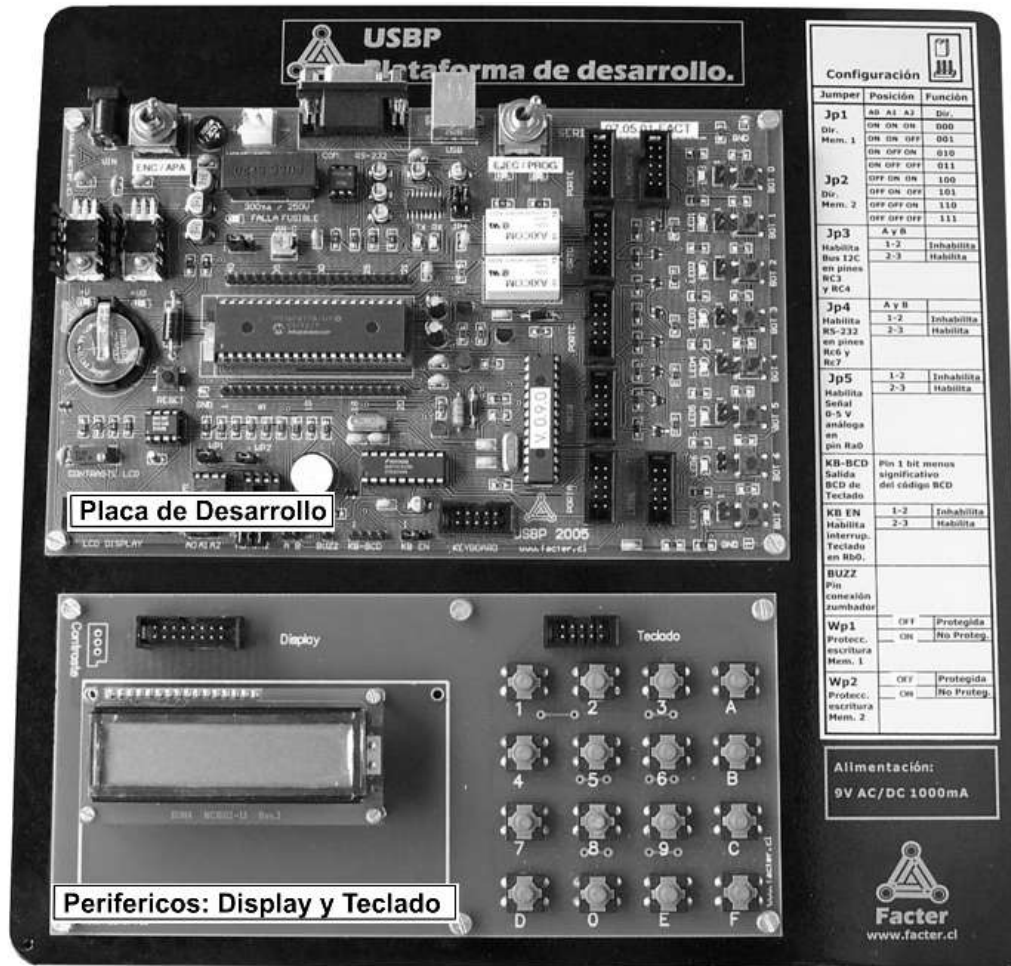


USBP SISTEMA DE DESARROLLO BASADA EN PIC16F877A MICROCHIP

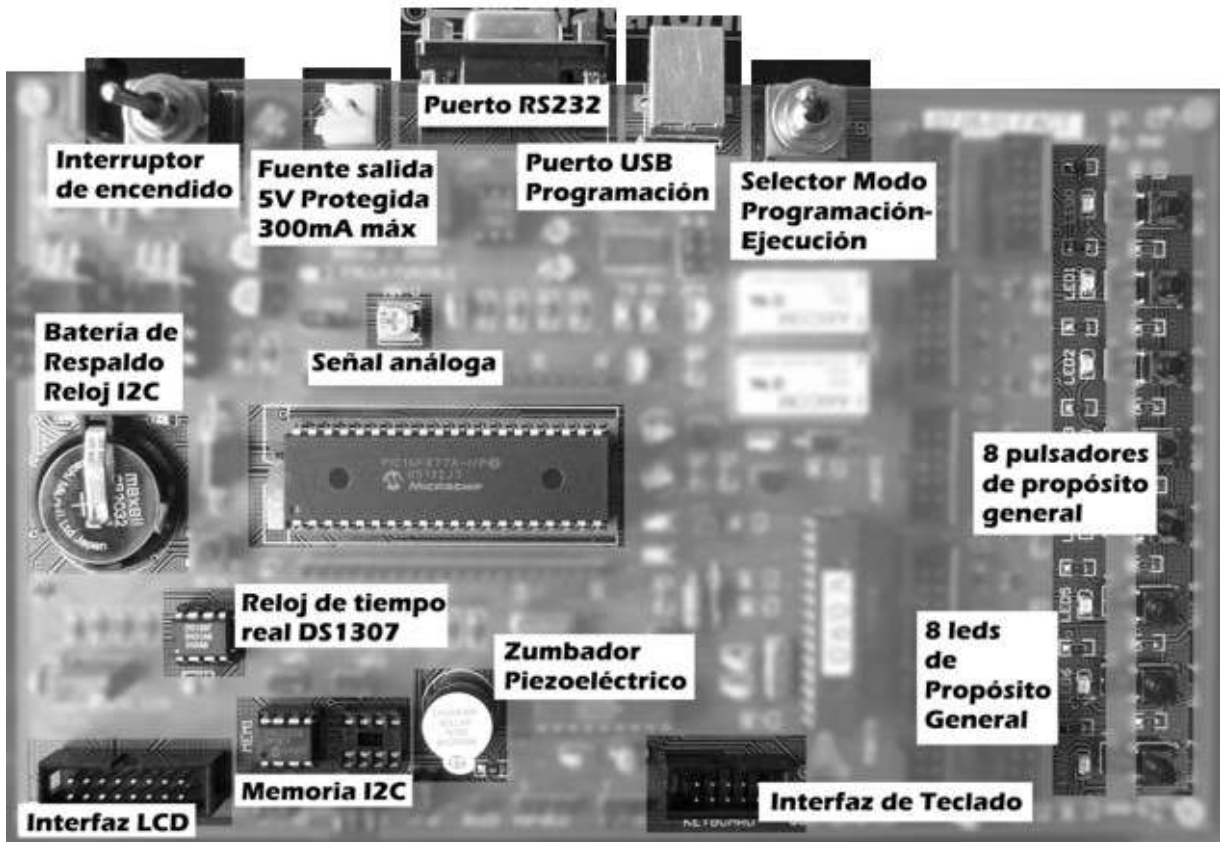


La tarjeta de desarrollo USBP, está orientada a la creación de prototipos, así como a la experimentación y prueba de diversos programas. Para ello consta de los periféricos comúnmente más utilizado para dichos fines, permitiendo un trabajo cómodo y un menor tiempo de implementación.

Características generales:

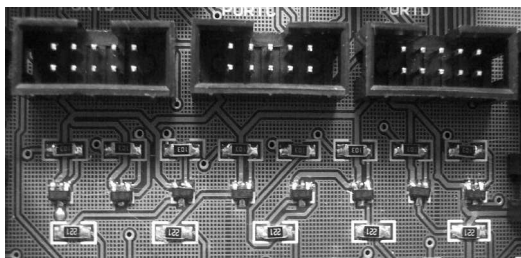
- Acceso a la totalidad de los pines del microcontrolador.
- Puertos De comunicación RS232, I2C y USB (éste último sólo programación)
- Bus I2C: Reloj de tiempo real con batería de respaldo (DS1307) y memoria serial de 256 kbits.
- Teclado de 16 botones con codificador BCD (74C922) con posibilidad de interrupción directa en pin RB0 y display de cristal líquido.con interfaz al Puerto D.
- 8 pulsadores y 8 leds de propósito general con posibilidad de interconexión directa a los puertos del microcontrolador.
- Pic16f877A montado en ZIF Sócket.
- Zumbador piezo eléctrico incorporado.

- Generación de señal analógica entre 0 y 5 V para simulación en RA0.
- Capacidad de programación, borrado y verificación del dispositivo mediante puerto USB.
- Salida de Voltaje para interconexión de circuitos externos.



1.- Puertos:

Los puertos del microcontrolador PIC16F877A son accesibles de manera directa a través de los conectores IDC ubicados en la placa e identificados bajo los nombres a los cuales corresponden (PORTA, PORTB, PORTC y PORTE).



Conectores IDC

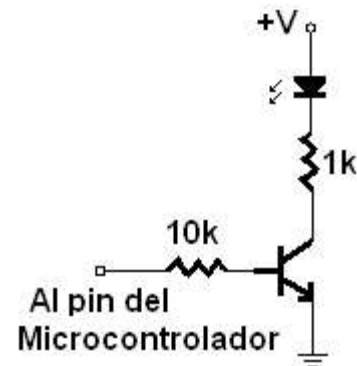
2.- Leds y Pulsadores:

La placa de desarrollo dispone de un conjunto de 8 leds y 8 pulsadores, los que pueden ser conectados a los puertos

mediante el uso de los conectores IDC y el cable correspondiente.

2.1 Leds:

Al unir mediante el cable IDC los leds al puerto del microcontrolador, se obtiene la siguiente configuración:

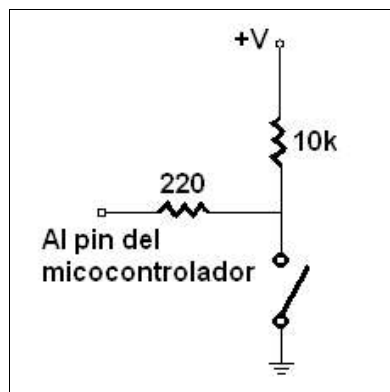


Quedando el encendido del led en lógica positiva respecto del pin del microcontrolador. Cabe recordar que para poder encender el led, el puerto ha de estar configurado como digital.

Ésta configuración se repite para todos los pines que componen el puerto.

2.2 Pulsadores:

La configuración obtenida al unir el puerto del microcontrolador con los pulsadores es la siguiente:



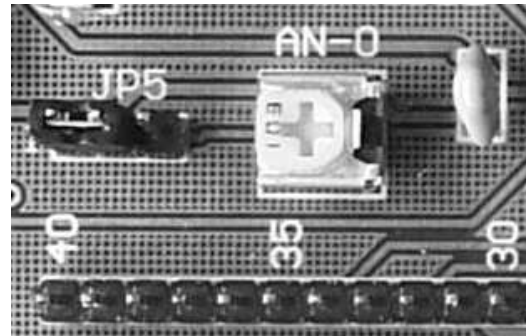
Luego para un estado normal abierto del pulsador, el voltaje presente en el pin del puerto será de 5 VDC, y cambiará 0 Vdc cuando éste sea pulsado.

Frente a cada pulsador hay disponible un conector que permite utilizar un pulsador externo. Al hacer uso de ésta capacidad, éste estará en paralelo con el incorporado en la placa.

3.- Entrada Análoga

Es posible simular una señal análoga que excursione entre 0 y 5 V mediante el uso del potenciómetro **AN-0**. Para ello se requiere fijar el jumper **JP5** en las posiciones 2-3, de ésta manera, la señal análoga se replica en el pin RA0 del microcontrolador. (Recuerde programar el Pin RA0 como entrada análoga).

Si no se requiere el uso del potenciómetro para generar la señal análoga recuerde mantener el jumper **JP5** en los pines 1-2.

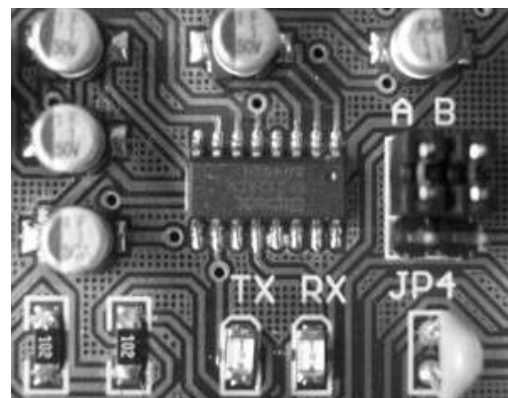


4.- Puertos de Comunicaciones:

La tarjeta de desarrollo dispone de dos puertos de comunicaciones: un RS232 y un bus I2C , además de un puerto de sólo programación USB.

4.1 RS-232:

La comunicación RS232 está basada en el transceiver MAX232. y se lleva a cabo mediante los pines RC6 y RC7 del microcontrolador (RC6 Transmisión, y RC7 Recepción).



Para habilitar dicha comunicación, es necesario posicionar simultáneamente los Jumpers A y B de **JP4** en las posiciones 2-3. En caso contrario, de no querer hacer uso de ésta característica, se deben mantener los jumpers en las posiciones 1-2.

Al Conector DB9 ubicado a un costado de la placa le corresponde el siguiente pinout:

Pin DB9 Hembra	Señal
Pin 1	N.C.
Pin 2	Rx
Pin 3	Tx

Pin 4	N.C.
Pin 5	Gnd
Pin 6	N.C.
Pin 7	N.C.
Pin 8	N.C.
Pin 9	N.C.

Existen además dos leds de estado para la comunicación RS232 ubicados a un costado del transceiver, los cuales están "colgados" a las líneas de TX y RX y parpadean cada vez que haya datos transfiriéndose.

4.2 Bus I2C:

La tarjeta incorpora un bus I2C, constituido por un reloj de tiempo real DS1307 (Dallas semiconductor) y una memoria Serial 24LC256 (Microchip). A su vez tiene la capacidad de incorporación de una segunda memoria serial. El reloj de tiempo real posee una batería de 3V de respaldo, y un reloj cristal de 32,768KHz.

Para la habilitación del Bus I2C es necesario posicionar los jumpers A y B de **JP3** en las posiciones 2-3. Con ello se accede al bus I2C por los pines RC4 (datos) y RC3 (Reloj).

En caso de no requerir la comunicación I2C, los jumpers que componen **JP3** deben permanecer en la posición 1-2. Las direcciones de las memorias al interior del bus I2C pueden ser configuradas mediante los jumpers que componen **JP1** y **JP2**. (A0 A1 y A2).

Donde A2 es el bit menos significativo.

También es posible proteger las memorias contra escritura mediante los jumpers **WP1** y **WP2** para las memorias 1 y 2 respectivamente. Cuando éstos se encuentran en estado OFF, la memoria se encuentra protegida, y en estado ON ésta no lo está.

4.3 Bus de programación USB.

La tarjeta posee un puerto de programación USB, a través del cual es posible cargar un programa en el Microcontrolador, borrarlo o verificar su contenido, pero no es posible establecer

comunicación directa con el Microcontrolador 16F877A para otros fines.

5.- Teclado

La tarjeta dispone de un codificador de teclado 74C922, el cual entrega en BCD el código correspondiente a la tecla presionada.

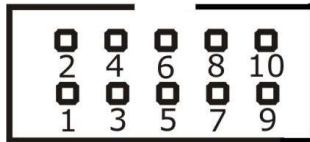
El pin 1 en el jumper **KB-BCD** corresponde al bit menos significativo del código BCD. El valor correspondiente según la tecla presionada se muestra en la siguiente tabla:

TECLA	Código BCD
1	1111
2	1110
3	1101
A	1100
4	1011
5	1010
6	1001
B	1000
7	0111
8	0110
9	0101
C	0100
D	0011
Ø	0010
E	0001
F	0000

A su vez el codificador entrega una señal de interrupción, la cual se activa cada vez que una tecla es presionada indicando que el código BCD está disponible para ser capturado. Para acceder a dicha señal de interrupción, es necesario configurar el jumper **KB EN**. Cuando éste se encuentra en la posición 2-3 la interrupción accederá al Pin RB0. Se requiere que dicha interrupción esté habilitada y manejada a nivel de firmware.

En caso de no requerir el uso del teclado, o en su caso de la interrupción, el Jumper **KB EN** debe mantener la posición 1-2.

El pinout correspondiente al conector IDC del teclado es el siguiente:



Pin IDC	Teclado
1	Fila 2
2	Fila 1
3	Fila 3
4	Columna 1
5	Columna 2
6	
7	Columna 3
8	
9	Columna 4
10	Fila 4

Donde :

Fila 1: Teclas 1-2-3-A
 Fila 2: Teclas 4-5-6-B
 Fila 3: Teclas 7-8-9-C
 Fila 4: Teclas D-0-E-F

Columna 1: Teclas 1-4-7-D
 Columna 2: Teclas 2-5-8-0
 Columna 3: Teclas 3-6-9-E
 Columna 4: Teclas A-B-C-F

6.- Display LCD

La tarjeta dispone de una interfaz a un display de cristal líquido (LCD) de cuatro hilos. Y su control está a cargo del puerto D del microcontrolador.

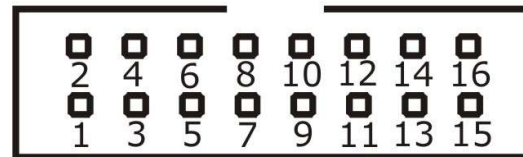
La tabla siguiente muestra la asignación de pines obtenida al interconectar el display LCD con la tarjeta de desarrollo.

Pinout del LCD		Pin microcontrolador
1	Gnd	Gnd
2	VDD	Vdd
3	V Contraste	Ajustable mediante Potenciómetro
4	RS	RD1
5	R/W	RD2
6	E	RD0
7	Data0	
8	Data1	
9	Data2	
10	Data3	
11	Data4	RD4
12	Data5	RD5

13	Data6	RD6
14	Data7	RD7
15	V+ Backlight	Vdd
16	Gnd Backlight	Manejado por RD3 RD3 en alto= Backlight ON RD3 en bajo= Backlight OFF

También es posible intercambiar el display LCD por uno de 4 líneas por 20 columnas, siempre y cuando éste sea pinout compatible con el indicado en la tabla anterior.

El pinout del conector IDC del Display es el siguiente:

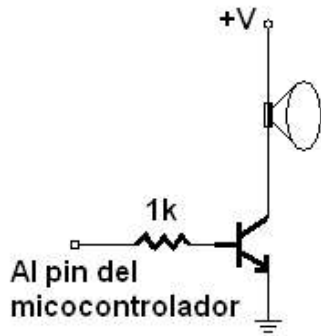


Pin IDC	Pin LCD
1	5 R/W
2	4 RS
3	6 E
4	3 VContraste
5	7 Data 0
6	2 VDD
7	8 Data 1
8	1 Gnd
9	9 Data 2
10	16 Gnd Backlight
11	10 Data 3
12	15 V Backlight
13	11 Data 4
14	14 Data 7
15	12 Data 5
16	13 Data 6

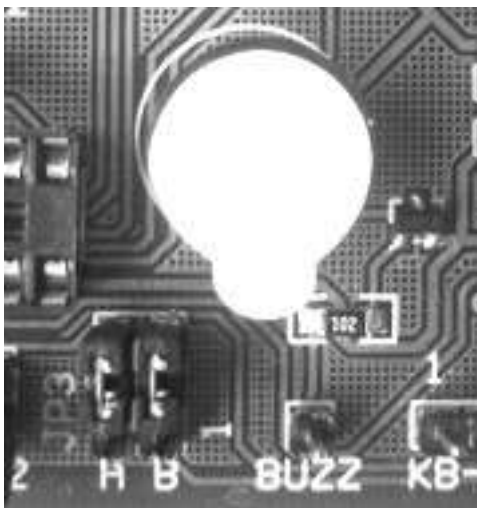
En caso de que el Display no sea utilizado, éste deberá desconectarse de la placa principal.

7.- Zumbador Piezo eléctrico:

Se incorpora un zumbador piezo eléctrico de 5 VDC, el cual es manejado por el circuito de la figura:



Dicho zumbador puede ser controlado por cualquier pin del microcontrolador previamente configurado como salida digital. Para ello basta con interconectar el pin **BUZZ** con el pin del microcontrolador escogido mediante el uso del cable apropiado para tal fin.

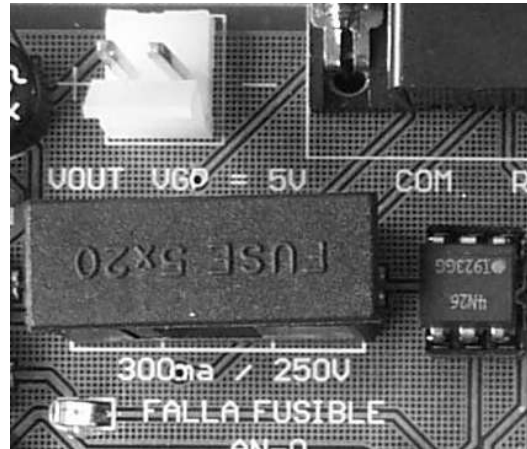


8.- Salida de Voltaje auxiliar

Se dispone de una salida de voltaje auxiliar de 5V protegida mediante un fusible, de una capacidad de corriente máxima de 300 mA. A través de ella es posible alimentar circuitos externos cuyo consumo no exceda la capacidad de dicha fuente.

Ante una falla en el fusible debido a algún cortocircuito producido por algún circuito externo, el fusible se quemará, y se encenderá el led **FALLA FUSIBLE** indicando su estado defectuoso. Éste debe ser reemplazado por uno de **similares Características**, de no ser así, **se pueden provocar daños**

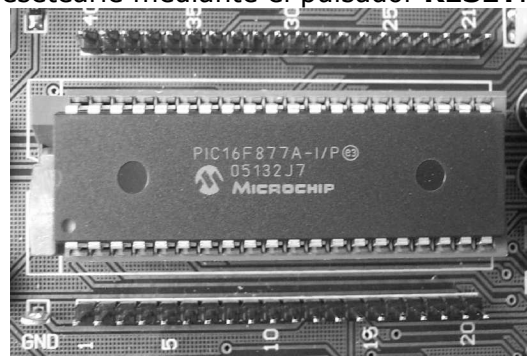
irreparables en la tarjeta de desarrollo.



Al hacer uso de ésta fuente, se dispone de una tierra común de referencia, pues su tierra está igualada a la de la placa de desarrollo. De igual manera existen puntos dispuestos en distintos lugares dentro de la placa cuya función es la de acceder a una tierra de referencia. Éstos últimos son fácilmente identificables, pues están rotulados como GND.

9.- El Microcontrolador

El microcontrolador de desarrollo corresponde a un PIC16F877A, y trabaja con un cristal de 4 MHz. Es posible resetearle mediante el pulsador **RESET**.



Se puede acceder a cualquiera de los 40 pines que lo componen, esto se logra a través de los conectores tipo slim (Poste Plano) dispuestos a los costados del zif socket.

10.- El encendido:

Antes de encender la tarjeta de desarrollo, se debe verificar que el voltaje de alimentación no exceda el recomendado de 9 VAC ó 9 VDC. En caso de alimentar con 9 VDC, la polaridad no es relevante, puesto que la fuente de poder está protegida ante inversiones.

Luego, basta con activar el interruptor basculante de encendido, y llevarlo de la posición **APA** (apagado) a la posición **ENC** (encendido). En éste instante debieran permanecer encendidos simultáneamente los leds **+V** y **+VG**, los cuales son indicadores del buen funcionamiento de las fuentes que se encargan de suministrar energía a la placa.

Para apagarla, basta con revertir el estado del interruptor llevándolo a su posición **APA**.

11.- La Programación

La programación del dispositivo se puede realizar ya sea mediante el uso de un programador externo (extrayendo el microcontrolador del zif Sócket) ó mediante el puerto de comunicaciones USB a través del software que acompaña la tarjeta de desarrollo.

El sistema de programación mediante el puerto USB está basado en la comunicación establecida entre el microcontrolador PIC16C745 y el PIC 16F877A. El primero se comunica hacia el PC e interactúa con la interfaz gráfica, encargándose del manejo de la memoria de programa del 16F877A. De ésta forma es posible: cargar un programa, limpiar o leer la memoria y verificar si el programa es equivalente al cargado en la interfaz gráfica.

El código fuente ha de ser compilado de manera externa, ya sea en MPLAB o con algún compilador de alto nivel si fuese necesario. Luego el archivo resultante de extensión *.hex* ha de ser cargado mediante la interfaz gráfica.

Los pasos a seguir para poder comunicar la tarjeta vía puerto USB son los siguientes:

1.- **Se debe apagar** la tarjeta de desarrollo.

2.- Se debe conectar el cable USB entre la tarjeta de desarrollo y el PC.

3.- Se posiciona el interruptor de palanca en la posición **PROG**.

4.- Se enciende la Tarjeta mediante el interruptor de encendido (posición **ENC**). (Si es la primera vez que se realiza ésta operación, el sistema operativo Windows XP o Windows 2000 buscará el driver apropiado).

5.- Se ejecuta la interfaz gráfica.

6.- Se procede a cargar un programa en memoria, limpiar la memoria o verificar su contenido. Una vez finalizado, se desconecta el cable USB y se lleva el interruptor a la posición **EJEC**.

Es recomendable apagar y encender nuevamente la tarjeta con tal de permitir que todos los periféricos se inicialicen de buena manera.

12.- Configuración (Jumpers):

JP1, JP2: Fijan la dirección de las memorias seriales al interior de bus I2C. poseen tres jumpers: A0, A1 y A2. Éste último corresponde al bit menos significativo. El estado ON corresponde a un "0" lógico, y el estado OFF a un "1" lógico.

Una tabla resumen con las posibles combinaciones se muestra a continuación.

A0	A1	A2	Dirección
ON	ON	ON	000
ON	ON	OFF	001
ON	OFF	ON	010
ON	OFF	OFF	011
OFF	ON	ON	100
OFF	ON	OFF	101
OFF	OFF	ON	110
OFF	OFF	OFF	111

JP3: Habilita el uso del Bus I2C mediante los pines RC3 (reloj) y RC4 (Datos). Ambos jumpers (**A** y **B**) son simultáneos, y deben permanecer en la posición 2-3

en caso de querer hacer uso del bus, o en la posición 1-2 en caso contrario.

JP4: Habilita el uso del puerto RS232 a través de los pines RC6 (Transmisión) y RC7 (Recepción). Los jumpers **A** y **B** trabajan de manera simultánea, y ambos deben estar en las posiciones 2-3 en caso de querer uso del puerto, ó en las posiciones 1-2 en caso contrario.

JP5: Habilita el uso del potenciómetro **AN-0** como generador de una señal análoga que excursiona entre 0 y 5 VDC con conexión directa hacia el Pin RA0. En la posición 2-3 ésta función está habilitada, en caso de no ser requerida, debe permanecer en la posición 1-2.

KB-BCD: Entrega el código BCD. En donde el Pin 1 es el bit menos significativo.

Requiere que el teclado esté previamente conectado mediante el cable IDC a la tarjeta principal (Conectores **TECLADO** unidos mediante el cable IDC correspondiente).

KB EN: Éste jumper permite habilitar la interrupción al pin RB0 desde el codificador de teclado. En la posición 2-3, ésta se encuentra habilitada, de no ser usada, se debe mantener en la posición 1-2.

BUZZ: Pin de conexión para manejar el encendido del zumbador piezo eléctrico. Puede ser conectado a cualquier pin digital del microcontrolador mediante los conectores slim que se encuentran a los costados del ZIF Sócket.

WP1, WP2: Jumpers de protección contra escritura de las memorias seriales **MEM1** y **MEM2** respectivamente. En posición OFF, éstas se encuentran protegidas, en ON no lo están.

13.- Documentación:

- **PIC16F877A:** Microcontrolador. Fabricante: Microchip (www.microchip.com)
Hoja de datos: <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>
 - **24LC256:** Memoria serial I2C de 256 kbits (32 kbytes) Fabricante: Microchip (www.microchip.com).
Hoja de datos: <http://ww1.microchip.com/downloads/en/DeviceDoc/21203M.pdf>
 - **DS1307:** Reloj/calendario de tiempo real I2C. Fabricante: Dallas semiconductor (www.dalsemi.com)
Hoja de datos: <http://pdfserv.maxim-ic.com/en/ds/DS1307.pdf>
 - **74c922:** Codificador BCD de teclado CMOS. Fabricante: Fairchild (www.fairchildsemi.com)
Hoja de datos: <http://www.fairchildsemi.com/ds/MM%2FMM74C922.pdf>
-

14.- Condiciones de Garantía.

El siguiente producto tiene un período de garantía de tres meses a partir de la fecha de venta. Ésta cubre fallas de fabricación o fallas que se puedan presentar bajo condiciones de uso normal del producto.

La garantía no cubre daños provocados por uso, almacenamientos o traslados indebidos.

La garantía no es exigible fuera del plazo establecido, aunque el producto no haya sido usado.

La garantía no cubre daños provocados al Microcontrolador PIC16F877A.

Para hacer uso de la garantía, debe ponerse en Contacto con Factor Ltda.

Factor Ltda.

Arlegui 160 oficina 2
Viña del Mar, Chile
Fono: (56)(32) 684194
factor@factor.cl

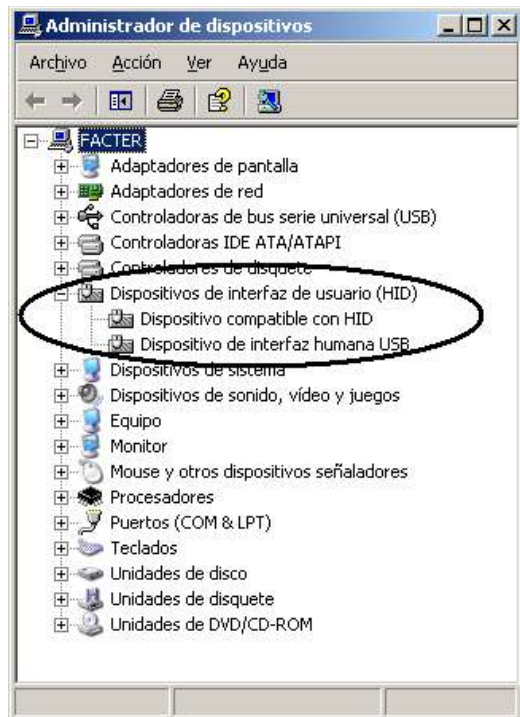
15.-Interfaz Gráfica de programación Vía Puerto USB

La interfaz de programación está basada en el uso del PIC 16C745, el cual accede a la memoria de programa del PIC16F877A permitiendo su grabación, lectura y verificación.

El software es compatible con Windows XP y Windows 2000.

Antes de ejecutar el software, la tarjeta de desarrollo debe estar apagada. Luego, se conecta el cable USB en algún puerto disponible del PC. Una vez hecho esto, se lleva el interruptor de palanca a la posición **PROG** con lo cual se enciende el Led **PROG**.

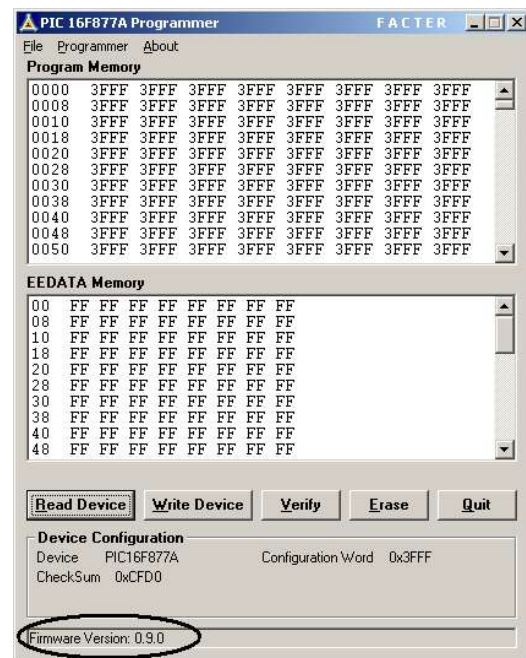
Al conectar por primera vez la tarjeta de desarrollo con el PC vía puerto USB, el Sistema operativo buscará el Driver de manera automática y configurará el dispositivo como **Dispositivo de Interfaz de usuario (HID)**.



Una vez reconocida la plataforma y configurada en forma automática por el sistema operativo, se ejecuta la interfaz de programación.

Si todo está correcto, ha de aparecer en la etiqueta inferior de la

interfaz la versión del firmware actual de la tarjeta de desarrollo tal como en la figura siguiente:



En caso de aparecer en dicha posición el mensaje "Programmer not found..", se debe verificar la correcta conexión del cable USB tanto en la tarjeta como en el PC, también la correcta configuración del puerto USB en el PC.

La interfaz consta de cuatro operaciones básicas en la programación de un microcontrolador:

1.- Leer su código (**READ DEVICE**) para posteriormente almacenarlo en un archivo de extensión .hex.

2.- Leer un archivo de extensión .hex y cargarlo en la memoria de programa del microcontrolador (**WRITE DEVICE**).

3.- Leer un archivo de extensión .hex y posteriormente verificar si corresponde al que permanece en la memoria del PIC (**VERIFY**).

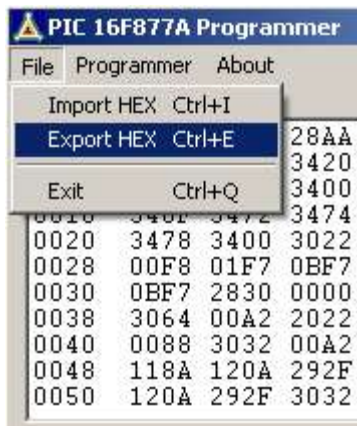
4.- Borrar la memoria del PIC (**ERASE**).

15.1.- Leer el código (READ DEVICE)

Es posible leer el código ya sea presionando el botón correspondiente (**READ DEVICE**) o en su caso mediante el uso del menú superior:



Cuando se lee la memoria del Microcontrolador, aparece una barra de estado que muestra el progreso de la operación. Se leen de manera simultánea la memoria de datos (EEPROM), luego la memoria de programa y finalmente la palabra de configuración. Finalmente se despliega su contenido en las ventanas correspondientes. Dicho contenido puede ser almacenado en un archivo de extensión *.hex*, lo cual se logra con la opción Export HEX.

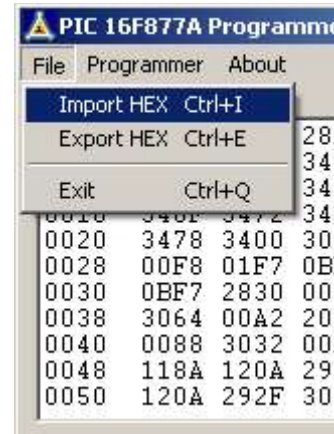


15.2.- Cargar un programa en el Microcontrolador (WRITE DEVICE).

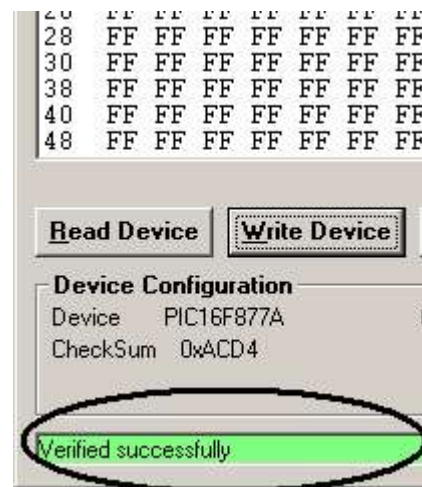
Al presionar este botón, se almacena el código presente en la ventana Program Memory, así como los datos de la ventana EEDATA Memory.

Previamente se debe importar un archivo *.hex* resultante de la compilación de un archivo fuente ya sea mediante MPLAB o algún compilador de alto nivel.

Una vez



importado, tanto el código como los datos a grabar en la memoria EEPROM se actualiza en las pantallas correspondientes. Luego al presionar el botón **WRITE DEVICE** se comienza con la escritura. Una barra de estado indica el progreso, y el resultado exitoso se indica en la etiqueta inferior acompañada con el cambio de color



Cada vez que se realiza la grabación del microcontrolador, éste se carga con la palabra de configuración por defecto 0x3f39, es decir:

Oscillator: Xtal
 Watchdog Timer: Off
 Power up timer: Off

Brown Out detect: Off
Low Voltage Program: Off
Flash Program Write Protection: Off
Background Debug: Disable
Data EEPROM Read Protect: Off
Code Protect: Off

15.3.- Verificar el código almacenado en el Microcontrolador. (Verify)

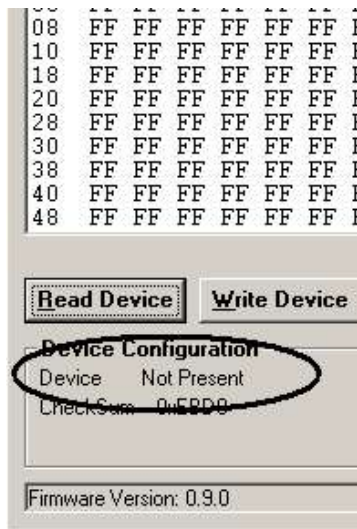
Al presionar este botón se lleva a cabo la verificación entre el código almacenado en la memoria del Microcontrolador con la que permanece en la ventana Program Memory. También se verifica la Memoria de datos EEPROM.

15.4.- Borrar la memoria del microcontrolador (Erase)

Esta función permite limpiar la memoria de datos EEPROM del Pic, así como su memoria de programa.

15.5 - Observaciones.

1.- La interfaz gráfica consulta de manera constante la presencia del microcontrolador en la placa de desarrollo, por ende en caso de aparecer el mensaje "Device Not Present" como en la figura, revise que efectivamente éste se encuentre montado en el ZIF socket, o en su caso esté bien introducido en dicho zócalo.



El microcontrolador debe ser puesto de la forma correcta en el ZIF Socket, manteniendo la correcta

orientación del pin 1. En caso contrario se pueden provocar daños tanto al Microcontrolador como al sistema de desarrollo.

2.- En caso de requerir una palabra de configuración distinta a la utilizada por defecto, el programa se deberá cargar haciendo uso de un grabador externo. Si la palabra de configuración es incorrecta en cuanto al tipo de oscilador, no se garantiza el correcto funcionamiento del sistema de desarrollo.

3.- El programador es considerado por Microchip bajo la categoría de prototipo, debido a que la verificación del código la realiza al igual que PICSTART Plus® sólo bajo la condición de VDD=5VDC.

4.- La plataforma es suministrada con un Microcontrolador PIC16F877A. El sistema de desarrollo puede hacer uso también de microcontroladores cuyos pinout sean compatibles (PIC16F877, PIC16F874), pero éstos no pueden ser grabados en la placa vía puerto USB. Éstos deberán ser grabados con algún sistema de grabación externo.

Rev.01

Factor Ltda.
Arlegui 160 oficina 2
Viña del Mar, Chile
Fono: (56)(032) 684194
factor@factor.cl

16.- Programas de Prueba

En la plataforma de desarrollo se incluyen algunos programas previamente compilados con tal de explorar de manera práctica las capacidades que ésta posee.

El programa cargado por defecto en la plataforma consiste en el manejo simultáneo del Display de Cristal Líquido, pulsadores y leds de propósito general, interfaz rs232, reloj de tiempo real, memoria serial.

16.1.- Prog1: test_leds.hex

Configuración:

LEDS= PORTB
PULSADORES=PORTA
TECLADO Y DISPLAY CONECTDOS A LA PLACA PRINCIPAL.

Jumpers:

JP1: A0, A1, A2 = ON
JP2: A0, A1, A2 = OFF
JP3: A,B = 2-3 (habilita)
JP4: A,B=2-3 (habilita)
JP5: 1-2 (inhabilita)
KB-BCD: no conectado.
KB EN:1-2 (inhabilita)
Buzz: conectar al pin 8 del micro. (RE0)
Wp1:ON
Wp2:ON

Éste programa maneja la pantalla de Cristal líquido 2x16 y se comunica vía interfaz serial RS232 según el formato 9600 8 n 1.

Al iniciarse se despliega una pantalla en el display acompañado del correspondiente sonido del zumbador, luego se lee parte del contenido de la memoria serial, y se visualiza via puerto rs232 (puede hacer uso del archivo de hyperterminal ya creado, y que se adjunta en el CD).

Luego, en el display LCD se muestran la fecha y la hora leídas desde el reloj de tiempo real DS1307.

Si se presionan los botones de propósito general del BOT0 al BOT4, se encenderán los leds.

16.2.- Prog2: lcd_kbd.hex

Configuración:

Jumpers:

JP1: JP1: A0, A1, A2 = ON
JP2: A0, A1, A2 = OFF
JP3: A,B = 1-2 (inhabilita)
JP4: A,B=2-3 (habilita)
JP5: 2-3 (habilita)
KB-BCD:a los pines RB4-RB7 (RB4= bit BCD menos significativo).
KB EN:2-3 (habilita)
Buzz: conectar al pin 8 del micro. (RE0)
Wp1:ON
Wp2:ON

Éste programa implementa el teclado 4X4, el cual al ser presionado muestra la tecla correspondiente en el Display LCD.

De igual forma es posible enviar caracteres ascii mediante el hyperterminal en formato 9600 8 n 1, e irá desplegándolos en pantalla.

La entrada análoga también es considerada y se envía vía RS232 según el formato antes mencionado de manera regular el valor que hay presente en el pin RA0.

Revisión:

Serie: _____

Contenido:

- Tarjeta de desarrollo USBP con teclado y display 2x16
- 1 Pic16f877A
- 3 Cables planos IDC de 10 pines
- 1 cable de 1 pin conexión zumbador
- 1 cable 4 terminales Slim conexión salida BCD
- 1 cable plano IDC de 16 pines
- 1 adaptador de poder 9VDC 1000mA
- 1 cable USB
- 1 cable serial con conectores DB9
- 1 CD interfaz gráfica, manuales y hojas de datos.
- 1 Manual técnico impreso

Tests:

- Inspección Física: _____
- Adaptador de poder: _____
- Puerto USB : _____
- Puerto 232 Tx y RX: _____
- Bus I2C:
 - Memorias: _____
 - RTC: _____
- Leds y pulsadores: _____
- Display LCD: _____
- Teclado y codificador: _____
- Piezoeléctrico: _____
- Señal Análoga: _____

Fecha: _____

Firma
Revisor: _____

Indice:

	Pág.
Características Generales	1
1.- Puertos	2
2.- Leds y Pulsadores	2
2.1.- Leds	2
2.2.- Pulsadores	3
3.- Entrada Análoga	3
4.- Puertos de Comunicación	3
4.1.- RS-232	3
4.2.- Bus I2C	4
4.3.- Bus de Programación USB	4
5.- Teclado	4
6.- Display LCD	5
7.- Zumbador Piezo eléctrico	6
8.- Salida Auxiliar de Voltaje	6
9.- El Microcontrolador	6
10.- El Encendido	7
11.- La Programación	7
12.- Configuración (Jumpers)	7
13.- Documentación	8
14.- Condiciones de Garantía	9
15.- Interfaz Gráfica de programación vía Puerto USB	10
15.1 Leer el código (READ DEVICE)	11
15.2 Cargar un programa en el Microcontrolador (WRITE DEVICE)	11
15.3 Verificar el Código almacenado en el Microcontrolador (VERIFY)	12
15.4 Borrar la Memoria del Microcontrolador (ERASE)	12
15.5 Observaciones	12
16.- Programas de Prueba	13
16.1 Prog1: <i>test_leds.hex</i>	13
16.2 Prog2: <i>lcd_keyb.hex</i>	13
ANEXO	
Esquema general de jumpers y conexionado	