primero pone a cero la línea de reloj y luego la línea de datos empieza con el bit de inicio, continúa con los 8 bits de datos, el bit de paridad, que es uno y el de parada para completar los 11 bits, luego espera un tiempo largo hasta que llegue a 256 el contador CONT, para luego volver a enviar el código.

Este programa fue el primero en el que se transmitió el carácter D hacia la computadora sin embargo, no siempre lo recibía y rápidamente la computadora se alarmaba porque tenía la pila de entrada llena.

Con base en lo anterior se optó por enviar primero el bit de inicio antes de poner a cero la línea de reloj mediante la rutina DELAY 2, en otras palabras, se envía primero el bit de inicio, pasa a la rutina DELAY 2 y luego comienza el reloj; para cuando la línea de reloj cambia a cero ya está en la línea de datos el bit correspondiente y unos instantes más tarde es cambiado dicho bit; también se optó por cambiar la rutina

para enviar ambos códigos de liberación de tecla, con esto se revolió el problema de que no siempre identificaba al código de la tecla seleccionada, concluyendo entonces que la línea de reloj actúa en el flanco negativo, es decir que cada uno de los bits de la transmisión serial es leído cuando cambia de uno a cero la línea de reloj. En las bases de funcionamiento de teclado convencional solamente establece que permanecen en 1 ambas líneas mientras no se envíen datos entre el teclado y la computadora pero no establece que es necesario que la línea de datos esté estable al momento de bajar la línea de reloj, todos estos tiempos se calcularon experimentalmente, variando dichos tiempos hasta obtener el mejor resultado.

Una vez establecidos la trama de datos y con la rutina de envío de caracteres funcionando hacia la computadora, se procedió entonces, al diseño del circuito definitivo y al diseño de programa completo tal como se describen en las siguientes dos secciones.

2. **Diseño del circuito:** El circuito está constituido por un PIC 16F84, el cual es el microcontrolador que se encarga de manejar tanto la pantalla donde se despliegan cada una de las letras y símbolos correspondientes a cada una de los caracteres del teclado, como el envío de los códigos especiales hacia la computadora es en paralelo con un teclado convencional; cada vez que se presiona el botón central, se envía un código directamente a la computadora a través de dos líneas una de datos y una de reloj, en virtud que la comunicación entre el teclado y el CPU es asincrónica.

Como se menciona anteriormente (ver marco teórico) se utilizan 5 líneas de conexión entre el teclado y el CPU: la primera corresponde a la línea de datos, una de reloj, restablecimiento, tierra y alimentación 5 V consecutivamente; el teclado especial se conecta en paralelo la línea de datos, de reloj y utiliza la energía de la fuente de alimentación entre 5 voltios y tierra, esta última se utiliza para todos los componentes activos del circuito, de la misma forma que un teclado convencional. Se utiliza un conector macho para conectarlo directo al CPU por medio de un cable y hacia el circuito del teclado especial y un conector hembra donde deberá ser conectado el teclado convencional. La línea de reloj se conecta directamente a la salida del microcontrolador, sin embargo para la línea de datos es necesario un relee para separar el teclado especial del convencional, en razón de la comunicación bidireccional que existe entre el teclado convencional y el CPU, sobre todo al momento del arranque de la computadora; además las teclas como CAPS LOCK, NUM LOCK y los "led's" correspondientes a dichas teclas utilizan, como se explica en la parte del teclado, una

comunicación bidireccional también. Debe separarse la línea de datos a través de un rele, este tiene dos contactos uno normalmente cerrado NC y otro normalmente abierto NO; existe un pin común entre estos dos contactos así cuando el rele no se ha activado NC se mantiene una conexión entre el teclado convencional y la computadora para permitir el intercambio de información, cuando el rele se activa corta la comunicación anterior y establece otra entre la línea de datos serial del microcontrolador y el CPU, esta comunicación dura el tiempo necesario para enviar el “SCAN CODE” y luego vuelve a su estado normal.

Se muestra en el diagrama del circuito (ver apéndice A) que el pin 3 correspondiente a la salida A4 del microcontrolador se utiliza para activar el rele, considerando que este pin en especial es de colector abierto es necesario conectarla a 5 voltios mediante la resistencia R9, luego está conectada a la resistencia R8 junto con el transistor Q1 y el rele, cuando el pin A4 está en 1 el transistor Q1 conecta el rele a tierra activándolo y conectando la línea de datos hacia el computador.

El controlador PIC1684 utiliza un cristal de 4 Mhz. y dos capacitores C1 y C2 para su reloj, interno, los cuales están conectados en las terminales OSC1 y OSC2, esta opción del oscilador es la llamada XT (ver funcionamiento del microcontrolador) y es la opción de reloj que fue colocada en el principio del programa del PIC.

Las terminales A0 – A3 se utilizan como bus de datos para la pantalla de cristal líquido, como se muestra en la tabla 8, son necesarias únicamente 4 líneas ya que primero se envía la parte alta del carácter y luego la parte baja.

Además de estas cuatro líneas de datos son necesarias tres más para las líneas de control:

- B5 corresponde a la señal RS del controlador de la pantalla líquida, y se utiliza para diferenciar un código de datos a desplegar o un comando hacia el módulo de pantalla de cristal líquido.
- B6 se utiliza como señal de lectura / escritura (R/W).
- B7 es la línea habilitadora del módulo de la pantalla de cristal líquido.

En la sección 1 del Marco teórico se explica que la comunicación entre el teclado y el computador es asíncrona y para ello es necesario una línea de datos y una de reloj:

- Línea de datos: lleva cada uno de los códigos especiales del carácter a desplegar en una transmisión serial y para la cual se utiliza el pin RB3.

- Línea de reloj: programado en el controlador como salida en el pin RB4, proporciona la línea de reloj a 10 KHz.

Tabla 8

Uso de los pines del controlador

Nombre del pin	Entrada o salida	Descripción
RB0	Entrada	Indicador de algún botón presionado.
RB1	Entrada	Botón derecho
RB2	Entrada	Botón Izquierdo
RB3	Salida	Datos
RB4	Salida	Reloj
RB5	Salida	Registro de inst./datos (módulo de cristal líquido)
RB6	Salida	Lectura / escritura (módulo de cristal liquido)
RB7	Salida	Habilitadora (módulo de cristal liquido)
RA0	Salida	1er Bit del bus de datos (módulo de cristal líquido)
RA1	Salida	2do. Bit del bus de datos (módulo de cristal líquido)
RA2	Salida	3er. Bit del bus de datos (módulo de cristal líquido)
RA3	Salida	4to. Bit del bus de datos (módulo de cristal líquido)
RA4	Salida	Relee

En el diagrama del circuito (ver apéndice A) se muestra que a la salida de RB3 y RB4 están conectados un inversor por medio de una compuerta 74LS04 y a la salida de este nuevamente otro inversor por medio de un transistor, la señal primero es invertida por la compuerta y luego otra vez por medio del transistor; Con el objeto de 1) Proporcionar la corriente necesaria de la transmisión serial hacia la computadora ya que los pines programados como salidas del PIC no proveen de la corriente necesaria y b) utilizar lógica positiva dentro del programa y no una lógica negativa que impediría un mejor entendimiento del mismo.

Los pines 6, 7 y 8 fueron programados como entradas en el microprocesador, los pines 7 y 8 están conectados al los interruptores SW1 y SW2; cada vez que presiona el botón de la derecha. => El interruptor SW1 se cierra y conecta la entrada RB1 a tierra o un cero según la lógica positiva TTL, cuando es liberado regresa a su estado normal conectado a través de una resistencia a 5 voltios (1 lógico); de la misma forma el interruptor SW2 está conectado a la entrada RB2 en el pin 8 del microprocesador y opera de la misma forma que el anterior con el botón izquierdo <=.

Los interruptores SW1 Y SW2 también están conectados a la entrada de una compuerta AND 78LS08 y esta, a su vez, se conecta a la entrada de otra compuerta AND 78LS08 junto con el interruptor SW3 correspondiente al botón central; cada vez que se presiona uno de estos botones a la salida de la segunda compuerta AND habrá un 0 que negado por medio de la compuerta NOT será un 1 conectado al pin 6 correspondiente a RB0/INT del microcontrolador, este último se utiliza para generar una interrupción dentro del programa. La interrupción (ver programa apéndice C) ocasiona la lectura de los pines 6 y 7; si el pin 6 (RB0) está en 0 el botón derecho fue presionado; si el pin 7 (RB1) está en 0 el botón izquierdo fue presionado y si se generó una interrupción y ninguno de las entradas RB0 o RB1 están en 0 el botón central fue presionado, nótese que este botón solamente ocasiona una interrupción ya que no está conectado a ninguna entrada del controlador, con el objetivo de utilizar los pines de entrada o salida (según sean programados) de una forma más eficiente.

3. Diseño del programa: El programa del PIC16F84 está diseñado para dos tipos de teclados, en el teclado 1 sólo trabaja con letras y números, mientras que el teclado 2 contempla la totalidad de los símbolos de teclado, y por ello, será el que se explique a continuación.

Previo a la explicación del programa principal es conveniente exponer algunos de los procedimientos utilizados dentro del cuerpo del mismo, para evitar que siguiendo el orden del programa se llame a rutinas que no se han explicado (ver apéndice C).

a. Rutina RET: Su función es hacer que el PIC espere un milisegundo, para esto se utiliza un controlador interno CONT5, este va disminuyendo de uno en uno desde 255 hasta cero, tomando en cuenta

que cada ciclo del PIC toma un microsegundo con un reloj de 4 Mhz. y que dicha rutina tiene tres instrucciones, dos de un ciclo y la última de dos ciclos, con lo cual se obtiene el siguiente resultado:

$$1 \text{ microsegundo} \times 4 \times 255 = 1.02 \text{ ms.}$$

b. Rutina DELAY: dicha rutina tiene la función de esperar un tiempo de aproximadamente 250 ms. y utiliza para ello dos contadores CONT 5 Y CONT 4 en cascada uno con el otro.

c. Rutina DELAY2: opera diferente a las dos anteriores por utilizar el temporizador interno del PIC, como se explicó con anterioridad este temporizador decrementa su valor en uno cada ciclo de reloj; en esta rutina se revisa que el contador llegue a 26, cuando llega a 26 o más regresa al procedimiento del cual fue llamado. Se utiliza para la transmisión serial (ver rutina siguiente) para hacer posible la línea de reloj del teclado a 10 Mhz. Tomando en cuenta 26 ciclos de reloj más el tiempo que tarda la rutina de transmisión serial en poner un 1 ó un 0 se obtiene un tiempo aproximado de medio ciclo a 10 Mhz.

d. Rutina SERIAL: Antes de comenzar la rutina en sí, tiene una bandera llamada SERIAL 2, esta última se utiliza para hacer una llamada a la rutina TABLA 2 donde obtiene "SCAN CODE" de la tecla que será transmitida.

Tiene la función de desplegar de manera serial en dos líneas, la línea de reloj y la línea de datos, tomando en cuenta que ambas están sincronizadas por la rutina anterior DELAY2. Esta rutina se repite hasta que se completan los 11 bits, se utiliza la variable CONT para llevar el control del número de bit que se transmitirá de la siguiente forma a) si es el primer bit o bit de comienzo (start bit) se pone un cero en la línea de datos, luego se pone en cero la línea de reloj y se va corriendo hacia la derecha la variable DIGITO, donde se almacenó el dato que se obtuvo de la rutina TABLA 2, el bit a transmitirse se almacena en la posición 0 de la variable TEMP c) si el número de bit a transmitirse está entre segundo y noveno solamente hace el corrimiento de la variable DIGITO d) si es el décimo bit a transmitirse, se almacena en TEMP el bit de paridad y el programa ya no efectúa el corrimiento de la variable DIGITO, e) si se trata del undécimo bit el programa tampoco efectúa corrimiento y coloca en la variable TEMP el bit de parada (stop bit).

Después de todas las condiciones anteriores en las cuales se determina qué bit será colocado en la variable TEMP se procede a levantar la línea de reloj y poner el bit 0 de la variable TEMP en la línea de datos. Es de notar que entre el cambio de la línea de reloj de cero a uno y viceversa se llama a la rutina DELAY2, la cual como se explicó anteriormente, espera un tiempo específico para que la frecuencia de línea de reloj sea de 10 Khz.

Dentro de esta rutina también se calcula el bit de paridad de la siguiente forma: cada vez que el dato a desplegar es corrido hacia la derecha se examina si es uno o un cero; si es uno, la variable PARIDAD se incrementa en uno, así en cada corrimiento se cuenta el número de unos que el byte contiene; así que cuando llegue el décimo bit (de paridad), dicha variable contendrá el número de unos del byte de datos; el programa examina el bit 0 de esta variable, si es 0 la cantidad de unos es par y pondrá en la variable TEMP un 1, si es la cantidad de unos impar y pondrá en la variable TEMP un cero.

e. Rutina ENVIAR: Esta rutina tiene la función de enviar un carácter a desplegar en la pantalla de corrimiento, para ello se sigue el diagrama de tiempo descrito (ver apéndice E) dentro de las especificaciones del mismo. Para poder enviar el carácter primero se guarda en la variable DESP y luego se divide en dos: la parte alta se almacena en la misma variable y la parte baja en la variable DESP2, primero se envía la parte alta en 4 bits y luego la parte baja. Para esto se utiliza el puerto A, ya que es de 5 bits y el quinto es usado para el manejo del relee (ver rutina serial). Es necesario dejar el bit 4 de las variables DESP y DESP2 en las condiciones iniciales para no afectar el bit del relee, por esto en el programa se pone el bit 4 de ambas variables en 0.

Antes de mandar a desplegar el dato se llama a la rutina RET, la cual espera el tiempo necesario para que el módulo de pantalla de cristal líquido reciba, procese y despliegue el carácter. El tiempo que dilata la rutina RET es suficiente según las especificaciones del módulo para no tener que leer la bandera de ocupado, pues es el tiempo máximo que tarda un carácter en desplegarse (ver apéndice E). Se incluye en esta rutina el tiempo de espera para evitar efectuar, en el programa principal, llamadas a la rutina RET cada vez que se va a llamar a la rutina DESP, por ser muchas las veces que, dentro del programa principal, se llama a dicha rutina, a fin de ahorrar memoria de programa.

Para desplegar el carácter se siguen los siguientes pasos: 1) se ponen en cero los bits 5 y 6 del puerto B, correspondientes a las líneas RS y R/W en el módulo de la pantalla respectivamente, 2) se pone en 1 el séptimo bit del puerto B correspondiente a la línea habilitadora del controlador 3) se pone la parte alta del carácter a desplegar en el puerto A, 4) se pone la línea habilitadora en 0, 5) se ponen la líneas de RS y R/W de regreso a 1 y finalmente 6) se retiran los datos del puerto A poniéndolo a 0.

Para desplegar la parte alta del carácter se repite el procedimiento anterior, con la diferencia que en vez de desplegar la variable DESP en el puerto A se despliega la variable DESP2 en el mismo puerto. Este procedimiento se efectúa a continuación del anterior.

Ya habiendo mandado tanto la parte alta primero y seguidamente la baja el módulo de la pantalla recibe los 8 bits que componen el carácter, para posteriormente desplegarlo.

La temporización sigue las especificaciones del módulo de la pantalla, es decir los tiempos mínimos y máximos que los datos y líneas de control deben permanecer para la comunicación entre el PIC y el módulo.

En la parte central de la rutina se encuentra MEDIO y sirve para la iniciación del módulo, saltando a esta parte, solamente se envían 4 bits como se requiere, en la explicación del cuerpo del programa se explica como se lleva a cabo.

f. **Rutina ENVIAR2:** Esta difiere de la anterior en que, en vez de poner a 0 RS se pone a 1, y se utiliza para enviar datos al módulo. La rutina ENVIAR se utiliza para desplegar palabras de control y la rutina ENVIAR2, para desplegar datos.

g. **Rutina ADELANTE:** Primero llama a la rutina común para desplegar el carácter, luego revisa la variable RUTINA para saber si se presionó algún botón; ya que cada vez que se presiona algún botón se genera una interrupción, lee el puerto y se almacena en la variable RUTINA si es que la interrupción proviene de esta parte del programa. Si la interrupción proviene del programa principal, en el momento en que se muestran las barras para seleccionar la velocidad de despliegue de caracteres, se almacena en la variable GUARDA, revisa la variable anterior para ver qué botón se presionó, para luego poder desviarse a la rutina correspondiente, ya sea INA o SCAN; si no se ha presionado ningún botón vuelve a leer el puerto B para revisar si todavía está presionando el botón de la derecha, considerando que la interrupción sólo se

genera cuando hay un cambio RB0. Si después de haber generado la interrupción, sigue la entrada de RB1 en 1 significa que el botón continúa presionado; cuando esto sucede, la velocidad de despliegue se incrementa a su máximo hasta que se libere el botón derecho. El objetivo de incrementar así la velocidad es una vez que la persona está utilizando el teclado y desee que los caracteres sean desplegados con mayor rapidez sólo tiene que dejar presionado dicho botón. Esta rutina es recurrente hasta que se presiona algún botón que provoca que el programa continúe con otro procedimiento, si esto no ocurre se incrementa el contador SALTO, vuelve al inicio de esta rutina y mediante el procedimiento COMUN despliega el siguiente carácter. Si esta última variable llega a 76 correspondiente al último carácter, se vuelve a poner en 1 para reiniciar el despliegue del primer carácter de forma cíclica en ambos sentidos.

h. Rutina INA: Es muy similar a la rutina ADELANTE, con la diferencia que en vez de incrementar la variable SALTO la disminuye para desplegar los caracteres hacia atrás. También lee el puerto B y revisa el bit correspondiente de igual forma que la rutina ADELANTE, cuando se deja presionado el botón izquierdo aumenta la velocidad de despliegue hacia atrás. Una vez la variable SALTO llega a 0 se asigna el valor 75 para desplegar el último carácter nuevamente, para mantener el despliegue cíclico.

i. Rutina SCAN: Esta rutina se encarga de enviar el código correspondiente del carácter seleccionado hacia la computadora, previamente revisa si el código que se mandó a desplegar corresponde a las teclas de SHIFT, ALT o CTRL, para llamar a la rutina correspondiente a cada una de ellas; cada vez que se llaman a estas rutinas se incrementa un contador y se revisa si es par o impar mediante el bit menos significativo: si es impar el contador envía el “SCAN CODE” de tecla presionada hacia la computadora sin código de tecla liberada, si es par envía ambos “SCAN CODES” de tecla liberada.

El objetivo de mandar una vez el “SCAN CODE” de tecla presionada y ambos “SCAN CODES” de tecla liberada en forma alternada es que estas teclas específicas operan como encendido / apagado, una vez seleccionada alguna de estas teclas por primera vez queda activa, es decir simulando que se dejó presionada dicha tecla en un teclado convencional, al seleccionarla por segunda vez queda desactivada, simulando la liberación de la misma. Cuando se trata de un “SCAN CODE” de una tecla convencional simplemente se

envía el “SCAN CODE” de tecla presionada y luego los dos de tecla liberada, el tiempo entre códigos está determinado por una rutina de retraso en la cual se utiliza la variable CONT4.

j. Rutina COMUN: Con esta rutina comienza en sí el despliegue de los caracteres hacia la pantalla, la rutina común revisa si se han seleccionado las teclas especiales SHIFT, ALT, CLTL y llama a las subrutinas correspondientes a cada uno de ellas; cuando se presionan dichas teclas queda grabada la selección en las variables TEST2, TEST3 Y TEST 4 respectivamente, además se despliegan caracteres especiales a los lados de la pantalla de corrimiento para indicar que fueron presionadas y así continúan hasta que se desactiven. El siguiente paso es llamar a la tabla correspondiente del carácter a desplegar. Si está la tecla “SHIFT” presionada se llama a la tabla de mayúsculas, si no está presionado se llama a la tabla de minúsculas. Antes de desplegar el carácter se revisa si no pertenece al conjunto de los caracteres especiales, los cuales requieren llamar a subrutinas específicas de cada uno de ellos, ya que se requiere desplegar más de una letra dentro de la pantalla de corrimiento. Por ejemplo los caracteres del F1 al F12 requieren, a partir de su código y dentro de la rutina, calcular el número que acompaña a la letra F del 1 al 9. De cada una de estas tablas se van tomando cada uno de los caracteres a desplegar incrementado al contador del programa (PC) la variable SALTO, la cual se incrementa o disminuye de uno en uno (ver rutina programa principal ADELANTE), luego se despliega el carácter en la pantalla y se pasa a la rutina TIEMPO la cual espera la cantidad de tiempo determinado por la velocidad seleccionada al principio.

Este proceso se hace cada vez que se despliega un carácter en la pantalla, incluso si es la primera vez, ya que está generalizado para el despliegue de cada uno de los mismos.

k. Programa principal: En la primera parte del programa se le asignan posiciones de memoria del PIC a cada una de las variables que serán utilizadas. Luego se le asigna valores a la palabra de configuración. Se configura el tipo de oscilador, que en este caso será de cristal, se apaga el temporizador WDT y el de inicio.

A continuación, a cada una de las variables se le da el valor de 0 para evitar que alguna pueda retener información de una corrida del programa anterior y causar problemas dentro del mismo.

A continuación el programa entra en un ciclo WFINT hasta que se presione un botón. Cada vez que se presiona un botón, este genera una interrupción, se lee el puerto para saber qué botón fue presionado; si fue el botón de la derecha se reduce el tiempo entre carácter y carácter desplegado y se aumenta la cantidad de barras gruesas en la escala de la velocidad; si se presiona el botón de la izquierda se va a la rutina MENOS en la cual se aumenta el tiempo de despliegue de caracteres y se disminuye una línea gruesa en la escala; por último al presionar el botón central se desvía el programa a la rutina CORR donde se guarda la velocidad establecida y continua el programa.

Tanto en rutina de aumento de tiempo de despliegue como en la de disminución se revisa que no pasen de los límites establecidos entre uno y ocho como tiempos mínimos y máximos. Al presionar el botón central queda establecido el tiempo de despliegue entre carácter y carácter.

El programa continúa con la rutina ADELANTE, se comienzan a desplegar los caracteres hacia adelante como lo indica el nombre del procedimiento por primera vez, lo continúa haciendo hasta que se presione algún botón. Si se presiona el botón de izquierda se va a rutina INA, la cual despliega los caracteres hacia atrás. Si se presiona el botón central se va a la rutina SCAN, en esta rutina se hace el procedimiento necesario para mandar el carácter a la computadora. Al mandar el carácter desde el teclado especial hacia la computadora entra en un proceso recurrente determinado por la rutina STANBY; Con presionar algún botón dentro de este procedimiento se va a la rutina ADELANTE, INA O SCAN para seguir desplegando los caracteres hacia delante, hacia atrás o enviarlo hacia la computadora respectivamente. Dentro de la rutina INA también es posible, de igual forma que en la rutina ADELANTE, cambiar el sentido del despliegue hacia atrás presionando el botón de la derecha; y mandar el carácter hacia la computadora presionando el botón central, regresando a la rutina STANDBY.

En resumen, en esta parte del programa se establece un ciclo recurrente entre las tres rutinas, desplegando los caracteres hacia delante, hacia atrás o bien envía el carácter a la computadora y se queda en espera que se presione algún botón.

Tabla 9
Variables en el programa

Variable	Descripción
SALTO	Contador para carácter a desplegar
CONT 4	Controlador de la rutina Delay
CONT5	Contador de la rutina RET
RUTINA	Guarda que botón fue presionado
DIGITO	Código a transmitirse serialmente
CONT	Bit a enviar en transmisión serial
TEMP	Dato a transmitirse serialmente
CONT2	Contador de uso general
CONT3	Contador de uso general
PARIDAD	Cuenta el número de unos o ceros para la paridad
DESP	Dato a desplegarse en la pantalla
CONT6	Contador de uso general
TEST	Muestra si se desplegó un carácter especial
TEST2	Muestra que la tecla "SHIFT" esta activa
TEST4	Muestra la tecla "Control" esta activa
DESP2	Almacena la parte baja del dato a desplegarse
VEL	Velocidad de despliegue
GUARDA	Almacena que botón fue presionado
TSTVEL	Indica que la interrupción proviene de cambio de velocidad
VEL2	Almacena la velocidad definitiva del despliegue de caracteres

V. CONCLUSIONES

Se diseñó e implementó un teclado especial para minusválidos. La mayor parte del teclado se realizó con componentes adquiridos localmente con excepción del microcontrolador y el módulo de pantalla de cristal líquido, ambos fueron importados.

Los resultados específicos son:

1. Se cumplió con los objetivos tanto el general como cada uno de los específicos.
2. Se demostró la compatibilidad entre los teclados con conector DIN 180 de 5 pines y con conectores PS2 de 6 pines.
3. Se concluyó que no existe un sistema físico que proporcione ayuda a discapacitados, únicamente programas que ofrecen una ayuda limitada.
4. Se logró establecer la trama de datos del teclado tomando en cuenta que la información recabada tanto en libros como en Internet es ambigua.

VI. RECOMENDACIONES

Tomando en cuenta el trabajo realizado se hacen las siguientes recomendaciones:

1. Según sea la necesidad es posible sustituir los botones por interruptores de otra clase por ejemplo sensores ópticos para otro tipo de discapacidades.
2. Utilizar el puerto USB para la conexión del teclado especial, utilizando este protocolo de comunicación, tomando en cuenta que cada vez existen más dispositivos externos en el mercado que se conectan a este puerto.

VII. BIBLIOGRAFÍA

Chávez Zepeda, Juan José. *Elaboración de Proyectos de Investigación*. Segunda Edición, Guatemala 1994 74pp.

Tisher, Michael. *PC Intern*. 5ft. edition Data Becher edition. U. S. A. 1995 739 pp

8-Bit CMOS EEPROM Microcontroller Manual PIC16F84.

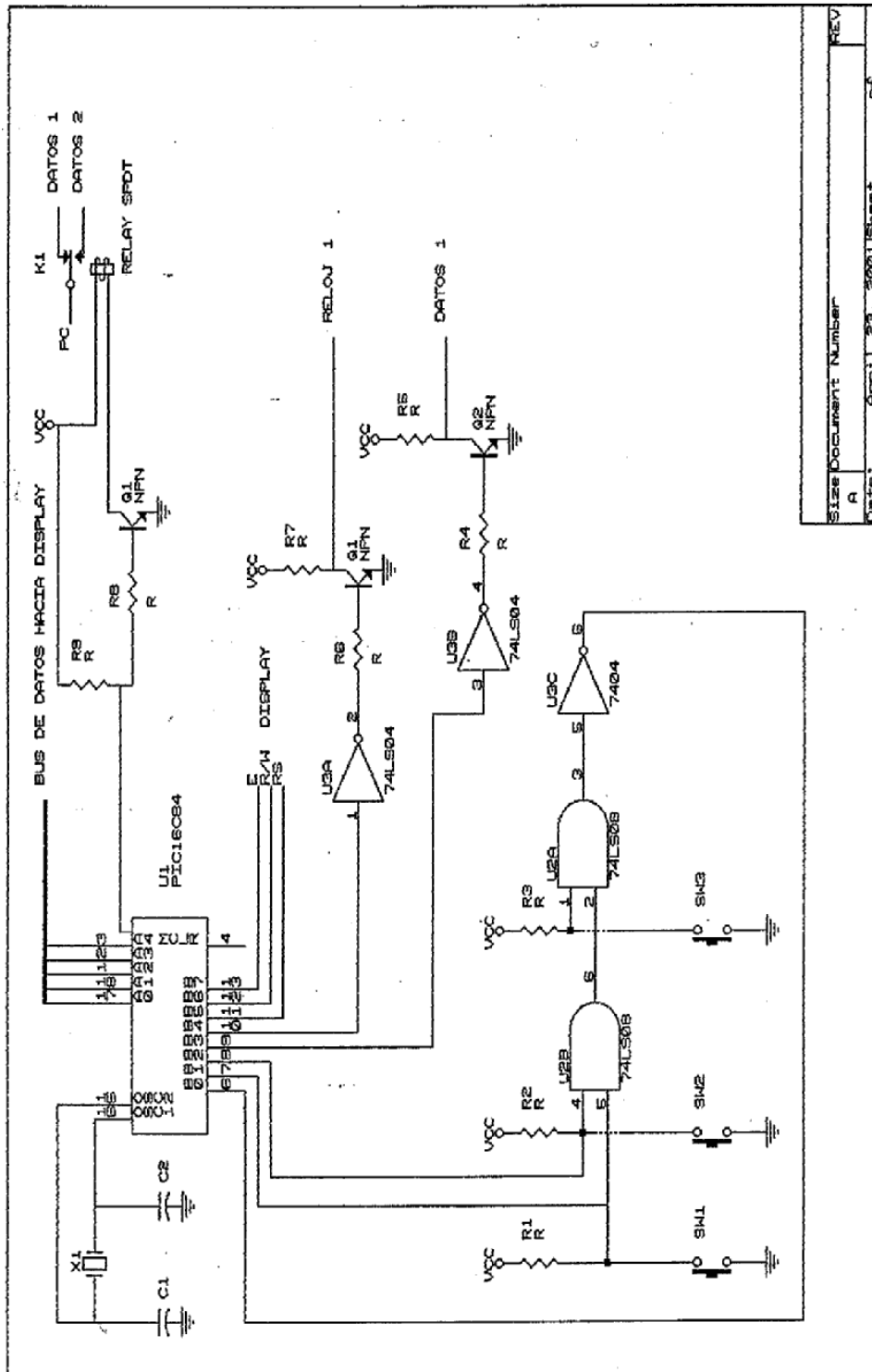
How To Use Hitachi's Built-in Controller Driver LCD-II (HD4480) Dot Matrix LCD Module Manual

[ftp: //mutech.com/mebers/ITU/pc-int/scancode.txt](ftp://mutech.com/mebers/ITU/pc-int/scancode.txt)

[www. Compaq/keyboards/](http://www.Compaq/keyboards/)

APÉNDICE A

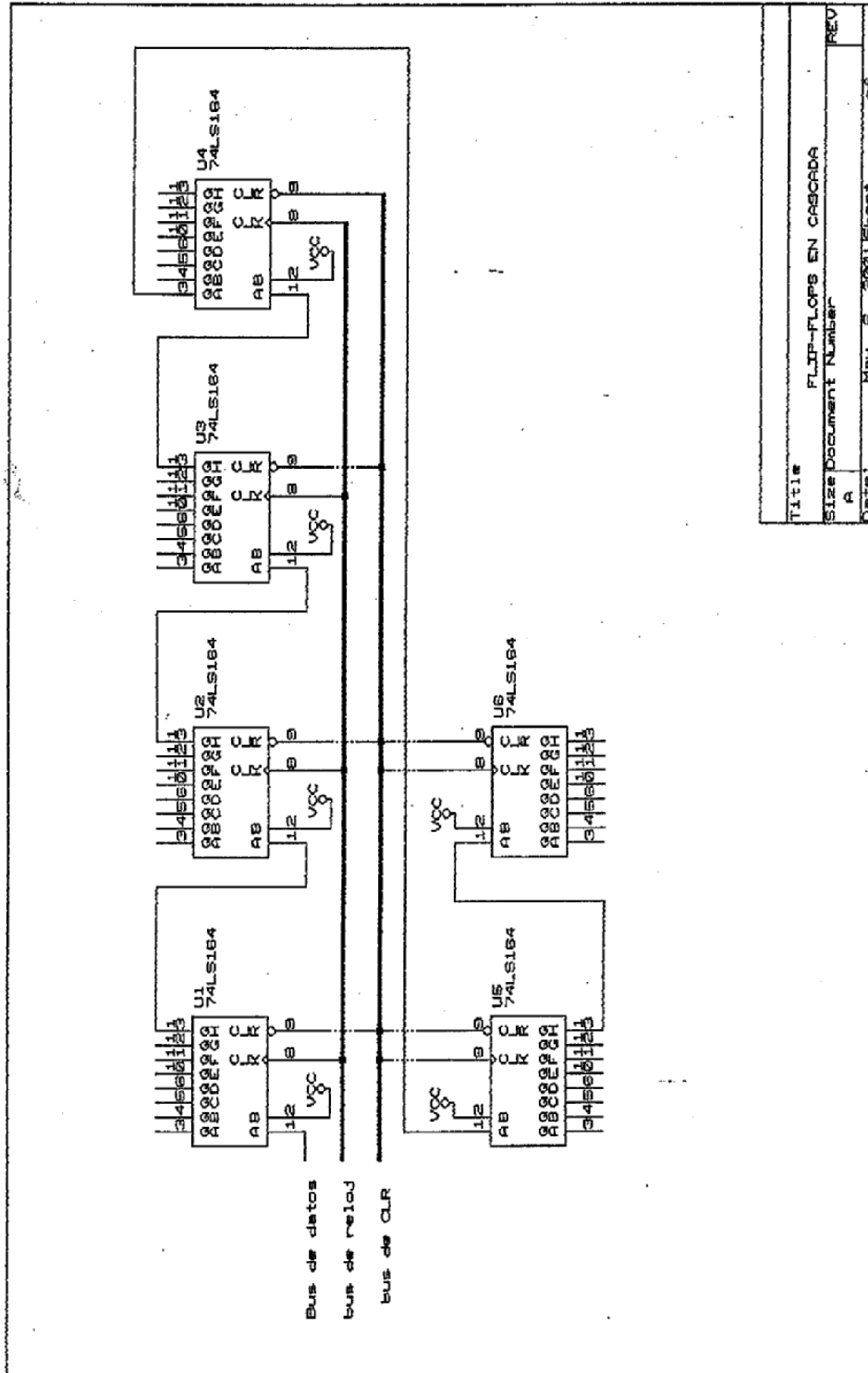
Diagrama del teclado especial



Size	Document Number	REV
A	A	
Date:	April 29, 2001	Sheet of

APÉNDICE B

Flip-Flops en cascada



Title	FLIP-FLOPS EN CASCAIDA
Size Document Number	A
REV	
Date:	May 9, 2001 Sheet 02

APÉNDICE C

Programa

```
LIST P=16C84
INCLUDE C:\PIC\P16C84.INC

SALTO EQU H'000C' ; COMIENZO DE DECLARACIÓN DE VARIABLES
CONT4 EQU SALTO+1 ; CONTADOR PARA DELAY
CONT5 EQU SALTO+2 ; CONTADOR RUTINA RET
RUTINA EQU SALTO+3 ; GUARDA QUE BOTÓN FUE PRESIONADO
DIGITO EQU SALTO+4 ; CÓDIGO A TRANSMITIRSE SERIALMENTE
CONT EQU SALTO+5 ; BIT A ENVIAR EN TRANS. SERIAL
TEMP EQU SALTO+6 ; DATO A TRASMITIRSE SERIALMENTE
CONT2 EQU SALTO+7
CONT3 EQU SALTO+8
PARIDAD EQU SALTO+9 ; NÚMERO DE UNOS PARA PARIDAD
DESP EQU SALTO+10 ; CÓDIGO A DESPLEGARSE EN PANTALLA
CONT6 EQU SALTO+11
TEST EQU SALTO+12 ; MUESTRA SI SE DESPLEGÓ CAR ESPECIAL
TEST2 EQU SALTO+13 ; MOSTRAR QUÉ SHIFT ESTA ACTIVO
TEST3 EQU SALTO+14 ; MOSTRAR QUÉ ALT ESTA ACTIVO
TEST4 EQU SALTO+15 ; MOSTRAR QUÉ CLT ESTA PRENDIDO
DESP2 EQU SALTO+16 ; PARTE BAJA DE CÓDIGO A DESPLEGARSE
DESP3 EQU SALTO+17 ; CONTADOR PARA RUTINA DE F1-F9
VEL EQU SALTO+18 ; VELOCIDAD ENTRE CAR Y CAR
GUARDA EQU SALTO+19 ; ALMACENA QUÉ BOTÓN FUE PRESIONADO
TSTVEL EQU SALTO+20 ; INDICA QUE LA INTERR. PROVIENE DE VEL
VEL2 EQU SALTO+21 ; FIN DE DECLARACIÓN DE VARIABLES
ORG 2007
DW _XT_OSC & _WDT_OFF & _PWRTE_OFF

ORG 7FFF
DW 84

ORG 0
CLRF SALTO
CLRF CONT4 ; LIMPIAR VARIABLES
CLRF RUTINA
GOTO DECVAR

ORG 4
NOP
BCF INTCON,INTF
BTFSS TSTVEL,0 ; ES LA INTERRUPCIÓN PARA VELOCIDAD?
GOTO NORMAL
MOVF PORTB,W ; RUTINA PARA VELOCIDAD
MOVWF GUARDA ; ALMACENA LO LEÍDO EN EL PUERTO EN GUARDA
BTFSS GUARDA,1
RETFIE
BTFSS GUARDA,2
RETFIE
```

```

        BCF  GUARDA, 3      ; INDICA QUE LA VELOCIDAD ES CORRECTA
        RETFIE
NORMAL  MOVF  PORTB, W
        MOVWF RUTINA
        BTFSS RUTINA, 1
        RETFIE
        BTFSS RUTINA, 2
        RETFIE
        BCF  RUTINA, 3      ; INDICA QUE EL BOTÓN CENTRAL FUE
PRESIONADO
        RETFIE
DECVAR
        CLRF  CONT5
        CLRF  RUTINA
        CLRF  DIGITO
        CLRF  CONT      ; LIMPIAR VARIABLES
        CLRF  TEMP
        CLRF  CONT2
        CLRF  CONT3
        CLRF  DESP
        CLRF  TEST
        CLRF  TEST2
        CLRF  TEST3
        CLRF  TEST4
        CLRF  VEL
        CLRF  VEL2
        CLRF  GUARDA
        CLRF  TSTVEL
        BSF  STATUS, RP0      ; PASA A LA PÁGINA 1
        BCF  OPTION_REG, T0CS ; HABILITAR EL TIMER
        CLRF  TRISA      ; PUERTO A COMO SALIDA
        MOVLW B'00000111'
        MOVWF TRISB      ; <0,1,2> COMO ENTRADAS (BOTONES)
        BCF  STATUS, RP0      ; PASA A LA PÁGINA 0
        CLRF  TMR0      ; LIMPIAR EL TEMPORIZADOR
        MOVLW B'01111000'
        MOVWF PORTB      ; RELEE EN 0 Y DEMÁS SALIDAS EN 1
        MOVLW H'FF'
        MOVWF RUTINA      ; RUTINA GUARDA EN 1
        MOVWF GUARDA
        BCF  GUARDA, 0      ; LIMPIAR INTERRUPCIONES
        CLRF  PORTA
        CALL DELAY
        CLRF  INTCON      ; DESHABILITAR LAS INTERRUPCIONES
        BSF  INTCON, GIE
        BSF  INTCON, INTE      ; HABILITAR LAS INTERRUPCIONES

UNO
        CALL  RET
        INCF  CONT4, F
        MOVF  CONT4, W      ; 1 MS
        SUBLW .15
        BTFSS STATUS, Z
        GOTO  UNO      ; ESTA RUTINA ESPERA 15 MS

```

```

        CLRF  CONT4
        MOVLW B'00000011'
        MOVWF DESP2
        CALL  MEDIO

DOS
        CALL  RET
        INCF  CONT4,F
        MOVF  CONT4,W
        SUBLW .5
        BTFSS STATUS,Z
        GOTO  DOS ;ESTA RUTINA ESPERA 5MS
        CLRF  CONT4
        MOVLW B'00000011'
        MOVWF DESP2
        CALL  MEDIO

TRES
        CALL  DELAY2
        INCF  CONT4,F
        MOVF  CONT4,W
        SUBLW .4
        BTFSS STATUS,Z
        GOTO  TRES ; ESTA RUTINA ESPERA 100 MICOSEG
        CLRF  CONT4
        MOVLW B'00000011'
        MOVWF DESP2
        CALL  MEDIO
        CALL  RET
        MOVLW B'00000010' ; CAMBIAR A 4 BITS
        MOVWF DESP2
        CALL  MEDIO
        MOVLW B'01000010' ; FUCTION SET
        CALL  ENVIAR
        MOVLW B'11100000' ; DISPLAY ON
        CALL  ENVIAR
        MOVLW B'00010000' ; CLEAR DISPLAY
        CALL  ENVIAR
        MOVLW B'01100000' ; ENTRY MODE SET
        CALL  ENVIAR
        MOVLW .5
        MOVWF VEL

LETRERO
        MOVLW B'01100101' ; V
        CALL  ENVIAR2
        MOVLW B'01010100' ; E
        CALL  ENVIAR2
        MOVLW B'11000100' ; L
        CALL  ENVIAR2
        MOVLW B'11110100' ; O
        CALL  ENVIAR2
        MOVLW B'00110100' ; C
        CALL  ENVIAR2
        MOVLW B'10010100' ; I

```

```

CALL ENVIAR2
MOVLW B'01000100' ; D
CALL ENVIAR2
MOVLW B'00010100' ; A
CALL ENVIAR2
MOVLW B'01000100' ; D
CALL ENVIAR2
MOVLW B'11110011' ; ?
CALL ENVIAR2

AVISO
BSF TSTVEL,0
BTFSS GUARDA,0 ; SI HAY INTERRUPCIÓN SE VA
GOTO AVISO ; A LA RUTINA DE VELOCIDAD
MOVLW B'11111111'
MOVWF GUARDA
BCF TSTVEL,0
MOVLW B'00100000' ; REGRESAR A POSICIÓN 1 EL DISPLAY
CALL ENVIAR
CALL RET
CALL RET
MOVLW B'11010100' ; M
CALL ENVIAR2
MOVLW B'10010100' ; I
CALL ENVIAR2
MOVLW B'11100100' ; N
CALL ENVIAR2
MOVLW B'00000010' ; ESPACIO EN BLANCO
CALL ENVIAR2
MOVLW B'11111111' ; ±
CALL ENVIAR2
MOVLW B'11111111' ; ±
CALL ENVIAR2
MOVLW B'11111111' ; ±
CALL ENVIAR2
MOVLW B'11111111' ; ±
CALL ENVIAR2
MOVLW B'11000111' ; |
CALL ENVIAR2
MOVLW B'11000111' ; |
CALL ENVIAR2
MOVLW B'11000111' ; |
CALL ENVIAR2
MOVLW B'00000010' ; ESPACIO EN BLANCO
CALL ENVIAR2
MOVLW B'11010100' ; M
CALL ENVIAR2
MOVLW B'00010100' ; A
CALL ENVIAR2
MOVLW B'10000101' ; X
CALL ENVIAR2
MOVLW B'10001000' ; PONER EL CURSOR EN POSICIÓN 9
CALL ENVIAR

```

```

WFINT
    CALL DELAY
    CALL DELAY
    BSF TSTVEL,0
    BTFSC GUARDA,1 ; SE PRESIONÓ BOTÓN DERECHO?
    GOTO MENOS
    MOVF VEL,W
    SUBLW B'00000001' ; VER SI YA LLEGÓ A 1
    BTFSC STATUS,Z
    GOTO CORR
    DECF VEL,F ; REDUCIR LA VAR VELOCIDAD
    MOVLW B'11111111' ; ±
    CALL ENVIAR2
    BSF GUARDA,1
MENOS
    BTFSC GUARDA,2 ; REDUCIR LA VAR VELOCIDAD
    GOTO CORR
    MOVF VEL,W
    SUBLW B'00001000' ; VER SI YA LLEGÓ A 8
    BTFSC STATUS,Z
    GOTO CORR
    INCF VEL,F
    MOVLW B'00000001' ; CORRER LA POSICIÓN DEL CURSOR
    CALL ENVIAR
    MOVLW B'11000111' ; |
    CALL ENVIAR2
    MOVLW B'00000001' ; CORRER LA POSICIÓN DEL CURSOR
    CALL ENVIAR
    BSF GUARDA,2
CORR
    MOVF VEL,W
    MOVWF VEL2
    BTFSC GUARDA,3 ; SE PRESIONÓ EL BOTÓN CENTRAL?
    GOTO WFINT
    BCF TSTVEL,0
    GOTO ADELANTE
STANDBY
    BCF TEST,0 ; ESPERA A QUE SE PRESIONE TECLA
    BTFSS RUTINA,1
    GOTO ADELANTE
    BTFSS RUTINA,2
    GOTO INA
    BTFSS RUTINA,3
    GOTO SCAN
    GOTO STANDBY
SCAN
    MOVF DESP,W
    SUBLW B'00101011' ; ES EL CÓDIGO DE SHIFT?
    BTFSC STATUS,Z
    GOTO NOSFT
    MOVF DESP,W
    SUBLW B'00001100' ; ES EL CÓDIGO DE ALT?
    BTFSC STATUS,Z
    GOTO NOALT
    MOVF DESP,W
    SUBLW B'10111100' ; ES EL CÓDIGO DE CTRL?

```

```

        BTFSC STATUS,Z
        GOTO NOCTRL
        CALL SERIAL
EMP2    MOVLW B'01000000' ; ASIGNAR TIEMPO ENTRE SCAN CODES
        MOVWF CONT4
EMPEZAR
        CALL RET
        DECF CONT4,F
        BTFSS STATUS,Z
        GOTO EMPEZAR
        MOVLW H'F0' ; 1er SCAN CODE DE TECLA LIBERADA
        CALL SERIAL2
        CALL SERIAL ; 2do SCAN CODE DE TECLA LIBERADA
        BCF PORTA,4 ; DESACTIVAR EL RELAY
        GOTO STANDBY
NOSFT
        INCF TEST2,F
        BTFSC TEST2,0 ; ES PAR TEST2?
        CALL SERIAL ; SI ES IMPAR
        BTFSC TEST2,0
        GOTO STANDBY
        GOTO EMP2 ; SI ES PAR
NOALT
        INCF TEST3,F ; SIMILAR A LA RUTINA NOSFT
        BTFSC TEST3,0
        CALL SERIAL
        BTFSC TEST3,0
        GOTO STANDBY
        GOTO EMP2
NOCTRL
        INCF TEST4,F ; SIMILAR A LA RUTINA NOSFT
        BTFSC TEST4,0
        CALL SERIAL
        BTFSC TEST4,0
        GOTO STANDBY
        GOTO EMP2
ADELANTE
        CALL COMUN
        BTFSS RUTINA,2 ; HUBO INTERRUPCIÓN?
        GOTO INA
        BTFSS RUTINA,3
        GOTO SCAN
        MOVF VEL2,W ; VEL = VEL2
        MOVWF VEL
        MOVLW .1
        BTFSS PORTB,1
        MOVWF VEL ; SI SE MANT.TECLA PRESIONADA CAMBIA
VELOCIDAD
        INCF SALTO,F
        MOVF SALTO,W
        SUBLW .76 ; NÚMERO DE CARACTERES
        BTFSC STATUS,Z
        CLRF SALTO

```

```

GOTO ADELANTE

INA
CALL COMUN
BTFSS RUTINA,1 ; HUBO INTERRUPCIÓN?
GOTO ADELANTE
BTFSS RUTINA,3
GOTO SCAN
MOVF VEL2,W
MOVWF VEL
MOVLW .1
BTFSS PORTB,2 ; FUE PRESIONADO EL BOTÓN DERECHO
MOVWF VEL ; CAMBIO DE VELOCIDAD
DECFSALTO,F
MOVF SALTO,W
SUBLW .75 ; NÚMERO DE CARACTERES -1
MOVLW .75
BTFSS STATUS,C
MOVWF SALTO
GOTO INA

COMUN
MOVLW B'00010000' ; LIMPIAR EL DISPLAY
CALL ENVIAR
CALL RET
CALL RET
CALL RET
BTFSS TEST2,0 ; VER SI SHIFT ESTA ENCENDIDO
GOTO PRBA
MOVLW B'00001000' ; PONER EN POSICIÓN 1 EL CURSOR
CALL ENVIAR
MOVLW B'00101011' ; INDICAR SI SHIFT ESTÁ ACTIVO
CALL ENVIAR2
PRBA BTFSS TEST3,0 ; VER SI ALT ESTÁ ACTIVO
GOTO PRBA2
MOVLW B'10111000' ; PONER EN POSICIÓN 12 EL CURSOR
CALL ENVIAR
MOVLW B'00001100' ; INDICAR SI ALT ESTÁ ACTIVO
CALL ENVIAR2
PRBA2 BTFSS TEST4,0 ; VER SI CTRL ESTÁ ACTIVO
GOTO PRBA3
MOVLW B'00101000' ; PONER EN POSICION 3 EL CURSOR
CALL ENVIAR
MOVLW B'10111100' ; INDICAR SI CTRL ESTÁ ACTIVO
CALL ENVIAR2
PRBA3 MOVLW B'01111000' ; PONER EN POSICIÓN 8 EL CURSOR
CALL ENVIAR
CALL RET
BTFSS TEST2,0 ; REVISAR SI SHIFT ESTÁ ACTIVO
CALL TABLA3 ; TABLA DE MAYÚSCULAS
BTFSS TEST2,0
CALL TABLA
MOVWF DESP

```

```

SUBLW B'00000010' ; REVISAR SI ES CÓDIGO DE ESPACIO
BTFSC STATUS,Z
CALL ESPACIO
MOVF DESP,W
SUBLW B'00111010' ; REVISAR SI ES CÓDIGO DE ENTER
BTFSC STATUS,Z
CALL ENTER
MOVF DESP,W
SUBLW B'00101011' ; REVISAR SI ES CÓDIGO DE SHIFT
BTFSC STATUS,Z
CALL MAY
MOVF DESP,W
SUBLW B'00001100' ; REVISAR SI ES CÓDIGO DE ALT
BTFSC STATUS,Z
CALL TALT
MOVF DESP,W
SUBLW B'10111100' ; REVISAR SI ES CÓDIGO DE CTRL
BTFSC STATUS,Z
CALL CTRL
MOVF DESP,W
SUBLW B'00111101' ; REVISAR SI ES CÓDIGO DE ESC
BTFSC STATUS,Z
CALL ESC
MOVF DESP,W
SUBLW B'01011011' ; REVISAR SI ES CÓDIGO DE TAB
BTFSC STATUS,Z
CALL TAB
MOVF DESP,W
SUBLW B'01001010' ; REVISAR SI ES CODIOG DE BACKSPACE
BTFSC STATUS,Z
CALL BSP
MOVF DESP,W
SUBLW B'01001100' ; REVISAR SI ES CÓDIGO DE PgUp
BTFSC STATUS,Z
CALL PGUP
MOVF DESP,W
SUBLW B'10011100' ; REVISAR SI ES CÓDIGO DE PgDn
BTFSC STATUS,Z
CALL PGDW
MOVF DESP,W
SUBLW B'10001100' ; REVISAR SI ES CÓDIGO DE END
BTFSC STATUS,Z
CALL TEND
MOVF DESP,W
SUBLW B'01001101' ; REVISAR SI ES CÓDIGO DE INS
BTFSC STATUS,Z
CALL INS
MOVF DESP,W
SUBLW B'01111100' ; REVISAR SI ES CÓDIGO DE DEL
BTFSC STATUS,Z
CALL DEL
MOVF DESP,W
SUBLW B'10011011' ; REVISAR SI ES CÓDIGO DE HOME
BTFSC STATUS,Z

```

```

CALL HOME
MOVF  DESP,W
SUBLW B'10001010' ; REVISAR SI ES CÓDIGO DE UP
BTFSC STATUS,Z
CALL  UP
MOVF  DESP,W
SUBLW B'10101101' ; REVISAR SI ES CÓDIGO DE DOWN
BTFSC STATUS,Z
CALL  DN
MOVF  DESP,W
SUBLW B'00001110' ; REVISAR SI ES CÓDIGO DE F10
BTFSC STATUS,Z
CALL  F10
MOVF  DESP,W
SUBLW B'00011110' ; REVISAR SI ES CÓDIGO DE F11
BTFSC STATUS,Z
CALL  F11
MOVF  DESP,W
SUBLW B'00101110'
BTFSC STATUS,Z
CALL  F12
MOVF  DESP,W
MOVWF DESP2 ; HACER UNA COPIA DE DESP EN DESP2
MOVWF DESP3
SWAPF DESP2,F
BCF   STATUS,C
RRF   DESP2,F
BCF   STATUS,C
RRF   DESP2,F
BCF   STATUS,C
RRF   DESP2,F
BCF   STATUS,C
RRF   DESP2,F
MOVF  DESP2,W
SUBLW B'00000000'
BTFSC STATUS,Z
CALL  F1F12
MOVF  DESP,W
BTFSS TEST,0
CALL  ENVIAR2 ; DESPEGAR EL CARACTER
CLRF  TEST
MOVF  VEL,W ; VELOCIDAD
MOVWF CONT6
TIEMPO NOP
CALL  DELAY
DECF  CONT6,F
BTFSS STATUS,Z ; RUTINA PARA ESPERAR 250 MS
GOTO  TIEMPO
RETURN

DELAY
NOP
NOP
NOP

```

```

NOP
LOOP
    DECF  CONT5,F
    BTFSS STATUS,Z
    GOTO  LOOP
    DECF  CONT4,F
    BTFSS STATUS,Z
    GOTO  DELAY
    RETURN

SERIAL
    BSF   RUTINA,3
    CALL  TABLA2
SERIAL2
    BSF   PORTA,4           ; ACTIVAR EL RELAY
    CALL  RET
    CALL  RET
    CALL  RET
    CALL  RET
    CALL  RET
    BCF   PORTB,3
    MOVWF DIGITO
INICIO
    CALL  DELAY2
    INCF  CONT,F
    BCF   PORTB,4           ; PONER 0 EN CLK
    MOVF  CONT,W
    SUBLW .1
    BTFSC STATUS,Z
    BCF   PORTB,3           ; START BIT
    MOVLW B'11111111'      ; PARIDAD
    MOVWF TEMP
    BTFSC PARIDAD,0        ; SI PARIDAD ES IMPAR TEMP=0
    CLRF  TEMP
    MOVF  CONT,W
    SUBLW .9
    BTFSC STATUS,Z
    GOTO  SEGUIR           ; SALTAR SI LLEGÓ EL 9no BIT
    MOVLW B'11111111'
    MOVWF TEMP
    BTFSS STATUS,C
    GOTO  SEGUIR           ; SALTAR SI LLEGÓ EL 10 BIT
    MOVF  DIGITO,W
    BTFSC DIGITO,0
    INCF  PARIDAD,F        ; SI HAY UN 1 INCREMENTA PARIDAD
    BCF   STATUS,C
    RRF   DIGITO,F
    MOVWF TEMP
SEGUIR
    CALL  DELAY2
    BSF   PORTB,4           ; PONER UN 1 EN CLK
    BTFSC TEMP,0
    BSF   PORTB,3           ; SACAR EL DATO SERIAL
    BTFSS TEMP,0

```

```

BCF    PORTB, 3
MOVF   CONT, W
SUBLW  .11
BTFSS  STATUS, Z
GOTO   INICIO
CALL   RET
CLRF   TMR0
CLRF   CONT
CLRF   CONT2
CLRF   PARIDAD
RETURN

```

DELAY2

```

NOP
MOVF   TMR0, W
SUBLW  .26
BTFSC  STATUS, C
GOTO   DELAY2
CLRF   TMR0
RETURN

```

RET

```

DECF   CONT5, F
BTFSS  STATUS, Z
GOTO   RET
RETURN

```

ENVIAR

```

MOVWF  DESP                ; GUARDAR W EN DESP
SWAPF  DESP, W
MOVWF  DESP2               ; GUARDAR LA PARTE BAJA DEL DATO A ENVIAR
BCF    DESP2, 4
BCF    DESP, 4
CALL   RET
MOVLW  B'00011000'        ; PONER 0 EN RS Y R/W
MOVWF  PORTB
BSF    PORTB, 7           ; PONER UN 1 EN ENABLE
MOVF   DESP, W
MOVWF  PORTA              ; DESPLEGAR LA PARTE ALTA DEL DATO
MOVLW  B'01111000'
BCF    PORTB, 7           ; PONER UN 0 EN ENABLE
MOVWF  PORTB              ; R/W = 1  RS=1
CLRF   PORTA              ; PONER 0 EN DATOS

```

MEDIO

```

BCF    DESP2, 4
MOVLW  B'00011000'
MOVWF  PORTB
BSF    PORTB, 7
MOVF   DESP2, W
MOVWF  PORTA              ; DESPLEGAR LA PARTE BAJA DEL DATO
MOVLW  B'01111000'
BCF    PORTB, 7
MOVWF  PORTB
CLRF   PORTA
RETURN

```

ENVIAR2

```

MOVWF DESP           ; GUARDAR W EN DESP
SWAPF DESP,W
MOVWF DESP2
BCF  DESP,4
BCF  DESP2,4
CALL RET
MOVLW B'00111000'   ; PONER 1 EN RS Y 0 EN R/W
MOVWF PORTB
BSF  PORTB,7        ; PONER 1 EN ENABLE
MOVF  DESP,W
MOVWF PORTA         ; DESPLEGAR LA PARTE ALTA DE DATO
MOVLW B'01011000'
BCF  PORTB,7        ; ENABLE =0
MOVWF PORTB         ; R/W=1 RS=0
CLRF  PORTA
MOVLW B'00111000'
MOVWF PORTB
BSF  PORTB,7
MOVF  DESP2,W
MOVWF PORTA         ; DESPLEGA LA PARTE BAJA DEL DATO
MOVLW B'01011000'
BCF  PORTB,7
MOVWF PORTB
CLRF  PORTA
RETURN

```

ENTER

```

MOVLW B'00000001'   ; REGRESAR EL CURSOR
CALL  ENVIAR
MOVLW B'01010100'   ; ESCRIBIR E
CALL  ENVIAR2
MOVLW B'11100100'   ; ESCRIBIR N
CALL  ENVIAR2
MOVLW B'01000101'   ; ESCRIBIR T
CALL  ENVIAR2
BSF  TEST,0
RETURN

```

ESPACIO

```

MOVLW B'00000001'   ; REGRESAR EL CURSOR
CALL  ENVIAR
MOVLW B'00110101'   ; ESCRIBIR S
CALL  ENVIAR2
MOVLW B'00000101'   ; ESCRIBIR P
CALL  ENVIAR2
MOVLW B'00110100'   ; ESCRIBIR C
CALL  ENVIAR2
BSF  TEST,0
RETURN

```

MAY

```

MOVLW B'00000001'   ; REGRESAR EL CURSOR
CALL  ENVIAR

```

```

        MOVLW B'00110101'      ; ESCRIBIR S
        CALL  ENVIAR2
        MOVLW B'01100100'      ; ESCRIBIR H
        CALL  ENVIAR2
        MOVLW B'01000101'      ; ESCRIBIR T
        CALL  ENVIAR2
        BSF   TEST,0
        MOVLW B'00101011'
        MOVWF DESP
        RETURN

TALT
        MOVLW B'00000001'      ; REGRESAR EL CURSOR
        CALL  ENVIAR
        MOVLW B'00010100'      ; ESCRIBIR A
        CALL  ENVIAR2
        MOVLW B'11000110'      ; ESCRIBIR l
        CALL  ENVIAR2
        MOVLW B'01000111'      ; ESCRIBIR t
        CALL  ENVIAR2
        BSF   TEST,0
        MOVLW B'00001100'
        MOVWF DESP
        RETURN

CTRL
        MOVLW B'00000001'      ; REGRESAR EL CURSOR
        CALL  ENVIAR
        MOVLW B'00110100'      ; ESCRIBIR C
        CALL  ENVIAR2
        MOVLW B'01000111'      ; ESCRIBIR t
        CALL  ENVIAR2
        MOVLW B'11000110'      ; ESCRIBIR l
        CALL  ENVIAR2
        BSF   TEST,0
        MOVLW B'10111100'
        MOVWF DESP
        RETURN

ESC
        MOVLW B'00000001'      ; REGRESAR EL CURSOR
        CALL  ENVIAR
        MOVLW B'01010100'      ; ESCRIBIR E
        CALL  ENVIAR2
        MOVLW B'00110111'      ; ESCRIBIR s
        CALL  ENVIAR2
        MOVLW B'00110110'      ; ESCRIBIR c
        CALL  ENVIAR2
        BSF   TEST,0
        RETURN

TAB
        MOVLW B'00000001'      ; REGRESAR EL CURSOR
        CALL  ENVIAR
        MOVLW B'01000101'      ; ESCRIBIR T
        CALL  ENVIAR2

```

```

        MOVLW B'00010110'      ; ESCRIBIR a
        CALL  ENVIAR2
        MOVLW B'00100110'      ; ESCRIBIR b
        CALL  ENVIAR2
        BSF   TEST,0
        RETURN

BSP
        MOVLW B'00000001'      ; REGRESAR EL CURSOR
        CALL  ENVIAR
        MOVLW B'11110111'      ; ESCRIBIR <
        CALL  ENVIAR2
        MOVLW B'00001011'      ; ESCRIBIR -
        CALL  ENVIAR2
        MOVLW B'00001011'      ; ESCRIBIR -
        CALL  ENVIAR2
        BSF   TEST,0
        RETURN

PGUP
        MOVLW B'00000001'      ; REGRESAR EL CURSOR
        CALL  ENVIAR
        MOVLW B'00000101'      ; ESCRIBIR P
        CALL  ENVIAR2
        MOVLW B'01111110'      ; ESCRIBIR g
        CALL  ENVIAR2
        MOVLW B'01010101'      ; ESCRIBIR U
        CALL  ENVIAR2
        MOVLW B'00000111'      ; ESCRIBIR p
        CALL  ENVIAR2
        BSF   TEST,0
        RETURN

PGDW
        MOVLW B'00000001'      ; REGRESAR EL CURSOR
        CALL  ENVIAR
        MOVLW B'00000101'      ; ESCRIBIR P
        CALL  ENVIAR2
        MOVLW B'01111110'      ; ESCRIBIR g
        CALL  ENVIAR2
        MOVLW B'01000100'      ; ESCRIBIR D
        CALL  ENVIAR2
        MOVLW B'11100110'      ; ESCRIBIR n
        CALL  ENVIAR2
        BSF   TEST,0
        RETURN

TEND
        MOVLW B'00000001'      ; REGRESAR EL CURSOR
        CALL  ENVIAR
        MOVLW B'01010100'      ; ESCRIBIR E
        CALL  ENVIAR2
        MOVLW B'11100110'      ; ESCRIBIR n
        CALL  ENVIAR2
        MOVLW B'01000110'      ; ESCRIBIR d
        CALL  ENVIAR2
        BSF   TEST,0

```

```

RETURN
INS
MOVLW B'00000001' ; REGRESAR EL CURSOR
CALL ENVIAR
MOVLW B'10010100' ; ESCRIBIR I
CALL ENVIAR2
MOVLW B'11100110' ; ESCRIBIR n
CALL ENVIAR2
MOVLW B'00110111' ; ESCRIBIR s
CALL ENVIAR2
BSF TEST,0
RETURN
DEL
MOVLW B'00000001' ; REGRESAR EL CURSOR
CALL ENVIAR
MOVLW B'01000100' ; ESCRIBIR D
CALL ENVIAR2
MOVLW B'01010110' ; ESCRIBIR e
CALL ENVIAR2
MOVLW B'11000110' ; ESCRIBIR l
CALL ENVIAR2
BSF TEST,0
RETURN
HOME
MOVLW B'00000001' ; REGRESAR EL CURSOR
CALL ENVIAR
MOVLW B'10000100' ; ESCRIBIR H
CALL ENVIAR2
MOVLW B'11110110' ; ESCRIBIR o
CALL ENVIAR2
MOVLW B'11010110' ; ESCRIBIR m
CALL ENVIAR2
MOVLW B'01010110' ; ESCRIBIR e
CALL ENVIAR2
BSF TEST,0
RETURN
UP
MOVLW B'00000001' ; REGRESAR EL CURSOR
CALL ENVIAR
MOVLW B'01010101' ; ESCRIBIR U
CALL ENVIAR2
MOVLW B'00001111' ; ESCRIBIR p
CALL ENVIAR2
BSF TEST,0
RETURN
DN
MOVLW B'00000001' ; REGRESAR EL CURSOR
CALL ENVIAR
MOVLW B'01000100' ; ESCRIBIR D
CALL ENVIAR2
MOVLW B'11100110' ; ESCRIBIR n
CALL ENVIAR2
BSF TEST,0
RETURN

```

```

F1F12
    MOVLW B'00000001'      ; REGRESAR EL CURSOR
    CALL  ENVIAR
    MOVLW B'01100100'      ; ESCRIBIR F
    CALL  ENVIAR2
    MOVF  DESP3,W
    ADDLW B'00000011'      ; AGREGAR PARA NÚMEROS
    CALL  ENVIAR2
    BSF   TEST,0
    RETURN

F10
    MOVLW B'00000001'      ; REGRESAR EL CURSOR
    CALL  ENVIAR
    MOVLW B'01100100'      ; ESCRIBIR F
    CALL  ENVIAR2
    MOVLW B'00010011'      ; ESCRIBIR 1
    CALL  ENVIAR2
    MOVLW B'00000011'      ; ESCRIBIR 0
    CALL  ENVIAR2
    BSF   TEST,0
    RETURN

F11
    MOVLW B'00000001'      ; REGRESAR EL CURSOR
    CALL  ENVIAR
    MOVLW B'01100100'      ; ESCRIBIR F
    CALL  ENVIAR2
    MOVLW B'00010011'      ; ESCRIBIR 1
    CALL  ENVIAR2
    MOVLW B'00010011'      ; ESCRIBIR 1
    CALL  ENVIAR2
    BSF   TEST,0
    RETURN

F12
    MOVLW B'00000001'      ; REGRESAR EL CURSOR
    CALL  ENVIAR
    MOVLW B'01100100'      ; ESCRIBIR F
    CALL  ENVIAR2
    MOVLW B'00010011'      ; ESCRIBIR 1
    CALL  ENVIAR2
    MOVLW B'00100011'      ; ESCRIBIR 2
    CALL  ENVIAR2
    BSF   TEST,0
    RETURN

    ORG 300

TABLA2
    MOVLW 3
    MOVWF PCLATH           ; QUEDARSE EN 3XX
    MOVF  SALTO,W         ; TABLA DE SCAN CODES
    ADDWF PCL,F
    RETLW H'05' ;F1
    RETLW H'06' ;F2
    RETLW H'04' ;F3

```

```
RETLW H'0C' ;F4
RETLW H'03' ;F5
RETLW H'0B' ;F6
RETLW H'83' ;F7
RETLW H'0A' ;F8
RETLW H'01' ;F9
RETLW H'09' ;F10
RETLW H'78' ;F11
RETLW H'07' ;F12
RETLW H'76' ;ESC
RETLW H'16' ;1
RETLW H'1E' ;2
RETLW H'26' ;3
RETLW H'25' ;4
RETLW H'2E' ;5
RETLW H'36' ;6
RETLW H'3D' ;7
RETLW H'3E' ;8
RETLW H'46' ;9
RETLW H'45' ;0
RETLW H'4E' ; -
RETLW H'55' ; =
RETLW H'5D' ; \
RETLW H'66' ;BACKSPACE
RETLW H'0D' ;TAB
RETLW H'15' ;q
RETLW H'1D' ;w
RETLW H'24' ;e
RETLW H'2D' ;r
RETLW H'2C' ;t
RETLW H'35' ;y
RETLW H'3C' ;u
RETLW H'43' ;i
RETLW H'44' ;o
RETLW H'4D' ;p
RETLW H'54' ;[
RETLW H'5B' ;]
RETLW H'1C' ;a
RETLW H'1B' ;s
RETLW H'23' ;d
RETLW H'2B' ;f
RETLW H'34' ;g
RETLW H'33' ;h
RETLW H'3B' ;j
RETLW H'42' ;k
RETLW H'4B' ;l
RETLW H'4C' ;punto y coma
RETLW H'52' ;'
RETLW H'5A' ;ENTER
RETLW H'12' ;SHIFT
RETLW H'1A' ;z
RETLW H'22' ;x
RETLW H'21' ;c
RETLW H'2A' ;v
```

```

RETLW H'32' ;b
RETLW H'31' ;n
RETLW H'3A' ;m
RETLW H'41' ; ,
RETLW H'49' ; .
RETLW H'4A' ; /
RETLW H'14' ; CTRL
RETLW H'29' ; SPC
RETLW H'11' ; ALT
RETLW H'6C' ; Home
RETLW H'7D' ; PgUp
RETLW H'69' ; End
RETLW H'7A' ; PgDw
RETLW H'70' ; Ins
RETLW H'71' ; Del
RETLW H'75' ; Up
RETLW H'6B' ; LEFT
RETLW H'72' ; Dn
RETLW H'74' ; RIGHT

```

TABLA

```

MOVLW 3
MOVWF PCLATH ; QUEDARSE EN 3XX
MOVF SALTO,W ; TABLA PARA EL DISPLAY (MINÚSCULAS)
ADDWF PCL,F ; SUMAR W A PC
RETLW B'00010000' ;F1
RETLW B'00100000' ;F2
RETLW B'00110000' ;F3
RETLW B'01000000' ;F4
RETLW B'01010000' ;F5
RETLW B'01100000' ;F6
RETLW B'01110000' ;F7
RETLW B'10000000' ;F8
RETLW B'10010000' ;F9
RETLW B'00001110' ;F10
RETLW B'00011110' ;F11
RETLW B'00101110' ;F12
RETLW B'00111101' ;ESC
RETLW B'00010011' ;1
RETLW B'00100011' ;2
RETLW B'00110011' ;3
RETLW B'01000011' ;4
RETLW B'01010011' ;5
RETLW B'01100011' ;6
RETLW B'01110011' ;7
RETLW B'10000011' ;8
RETLW B'10010011' ;9
RETLW B'00000011' ;0
RETLW B'00001011' ;-
RETLW B'11010011' ;=
RETLW B'00000110' ;\
RETLW B'01001010' ;BACKSPACE
RETLW B'01011011' ;TAB
RETLW B'00011111' ;q
RETLW B'01110111' ;w

```

```

RETLW B'01010110' ;e
RETLW B'00100111' ;r
RETLW B'01000111' ;t
RETLW B'10011111' ;y
RETLW B'01010111' ;u
RETLW B'10010110' ;i
RETLW B'11110110' ;o
RETLW B'00000111' ;p
RETLW B'10110101' ;[
RETLW B'11010101' ;]
RETLW B'00010110' ;a
RETLW B'00110111' ;s
RETLW B'01000110' ;d
RETLW B'01100110' ;f
RETLW B'01111110' ;g
RETLW B'10000110' ;h
RETLW B'10101110' ;j
RETLW B'10110110' ;k
RETLW B'11000110' ;l
RETLW B'10110011' ;PUNTO Y COMA
RETLW B'01110010' ; '
RETLW B'00111010' ;ENTER
RETLW B'00101011' ;SHIFT
RETLW B'10100111' ;z
RETLW B'10000111' ;x
RETLW B'00110110' ;c
RETLW B'01100111' ;v
RETLW B'00100110' ;b
RETLW B'11100110' ;n
RETLW B'11010110' ;m
RETLW B'11000010' ; ,
RETLW B'11100010' ; .
RETLW B'11110010' ; /
RETLW B'10111100' ;CTRL
RETLW B'00000010' ;SPC
RETLW B'00001100' ;ALT
RETLW B'10011011' ;Home
RETLW B'01001100' ;PgUp
RETLW B'10001100' ;End
RETLW B'10011100' ;PgDw
RETLW B'01001101' ;Ins
RETLW B'01111100' ;DeL
RETLW B'10001010' ;Up
RETLW B'11110111' ;LEFT
RETLW B'10101101' ;Dw
RETLW B'11100111' ;RIGHT

```

TABLA3

```

MOVLW 3
MOVWF PCLATH ; QUEDARSE EN 3XX
MOVF SALTO,W ; TABLA PARA EL DISPLAY (MAYÚSCULAS)
ADDWF PCL,F
RETLW B'00010000' ;F1
RETLW B'00100000' ;F2
RETLW B'00110000' ;F3

```

```

RETLW B'01000000' ;F4
RETLW B'01010000' ;F5
RETLW B'01100000' ;F6
RETLW B'01110000' ;F7
RETLW B'10000000' ;F8
RETLW B'10010000' ;F9
RETLW B'00001110' ;F10
RETLW B'00011110' ;F11
RETLW B'00101110' ;F12
RETLW B'00111101' ;ESC
RETLW B'00010010' ;!
RETLW B'00000100' ;@
RETLW B'00110010' ;#
RETLW B'01000010' ;$
RETLW B'01010010' ;%
RETLW B'11100101' ;^
RETLW B'01100010' ;&
RETLW B'10100010' ;*
RETLW B'10000010' ;(
RETLW B'10010010' ;)
RETLW B'11110101' ;_
RETLW B'10110010' ;+
RETLW B'10100011' ;|
RETLW B'01001010' ;BACKSPACE
RETLW B'01011011' ;TAB
RETLW B'00010101' ;Q
RETLW B'01110101' ;W
RETLW B'01010100' ;E
RETLW B'00100101' ;R
RETLW B'01000101' ;T
RETLW B'10010101' ;Y
RETLW B'01010101' ;U
RETLW B'10010100' ;I
RETLW B'11110100' ;O
RETLW B'00000101' ;P
RETLW B'10110111' ;{
RETLW B'11010111' ;}
RETLW B'00010100' ;A
RETLW B'00110101' ;S
RETLW B'01000100' ;D
RETLW B'01100100' ;F
RETLW B'01110100' ;G
RETLW B'10000100' ;H
RETLW B'10100100' ;J
RETLW B'10110100' ;K
RETLW B'11000100' ;L
RETLW B'10100011' ; :
RETLW B'00100010' ; "
RETLW B'00111010' ; ENTER
RETLW B'00101011' ; SHIFT
RETLW B'10100101' ;Z
RETLW B'10000101' ;X
RETLW B'00110100' ;C
RETLW B'01100101' ;V

```

```
RETLW B'00100100' ;B
RETLW B'11100100' ;N
RETLW B'11010100' ;M
RETLW B'11000011' ; <
RETLW B'11100011' ; >
RETLW B'11110011' ; ?
RETLW B'10111100' ; CTRL
RETLW B'00000010' ; ESPACIO
RETLW B'00001100' ; ALT
RETLW B'10011011' ; Home
RETLW B'01001100' ; PgUp
RETLW B'10001100' ; End
RETLW B'10011100' ; PgDw
RETLW B'01001101' ; Ins
RETLW B'01111100' ; Del
RETLW B'10001010' ; Up
RETLW B'11110111' ; LEFT
RETLW B'10101101' ; Dw
RETLW B'11100111' ; RIGHT
END
```

APÉNDICE D

Manual del usuario

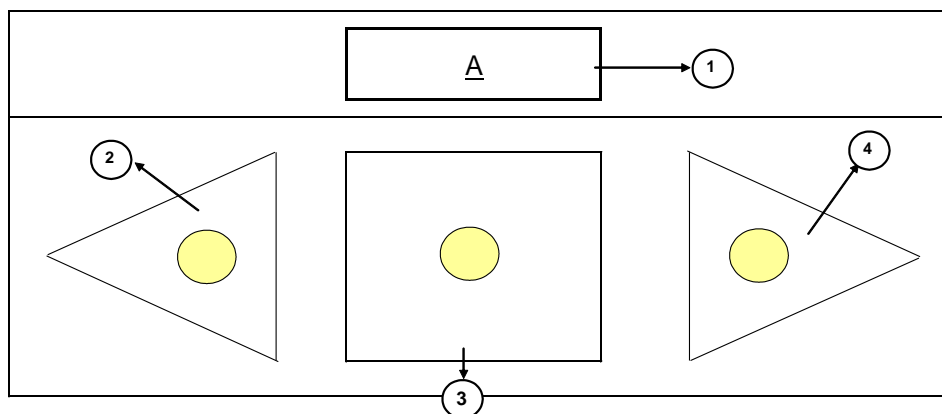
Este manual tiene como objetivo proporcionar al usuario del teclado una explicación sencilla acerca del funcionamiento del teclado especial para minusválidos

1. Disposiciones generales: Este teclado está diseñado para funcionar únicamente con computadoras tipo PC con teclado AT, pueden ser máquinas de escritorio o computadoras portátiles, antiguas o nuevas, desde computadoras con procesadores 286 hasta las más nuevas con los procesadores actuales

Componentes del teclado: (Ver figura # 1)

- Pantalla de corrimiento:(1) en esta pequeña pantalla aparecen cada una de las letras que se van desplegando.
- Botón izquierdo:(2) al presionar este botón se despliegan los caracteres hacia atrás
- Botón central:(3) al presionar este botón se envía a la computadora el carácter que está en la pantalla de corrimiento.
- Botón derecho(4): al presionar este botón se despliegan los caracteres hacia delante

Figura # 1



2. Modalidades del teclado. Existen dos tipos de modalidades del teclado:

TECLADO SIMPLIFICADO: Está diseñado para las personas que por primera vez van a hacer uso del teclado, y puedan acostumbrarse a usarlo con programas sencillos; en este teclado aparecen solamente letras y números (36 caracteres).

TECLADO COMPLETO: Está diseñado para que la persona que ya conoce el teclado anterior lo pueda usar en programas más completos como procesadores de palabras, hojas electrónicas, bases de datos etcétera donde podrá hacer uso de todos los símbolos que aparecen en un teclado normal y no solamente letras y números.

Para poder cambiar la modalidad del teclado seguir los pasos de la sección 4 d

3. Conexión del teclado especial. Antes de conectar el teclado especial cerciórese que la computadora esté APAGADA.

Para computadoras de escritorio se deben seguir los siguientes pasos

- Desconecte el teclado convencional, este conector se encuentra en la parte posterior del CPU de la computadora ver figura # 2. punto 4. Frecuentemente se encuentra junto a la entrada del ratón y puede diferenciarlos por sus símbolos y colores. Ver figura # 3

Figura # 2

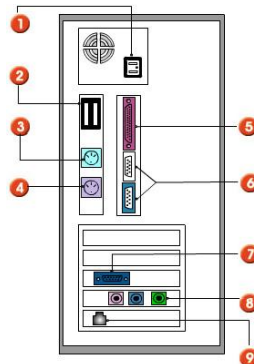
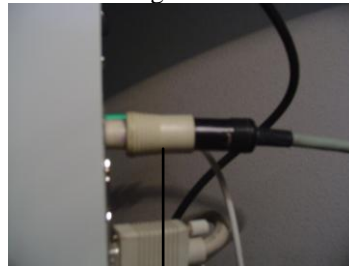


Figura # 3



- Conecte el cable del teclado especial en donde se encontraba el anterior. Ver figura # 4

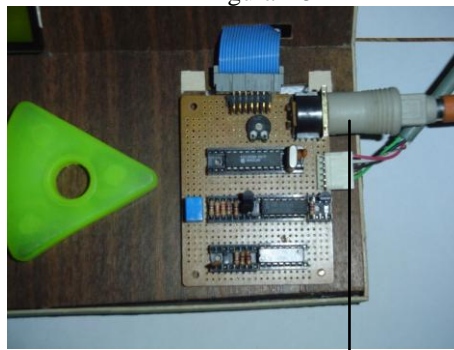
Figura # 4



(adaptador)

- Ahora conecte el cable del teclado convencional en el conector derecho del teclado especial ver figura # 5

Figura # 5



(adaptador)

Nótese en las figuras # 4 y # 5 marcadas como adaptador, estos adaptadores deben de ser utilizados para las computadoras que poseen teclados con conector mini din (ver figura # 3), para computadoras más antiguas que poseen conectores din 180 dichos adaptadores deberán ser retirados y conectar el cable directo con el conector din 180 tanto en la parte posterior de la computadora como en el teclado convencional de la misma forma que se explicó anteriormente.

Para computadoras que no tengan conector como el que se muestra en la figura # 3, si no que solamente USB se deberá utilizar un convertidor de USB a PS2 mostrado en la figura # 6, dicho convertidor se deberá colocar en el cable del teclado especial y utilizar un teclado que tenga conector que se muestra en la figura # 3.

Figura # 6



Para computadoras portátiles tipo Lap-Top, solamente es necesario conectar el teclado especial en la parte de atrás (donde aparece el símbolo de teclado o ratón).

4. Uso del teclado:

a. Selección de velocidad: Una vez conectados, tanto el teclado especial como el convencional, proceda entonces a encender la computadora y comenzar a hacer uso del teclado.

Lo primero que aparece en la pantalla del teclado es la palabra

VELOCIDAD?

Luego aparece el siguiente mensaje

MIN ■■■■ □ □ □ □ MAX
 --- --- --- ---

La figura anterior representa una escala con diferentes niveles de velocidad, la mostrada es la velocidad media, para incrementar la velocidad presione la tecla de la derecha, y para disminuirla presione la tecla de la izquierda, dicha escala representa 8 niveles diferentes de velocidad de despliegue de los caracteres.

Una vez seleccionada la velocidad de despliegue de caracteres se presiona el botón central indicando que la velocidad seleccionada es la correcta y comenzarán a desplegarse los símbolos en la pantalla.

b. **Uso de los botones:** Una vez seleccionada la velocidad comenzará a desplegarse en la pantalla del teclado los caracteres de uno en uno, el recorrido de los caracteres será en el mismo orden que aparecen en el teclado y de forma cíclica, es decir una vez termine el último carácter comienza a desplegar el primero, presionando el botón de la izquierda comenzarán a desplegarse los caracteres hacia atrás y presionando el botón del centro se selecciona la letra que aparece en la pantalla y se manda hacia la computadora, a continuación un ejemplo para mejor comprensión:

Las letras y números dentro del teclado común están distribuidas así:

```

1 2 3 4 5 6 7 8 9 0
  q w e r t y u i o p
    a s d f g h j k l
      z x c v b n m

```

Suponga que en este momento la pantalla de corrimiento del teclado especial muestra la letra Q (ver figura # 1) y quiere escribir la palabra “SOL” presione el botón derecho del teclado especial y en la pantalla de corrimiento del teclado van apareciendo las letras de una en una, en el siguiente orden:

```

q w e r t y u i o p a s

```

Al aparecer la letra “S” en la pantalla de corrimiento, presione la tecla central y la letra “S” aparecerá en la pantalla de la computadora, luego presione el botón izquierdo y en la pantalla del teclado van apareciendo las letras, de una en una en el siguiente orden:

```

s a p o

```

Al aparecer la letra “O” presione el botón central nuevamente, luego presione el botón derecho una vez más y aparecerán las letras, de una en una en el siguiente orden:

p a s d f g h j k l

Al aparecer la letra “L” presione el botón central nuevamente y habrá escrito la palabra “SOL” en la computadora personal.

En resumen, puede seleccionar las letras y números que van apareciendo en la pantalla de corrimiento del teclado especial hacia la izquierda o derecha en el siguiente orden:

1 2 3 4 5 6 7 8 9 q w e r t y u i o p a s d f g h j k l z x c v b n m

Y escribir cualquier palabra. Nótese que en la pantalla de corrimiento del teclado solo aparece una letra o número a la vez cuando se dice que se corren hacia la derecha o izquierda indica que desaparece la que en ese momento está y aparece la siguiente en el orden explicado.

En cualquier momento en el que se esté desplegando las letras hacia delante y hacia atrás se podrá acelerar la velocidad de despliegue a su máximo dejando presionado el botón derecho si las letras se despliegan hacia ese sentido o presionado el botón izquierdo atrás si se despliegan en el sentido inverso.

Cuando se utilice el teclado en el modo completo la forma en que aparecen los caracteres es similar al explicado, solo que en vez de desplegarse solamente números y letras también se despliegan y pueden enviarse hacia la computadora todos los símbolos que aparecen en un teclado convencional, en la siguiente sección se explica el funcionamiento de algunas teclas que operan de forma especial

c. **Teclas especiales:** Cuando se utiliza el teclado en modo completo existen teclas especiales como la de SHIFT, CTRL Y ALT, que funcionan de manera diferente que en el teclado convencional; estas teclas especiales se activan al seleccionarlas con el botón central y se desactivan seleccionándolas nuevamente, es decir, cuando se escoge una de estas teclas es como si quedara presionada la tecla teniendo la oportunidad de presionar cualquier otra tecla o seleccionarla con el teclado especial, una vez hecho esto se deberá escoger nuevamente la tecla especial para liberar la tecla.

Para mayor comprensión, suponga que desea presionar CLT F8, seleccionamos CLT, luego seleccione F8, y nuevamente seleccione CLT, nótese que al seleccionar CLT por primera vez aparece un símbolo

especial en el extremo izquierdo de la pantalla especial para indicar que está activa dicha función ver figura # 7, una vez es seleccionada esta tecla especial nuevamente se borra el símbolo que aparece en los extremos de la pantalla del teclado especial indicando que la función está desactivada.

Figura # 7



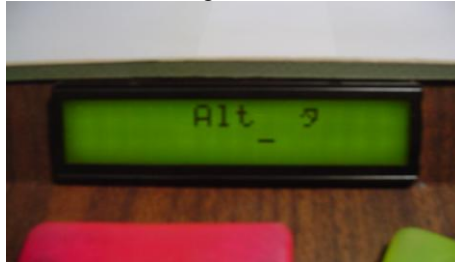
Así dichas teclas especiales pueden ser activadas y desactivadas individualmente teniendo el efecto de presionar varias teclas a mismo tiempo, cada vez que se selecciona una de estas teclas especiales es como si la tecla quedara presionada y cuando se presiona nuevamente como si fuera liberada. Nótese que cuando queda activa la tecla de SHIFT además de aparecer el símbolo especial en el extremo de la pantalla, ver figura # 8, las letras aparecen en mayúsculas y en vez de aparecer los números aparecen los símbolos de que se encuentran arriba de las teclas de números, dicha función es similar a la tecla CAPS LOCK o bloqueo de mayúsculas con la diferencia de en un teclado convencional una vez activada la tecla CAPS LOCK las teclas de los números aparecen como tal y no con los símbolos que se encuentran arriba de estas teclas.

Figura # 8



Para la tecla ALT el símbolo que indica que está activa es el que se muestra en la figura # 9 y funciona de manera similar a las de CLT y SHIFT.

Figura # 9

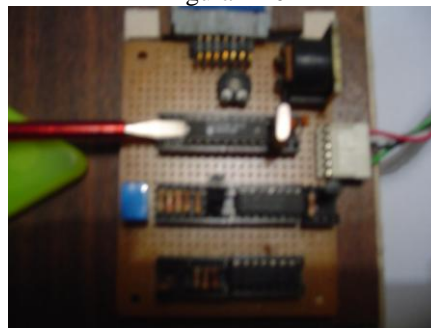


Nota: es muy importante el hecho que al estar activas estas teclas no se podrán usar los dos teclados al mismo tiempo, es en el único momento en el que ambos teclados el convencional y el especial no podrán ser utilizados uno después del otro dadas las diferencias de las características de estas teclas especiales (SHIFT, ALT y CTRL).

d. Cambio de modalidad del teclado: Como se explicó en la sección # 2 existen dos modalidades para el uso del teclado, para cambiar la modalidad del teclado es necesario el cambio del microcontrolador, para esto se deberán seguir los siguientes pasos;

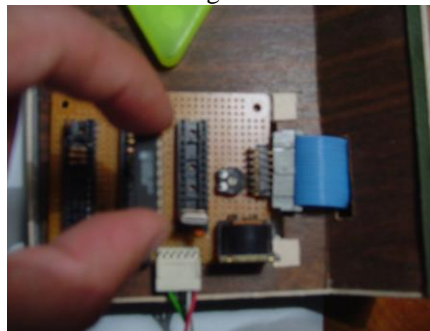
- Asegúrese que el teclado especial esté desconectado de la computadora, la pantalla deberá estar apagada.
- Inserte un desarmador pequeño y de punta plana entre el microcontrolador y la base del mismo ver figura # 10,

Figura # 10



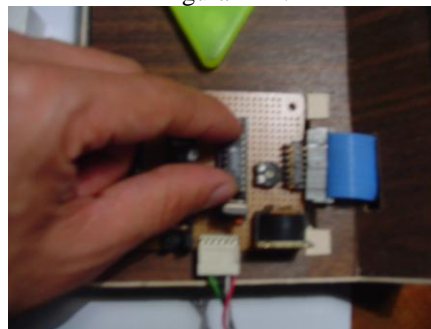
- Presione hacia abajo el mango del desarmador a manera de hacer palanca para que el microcontrolador pueda ser retirado hacia arriba.
- Tome el otro microcontrolador de la forma en que se muestra (figura # 11) y colóquelo en el lugar que se encontraba el anterior. Asegúrese que el microcontrolador sea puesto en la posición que se muestra, el colocar el microcontrolador en la posición inversa podría dañarlo o borrar la programación interna del mismo.

Figura # 11



- Presione hacia abajo para asegurarse que halla entrado en la posición correcta. Ver figura #12.

Figura # 12.



APÉNDICE E

Diagrama de tiempo del controlador de pantalla de cristal líquido

TIMING CHARACTERISTICS

Item	Symbol	Test condition	Min.	Typ.	Max.	Unit
Enable cycle time	t_{cyc}	Fig. 5, Fig. 6	1.0	—	—	μs
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

