

# **HARDWARE DEL PROYECTO**

# **PLACA MICROCONTROLADORA**

## MICROCONTROLADOR PIC16F877A

En este apartado haremos una pequeña introducción a los microcontroladores PIC, exponiendo las razones de uso de estos microcontroladores en general, y en particular para este proyecto que estamos tratando. Además introduciremos las principales características de los mismos. En otro apartado más adelante, entraremos más en profundidad en los microcontroladores PIC atendiendo a otras características de los mismos.

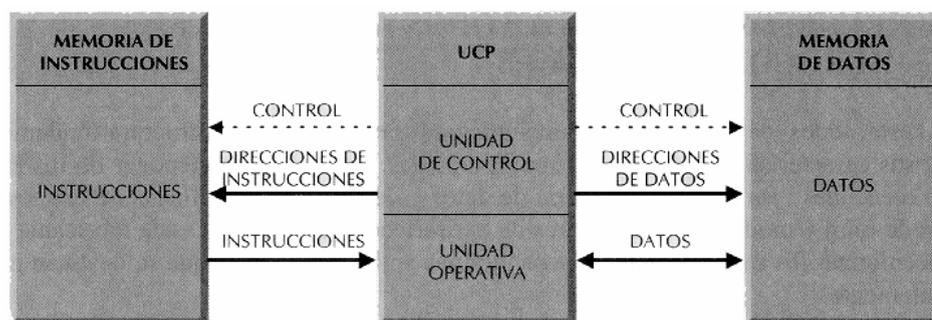
Las principales razones de la predilección de la mayoría de los diseñadores por los microcontroladores PIC pueden ser su velocidad, precio, facilidad de uso, información y herramientas de apoyo que posee. Esto es lo que da lugar a su imagen de sencillez y utilidad. En el caso de nuestro proyecto nos atuvimos a estas razones para el uso de este tipo de microcontrolador. La razón de inclinarnos por el PIC 16F877A, se debió a que este nos ofrecía el número de pines que nos hacían falta para nuestro proyecto, además que cumplía las exigencias que teníamos para el mismo.

Veamos las características principales más representativas de los PIC:

### **1ª. La arquitectura del procesador sigue el modelo Harvard**

En esta arquitectura, el CPU se conecta de forma independiente y con buses distintos con la memoria de instrucciones y con la de datos.

La arquitectura Harvard permite a la CPU acceder simultáneamente a las dos memorias. Además, propicia numerosas ventajas al funcionamiento del sistema como se irán describiendo.



**2ª. Se aplica la técnica de segmentación (“pipe-line”) en la ejecución de las instrucciones.**

La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de la siguiente. De esta forma se puede ejecutar cada instrucción en un ciclo (un ciclo de instrucción equivale a cuatro ciclos de reloj).

La segmentación permite al procesador ejecutar cada instrucción en un ciclo de instrucción equivalente a cuatro ciclos de reloj. En cada ciclo se realiza la búsqueda de una instrucción y la ejecución de la anterior.

Las instrucciones de salto ocupan dos ciclos al no conocer la dirección de la siguiente instrucción hasta que no se haya completado la de bifurcación.

**3ª. El formato de todas las instrucciones tiene la misma longitud**

Todas las instrucciones de los microcontroladores de la gama baja tienen una longitud de 12 bits. Las de la gama media tienen 14 bits y más las de la gama alta. Esta característica es muy ventajosa en la optimización de la memoria de instrucciones y facilita enormemente la construcción de ensambladores y compiladores.

**4ª. Procesador RISC (Computador de Juego de Instrucciones Reducido)**

Los modelos de la gama baja disponen de un repertorio de 33 instrucciones, 35 los de la gama media y casi 60 los de la alta.

### **5ª. Todas las instrucciones son ortogonales**

Cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.

### **6ª. Arquitectura basada en un banco de registros.**

Esto significa que todos los objetos del sistema (puertos de E/S, temporizadores, posiciones de memoria, etc.) están implementados físicamente como registros.

### **7ª. Diversidad de modelos de microcontroladores con prestaciones y recursos diferentes.**

La gran variedad de modelos de microcontroladores PIC permite que el usuario pueda seleccionar el más conveniente para su proyecto.

### **8ª. Herramientas de soporte potentes y económicas**

La empresa Microchip y otras que utilizan los PIC ponen a disposición de los usuarios numerosas herramientas para desarrollar hardware y software. Son muy abundantes los programadores, los simuladores software, los emuladores en tiempo real, ensambladores, Compiladores C, Intérpretes y Compiladores BASIC, etc.

## **FT245BM**

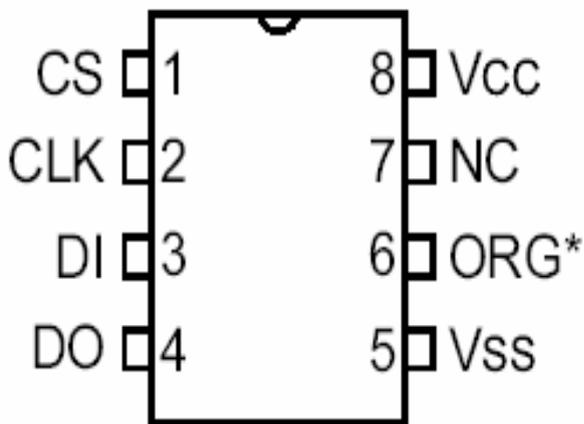
El FT245BM es el dispositivo que utiliza nuestra tarjeta microcontroladora para comunicarse con el PC. En el apartado 5 comentaremos las características principales que tiene este dispositivo.



## MEMORIA EEPROM 93C46B

La memoria EEPROM 93C46B es un dispositivo dedicado principalmente a comunicación en 16 bits. Debido a la avanzada tecnología CMOS con la que está fabricada esta memoria, son especialmente indicados para aplicaciones de memoria no-volátil para baja potencia.

### Descripción funcional



Cuando se conecta la pata ORG\* a  $V_{CC}$ , seleccionamos la organización (x16), mientras que si la conectamos a tierra, se selecciona la de (x8). Las instrucciones, direcciones y datos son introducidos por la patita DI en el pulso de subida del reloj (CLK). La patita DO normalmente se mantiene en un estado de alta impedancia excepto cuando está leyendo datos desde el dispositivo, o cuando está comprobando el estado READY/BUSY durante la programación. El estado READY/BUSY puede ser verificado durante una operación de borrado/escritura mediante un polling de la patita DO: DO a nivel bajo indica que la programación continúa, mientras que cuando DO está a nivel alto el dispositivo está preparado. DO entrará en estado de alta impedancia en el pulso de bajada de CS.

**TABLE 1-1: INSTRUCTION SET FOR X 16 ORGANIZATION (93XX46B OR 93XX46C WITH ORG = 1)**

Instruction	SB	Opcode	Address	Data In	Data Out	Req. CLK Cycles
ERASE	1	11	A5 A4 A3 A2 A1 A0	—	(RDY/BSY)	9
ERAL	1	00	1 0 X X X X	—	(RDY/BSY)	9
EWDS	1	00	0 0 X X X X	—	HIGH-Z	9
EWEN	1	00	1 1 X X X X	—	HIGH-Z	9
READ	1	10	A5 A4 A3 A2 A1 A0	—	D15 - D0	25
WRITE	1	01	A5 A4 A3 A2 A1 A0	D15 - D0	(RDY/BSY)	25
WRAL	1	00	0 1 X X X X	D15 - D0	(RDY/BSY)	25

# ADAPTADOR BASADO EN MICROCONTROLADOR PARA COMUNICACIONES CON EL PUERTO USB PLACA MICROCONTROLADORA

---

## 1.-CONDICIÓN START (COMIENZO)

El bit START es detectado por el dispositivo si CS y DI están ambos a nivel ALTO en el primer pulso de subida de CLK. Antes de que la condición de START es detectada, CS, CLK y DI pueden cambiar en alguna combinación (excepto para el caso de la condición START), sin resultar en una operación del dispositivo (READ, WRITE, ERASE, EWEN, EWDS, ERAL o WRAL) Tan pronto como CS se ponga a nivel alto, el dispositivo no continúa durante más tiempo en modo Standby. Una instrucción que sigue a una condición de START será ejecutada si los bits de código de la operación, dirección y datos se encuentran como deben para esa instrucción en particular.

## 2.-DATA IN/DATA OUT

Es posible conectar las patas DO y DI. Sin embargo, con esta configuración es posible que se produzca un “conflicto de bus” durante el “dummy zero” que precede a la operación READ (lectura), si A0 se encuentra a nivel alto. En estas condiciones el nivel de tensión en DO no estará definido y dependerá de la impedancia relativa entre DO y A0. Cuanto mayor sea la corriente que sale por A0, mayor será la tensión en DO. Para limitar esta corriente debe colocarse una resistencia entre DI y DO.

## 3.-PROTECCIÓN DE DATOS

Todos los modos de operación se inhiben cuando la tensión de alimentación Vcc está por debajo de un voltaje típico de

3.8V para los dispositivos de tipo ‘93C’. Las instrucciones EWEN y EWDS garantizan

**Pin Function Table**

Name	Function
CS	Chip Select
CLK	Serial Data Clock
DI	Serial Data Input
DO	Serial Data Output
Vss	Ground
NC	No internal connection
ORG	Memory Configuration
Vcc	Power Supply

una protección adicional contra una programación accidental durante el funcionamiento en condiciones normales.

Después del encendido, el dispositivo se encuentra automáticamente en modo EWDS (Para una protección añadida, debería ejecutarse una instrucción EWDS después de cada operación de escritura) Por todo ello, debe ejecutarse una instrucción EWEN antes de que pueda ser ejecutada una instrucción ERASE o WRITE.

#### **4.-FUNCIÓN ERASE**

La instrucción **ERASE** obliga a los bits de datos de la dirección especificada a ponerse en estado lógico “1”. CS es llevado a nivel bajo. Este flanco de bajada del pin CS inicia un ciclo de programación “self-time”, excepto para los dispositivos 93C en los que un flanco de subida de CLK antes de “last adress bit” inicia el ciclo de escritura. El pin DO indica el estado READY/BUSY del dispositivo si CS pasa a nivel alto después de cómo mínimo 250 ns a nivel bajo ( $T_{CSL}$ ). DO a nivel bajo indica que la programación continúa aún. DO a nivel alto indica que el registro ha sido borrado en la dirección especificada y el dispositivo está preparado para recibir la siguiente instrucción.

#### **5.-ERASE ALL (ERAL)**

La instrucción **ERAL** borrará el vector completo de memoria poniéndolo a nivel alto. El ciclo de instrucción de **ERAL** es idéntico al ciclo de **ERASE**, excepto por el diferente código de operación (ver fig. XX). El ciclo de **ERAL** es completamente auto temporizado y comienza en el flanco de bajada de CS, excepto en los dispositivos 93C donde el flanco de subida de CLK antes de la carga de último bit de datos inicia el ciclo de escritura. Controlar los tiempos en el pin CLK no es necesario después de que el dispositivo ha entrado en el ciclo **ERAL**.

El pin DO indica el estado READY/BUSY del dispositivo, si CS es llevado a nivel alto después de un mínimo de 250 ns a nivel bajo ( $T_{CSL}$ )

# ADAPTADOR BASADO EN MICROCONTROLADOR PARA COMUNICACIONES CON EL PUERTO USB PLACA MICROCONTROLADORA

---

Nota:  $V_{CC}$  debe ser 4.5V para un correcto funcionamiento de la función **ERAL**.

## 6.-ERASE/WRITE DISABLE y ENABLE

Todos los modos de programación tienen que ir precedidos de una instrucción ERASE/WRITE ENABLE. Una vez que es ejecutada ésta, la programación se encuentra habilitada hasta que una instrucción ERASE/WRITE Disable es ejecutada o es eliminada la alimentación  $V_{CC}$ .

Para proteger contra una perturbación accidental en los datos, la instrucción EWDS puede ser usada para desactivar cualquier función de lectura/escritura, y debe seguir a todas las operaciones de programación.

La instrucción de lectura READ, es independiente de las instrucciones EWEN y EWDS.

## 7.-READ

La instrucción READ extrae el dato de la dirección de memoria localizada en el pin DO. Un “dummy zero” precede siempre a la cadena de salida de 16 bits (debido a que ORG está a nivel alto). Los bits de datos de salida cambian en el flanco de subida de CLK y son estables después del tiempo de retraso especificado ( $T_{PD}$ ) La lectura secuencial es posible cuando CS es mantenido a nivel alto. Los datos de la memoria pasarán al siguiente registro y saldrá secuencialmente.

## 8. WRITE

La instrucción WRITE es seguida por 16 bits (ya que ORG está a nivel alto) de datos los cuales son escritos en la dirección especificada. Después de que el último dato es registrado por DI, el flanco de subida de CS inicia un ciclo de auto-borrado y programación auto temporizado. El pin DO nos indica el estado READY/BUSY del

# ADAPTADOR BASADO EN MICROCONTROLADOR PARA COMUNICACIONES CON EL PUERTO USB PLACA MICROCONTROLADORA

---

dispositivo, si CS es llevado a nivel alto después de un mínimo de 250 ns a nivel bajo ( $T_{CSL}$ ). Do a nivel lógico 0 indica que la programación se encuentra aún en progreso, mientras que un 1 lógico, indica que el registro ha sido escrito en la dirección especificada y el dispositivo está preparado para otra instrucción.

## Descripción de los pines

### Chip Select (CS)

A nivel alto selecciona el dispositivo; a nivel bajo lo deselecta y lo lleva a un estado de Stand-by. Sin embargo, un ciclo de programación que ya está en funcionamiento debe ser completado, independientemente de la señal de Chip Select (CS). Si CS pasa a nivel bajo durante el ciclo de programación, el dispositivo pasará a Standby tan pronto como el ciclo de programación se ha terminado.

CS debe permanecer a nivel bajo un mínimo de 250 ns ( $T_{CSL}$ ) entre dos instrucciones consecutivas. Si CS está a nivel bajo, la lógica interna se mantiene en estado de RESET

### Reloj serie (CLK)

La señal de reloj serie es utilizada para sincronizar la comunicación entre el dispositivo maestro y la EEPROM. Códigos de operación (Opcodes), direcciones y datos son contabilizados en el flanco de subida

CLK puede ser parado en cualquier punto de la secuencia de transmisión (a nivel alto o a nivel bajo), y puede continuarse en cualquier momento con respecto al tiempo de señal de reloj a nivel alto ( $T_{CKH}$ ) y al tiempo de señal de reloj a nivel bajo ( $T_{CKL}$ ). Esto da libertad al master para preparar los códigos de operación, direcciones y datos. CLK es una señal sin interés si CS está a nivel bajo, mientras que si CS está a nivel alto pero la condición START no ha sido aún detectada ( $DI=0$ ), pueden recibirse

# ADAPTADOR BASADO EN MICROCONTROLADOR PARA COMUNICACIONES CON EL PUERTO USB PLACA MICROCONTROLADORA

---

cualquier número de señales de reloj por el dispositivo sin cambiar su status (es decir, esperando a una condición de START).

No son necesarios ciclos de CLK durante la escritura en tiempo propio (self-timed). Después de detectar la condición de START debe proporcionarse un número especificado de ciclos de reloj (transiciones de nivel bajo-alto de CLK) Estos ciclos de reloj son necesarios para sincronizar en todas las opcode, direcciones y datos necesarios antes de que la instrucción sea ejecutada. En este caso CLK y DI se convierten en señales de entrada sin interés, esperando a que sea detectada una nueva condición de START.

## **Data In (DI)**

Data In (DI) es usada para marcar un bit de START, opcodes, direcciones o memorias, en sincronía con la señal CLK de entrada.

## **Data Out (DO)**

Data Out (DO) es usado en el modo READ para sacar datos en sincronía con la señal de entrada CLK ( $T_{PD}$  después del flanco de subida de CLK)

Este pin proporciona también información sobre el status READY/BUSY durante ciclos de ERASE y WRITE. La información de status READY/BUSY está disponible en el pin DO si CS es llevado a nivel alto después de haber estado a nivel bajo un mínimo del tiempo de chip select ( $T_{CSL}$ ) y se ha iniciado una operación de ERASE o WRITE.

La señal de status no estará disponible en DO si CS es mantenido a nivel bajo durante el ciclo completo de ERASE o WRITE. En este caso, DO es un pin en estado de alta impedancia. Si el status es comprobado después de un ciclo ERASE/WRITE, la línea de datos estará a nivel alto para indicar que el dispositivo está preparado.

# ADAPTADOR BASADO EN MICROCONTROLADOR PARA COMUNICACIONES CON EL PUERTO USB PLACA MICROCONTROLADORA

---

Nota: Mediante un bit de START y poniendo CS a nivel bajo borraremos el status READY/BUSY de DO.

## Organization (ORG)

Cuando el pin ORG está conectado a  $V_{cc}$  o a una lógica de nivel alto, se selecciona la organización de memoria (x16). Sin embargo cuando ORG es conectado a  $V_{ss}$  o a una lógica de nivel bajo, se selecciona la organización de memoria (x8).

Para un correcto funcionamiento, ORG debe ser unido al nivel lógico adecuado:

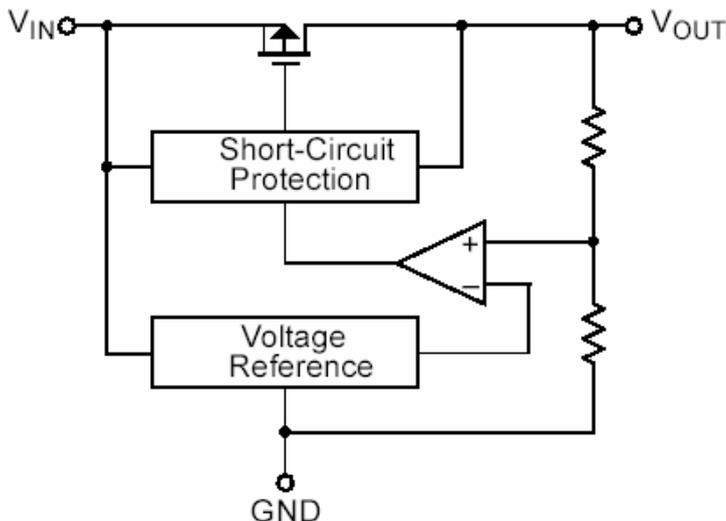
Los dispositivos 93XX46A siempre tienen organización “x8”

Los dispositivos 93XX46B siempre tienen organización “x16”

## REGULADOR DE TENSION TC55

Los reguladores de tensión TC55 componen una colección de reguladores de tensión CMOS positivos con un pequeño voltaje de dropout (LDO) que puede suministrar hasta 250 mA de corriente, con una diferencia de potencial entrada-salida

sumamente baja, de unos 380 mV a 200 mA.



Este pequeño voltaje de dropout, combinado con un bajo consumo de corriente de sólo 1.1  $\mu\text{A}$ , lo hace ideal para funcionar como una batería.

La pequeña diferencia de voltaje (voltaje de dropout), aumenta la

# ADAPTADOR BASADO EN MICROCONTROLADOR PARA COMUNICACIONES CON EL PUERTO USB

## PLACA MICROCONTROLADORA

---

vida del regulador. Ello también permite altas corrientes en pequeños paquetes cuando se manejan mínimas diferencias de potencial  $V_{IN} - V_{OUT}$ . El circuito también incorpora protección contra cortocircuitos para asegurar una fiabilidad máxima.

Otra importante ventaja de los reguladores TC55, consiste en que, a diferencia de los reguladores bipolares, la corriente suministrada por el TC55, no aumenta proporcionalmente a la corriente de la carga.

### Descripción de Pines

**TERMINAL DE TIERRA (GND):** Regulador de tierra. Conectar GND a la parte negativa de la salida y a la parte negativa del condensador de entrada. Solamente la corriente LDO bias (típicamente de  $\mu A$ ) fluye por este pin, por lo que no hay corrientes altas. La regulación de salida LDO se referencia a este pin. Debe minimizarse la caída de tensión entre este pin y la parte negativa de la carga.

**TENSIÓN DE SALIDA REGULADA ( $V_{OUT}$ ):** Conectaremos  $V_{OUT}$  a la parte positiva de la carga, y al terminal positivo del condensador de salida. El lado positivo del condensador de salida, debería estar situado tan cerca de la patita LDO  $V_{OUT}$  como sea factible. La corriente que sale por esta patita será igual a la corriente que fluirá por la carga.

**TABLE 3-1: PIN FUNCTION TABLE**

Pin No.	Symbol	Description
1	GND	Ground Terminal
2	$V_{OUT}$	Regulated Voltage Output
3	$V_{IN}$	Unregulated Supply Input

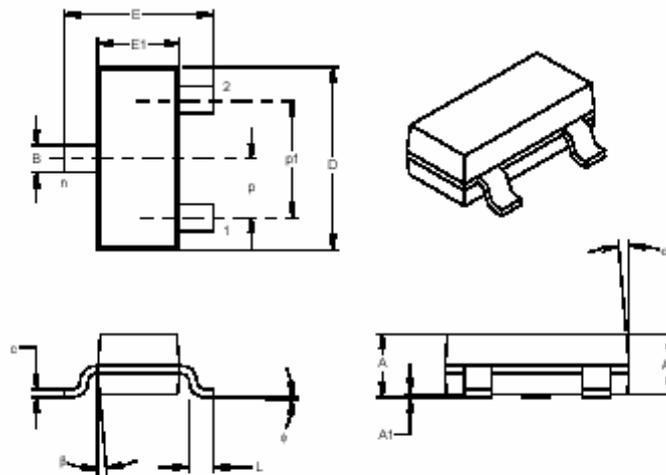
**TENSIÓN DE ENTRADA NO REGULADA ( $V_{IN}$ ):** Conectamos la entrada del voltaje de alimentación y el lado positivo de la capacidad de entrada a la patita  $V_{IN}$ . El condensador de entrada debe estar localizado tan cerca de  $V_{IN}$  como sea posible. La corriente que entra por esta patita es igual a la corriente que fluye por la carga, más la corriente LDO bias (típicamente  $1\mu A$ )

# ADAPTADOR BASADO EN MICROCONTROLADOR PARA COMUNICACIONES CON EL PUERTO USB PLACA MICROCONTROLADORA

---

## Información sobre el encapsulado

El encapsulado que se corresponde con el que hemos utilizado para la realización física de nuestra placa es el del SOT-23, cuyas dimensiones podemos ver en la figura adjunta.



Dimension Limits	Units	INCHES*			MILLIMETERS		
		MIN	NCM	MAX	MIN	NCM	MAX
Number of Pins	n		3			3	
Pitch	p	.098				0.96	
Outside lead pitch (basic)	p1		.076			1.92	
Overall Height	A	.095	.040	.044	0.89	1.01	1.12
Molded Package Thickness	A2	.095	.037	.040	0.88	0.96	1.02
Standoff §	A1	.000	.002	.004	0.01	0.06	0.10
Overall Width	E	.083	.093	.104	2.10	2.37	2.64
Molded Package Width	E1	.047	.051	.055	1.20	1.30	1.40
Overall Length	D	.110	.115	.120	2.80	2.92	3.04
Foot Length	L	.014	.018	.022	0.35	0.45	0.55
Foot Angle	φ	0	5	10	0	5	10
Lead Thickness	c	.004	.006	.007	0.09	0.14	0.18
Lead Width	B	.015	.017	.020	0.37	0.44	0.51
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

\* Controlling Parameter  
§ Significant Characteristic

Notes:  
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.  
JEDEC Equivalent: TO-236  
Drawing No. C04-104

## CRISTALES OSCILADORES

Los dos dispositivos principales de la tarjeta microcontroladora (PIC y FT) llevan un tanque resonante consistente en un cristal oscilador y un par de condensadores en cada pata a tierra. Este tanque resonante tiene como misión producir una frecuencia de resonancia bastante exacta cuando se coloca en un circuito inestable.

# ADAPTADOR BASADO EN MICROCONTROLADOR PARA COMUNICACIONES CON EL PUERTO USB PLACA MICROCONTROLADORA

---

En nuestra placa microcontroladora, el PIC lleva un cristal oscilador de 20 MHz con condensadores de 27 pF, mientras el FT lleva un cristal oscilador de 6 MHz y condensadores de 27 pF.

Estos dos cristales osciladores es conveniente ponerlos lo más próximo posibles al correspondiente dispositivo al que van unidos, e igualmente conviene colocar los dos condensadores simétricamente al cristal, para que la frecuencia resonante sea lo más exacta que se pueda.

## CONECTORES

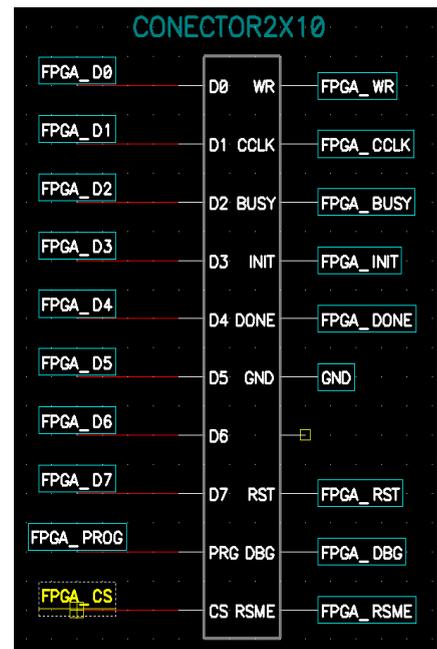
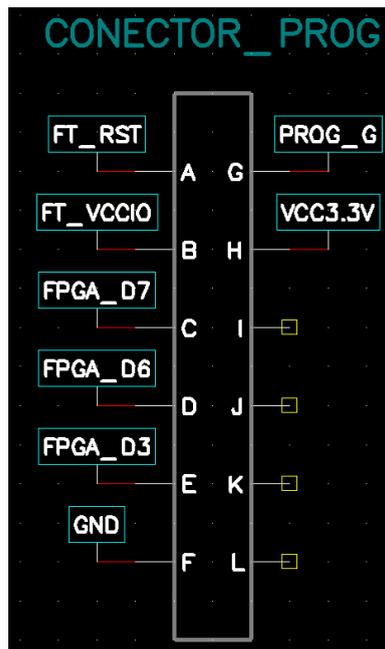
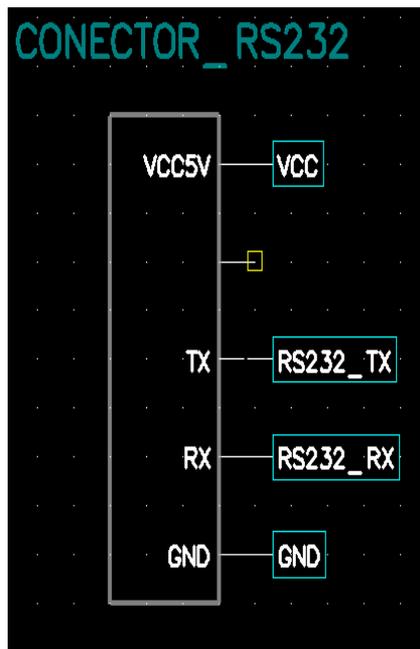
Aunque realmente no se trate de hardware en sí mismo, incluiremos en este apartado los conectores que se encuentran en la placa. Concretamente son tres:

- Un conector de 20 pines, que nos unirá la tarjeta microcontroladora con el hardware de UNSHADES, y permitirá comunicarnos con él, para realizar la depuración.
- Un conector para la programación del PIC, y que se unirá al grabador JDM, que es explicado en otro apartado dentro de este proyecto
- Por último se ha añadido un conector que permite comunicar la tarjeta vía RS232. Este conector se ha utilizado dentro de este proyecto para los primeros programas de aplicación, y comprobar que tanto el programador como la tarjeta funcionaban correctamente.

Podemos ver estos tres conectores en las figuras que se muestran a continuación:

# ADAPTADOR BASADO EN MICROCONTROLADOR PARA COMUNICACIONES CON EL PUERTO USB PLACA MICROCONTROLADORA

---



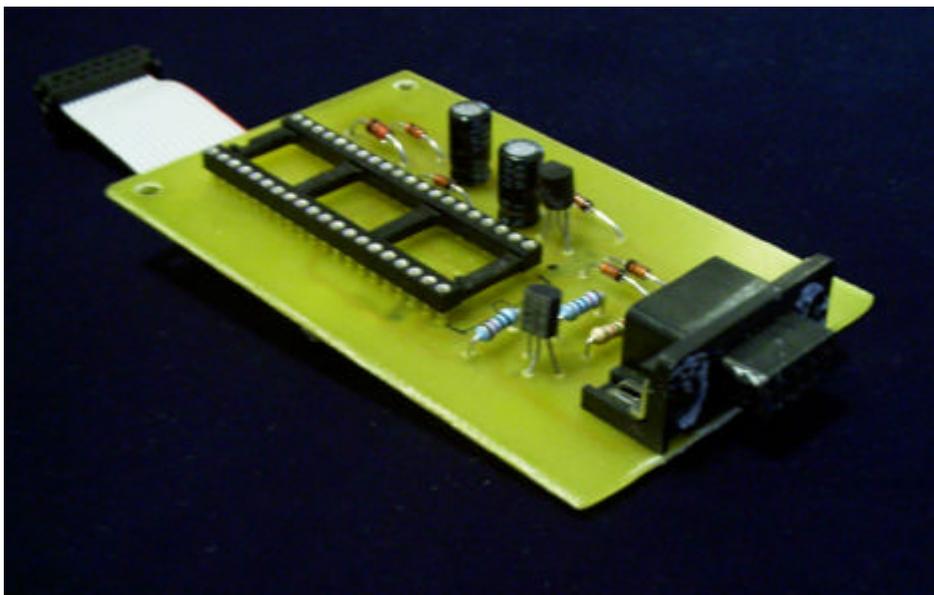
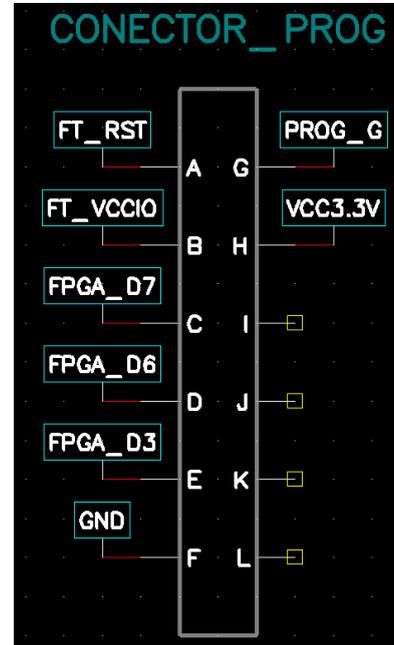
# **PLACA PROGRAMADORA**

## DISEÑO DE LA PLACA PROGRAMADORA

Para realizar la programación del PIC 16F877A de nuestra tarjeta microcontroladora, utilizaremos un programador JDM. Este tipo de grabador es totalmente compatible con los microcontroladores de la familia de los PIC, y en particular con el PIC utilizado en nuestro proyecto, el 16F877A.

Se trata de un circuito bastante simple, con una interfaz RS232, a través del cual se obtiene también la alimentación para el circuito grabador, por lo que no es necesaria una fuente externa.

Como nuestro PIC es de tipo SMD, debemos sacar las señales necesarias para la grabación de nuestro PIC y la llevaremos hasta un conector que se encuentra en nuestra tarjeta para grabar el PIC. El conector que va en nuestra tarjeta microcontroladora se muestra en la imagen de la página anterior.

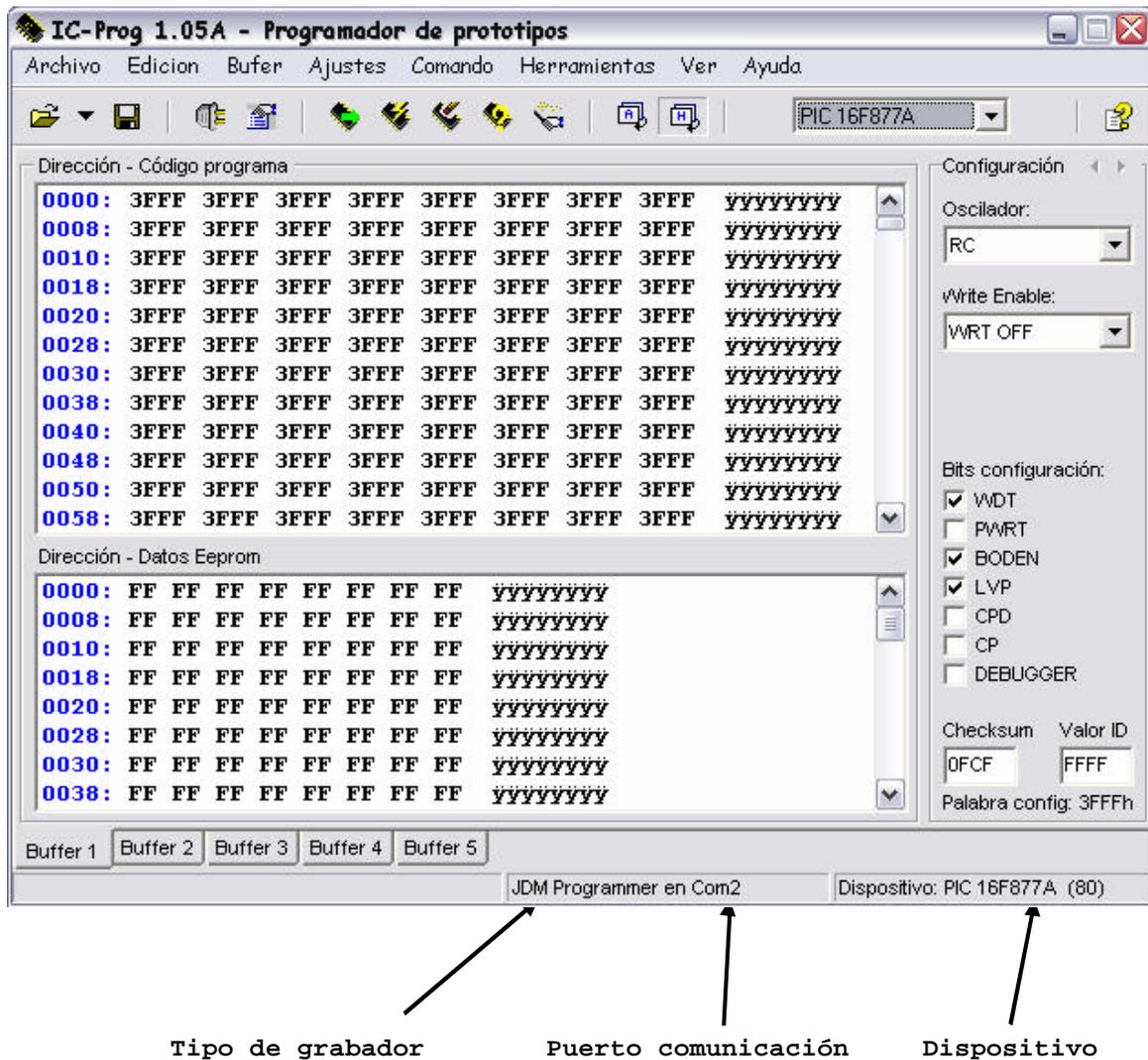


Se trata además de un grabador de uso muy sencillo, y que suele ofrecer muy buenos resultados.



# ADAPTADOR BASADO EN MICROCONTROLADOR PARA COMUNICACIONES CON EL PUERTO USB

## PLACA PROGRAMADORA



En primer lugar hemos de hacer la configuración de hardware. Para ello nos vamos al menú de **Ajustes** y dentro de éste, al **Tipo hardware** (también es posible entrar en este menú directamente pulsando **F3**) Allí hemos de elegir el tipo de hardware/grabador a utilizar, y también el puerto por el que se realizará la comunicación. Es posible también elegir otras opciones, pero en principio no les haremos mucho caso.

Tras la configuración introducimos el programa que queremos cargar en el microcontrolador. Este programa debe estar compilado en formato hexadecimal (es obligatorio debido a que el PIC que utilizamos es de gama media: PIC16F877A) El programa, tras escribirlo, lo hemos compilado con el MPLAB, y por tanto se debe encontrar ya en formato hexadecimal (.HEX)

# ADAPTADOR BASADO EN MICROCONTROLADOR PARA COMUNICACIONES CON EL PUERTO USB PLACA PROGRAMADORA

---

Los bits de configuración (WDT, PWRT, BODEN,...), son seleccionados directamente al cargarse el programa, ya que en el mismo programa hemos elegido si deben o no estar activos. Estos bits de configuración son los siguientes:

- **Oscilador:**
  - **LP:** Oscilador bajo consumo (32-200 KHz)
  - **XT:** Oscilador estándar (100 KHz-4MHz)
  - **HS:** Oscilador de alta velocidad
  - **RC:** Oscilador de bajo costo
  
- **Protección código:** Para el PIC16F877A que tiene una memoria de código de 8K
  - **CP 0000h-1FFFh:** Protección total del código
  - **CP 1000h-1FFFh:** Protección para ½ del código (parte alta)
  - **CP 1F00h-1FFFh:** Protección para ¼ del código (parte alta)
  - **CP OFF:** Sin protección para el código
  
- **Bits de configuración:**
  - **PWRT:** Activación del Timer de Conexión de Alimentación.
  - **BODEN:** Activación del Reset por Caída de Tensión.
  - **LVP:** Activación de la programación en Bajo Voltaje.
  - **CPD:** Activación del código de protección de la memoria EEPROM.
  - **WRT:** Activación del permiso de escritura en la memoria FLASH.
  - **DEBUGGER:** Activación del Modo Depurador en Circuito.

Una vez que hemos cargado el programa en el IC-Prog, pasamos a programar físicamente el microcontrolador. El proceso físico es el siguiente. Conectamos a través del puerto serie que hayamos elegido en el **Tipo Hardware** nuestro grabador JDM, y a

## **ADAPTADOR BASADO EN MICROCONTROLADOR PARA COMUNICACIONES CON EL PUERTO USB PLACA PROGRAMADORA**

---

su vez, el conector que hemos sacado del mismo, nos lo llevamos hasta el conector que tenemos en nuestra tarjeta microcontroladora, y que tiene los pines que hemos mostrado al principio de este apartado sobre la tarjeta programadora.

Dos cosas son muy importantes para el momento de la programación de nuestra tarjeta microcontroladora:

- En primer lugar, la tarjeta no debe estar nunca conectada al PC por el puerto USB durante la programación de la tarjeta. La razón es que el programador utiliza tensiones relativas a tierra entre 0 y  $-5V$ , en lugar de los 5 y 0V del USB. Si estuviesen conectados los dos al mismo tiempo, estaríamos conectando una fuente de  $-5V$  directamente a tierra.
  
- En segundo lugar, debemos quitar los jumpers que van colocados en el conector de programación para realizar la programación, que deben ser colocados de nuevo posteriormente, ya que estos jumpers están entre el FT y PIC para cortar VDD-VDD y RST-MCLR/VPP