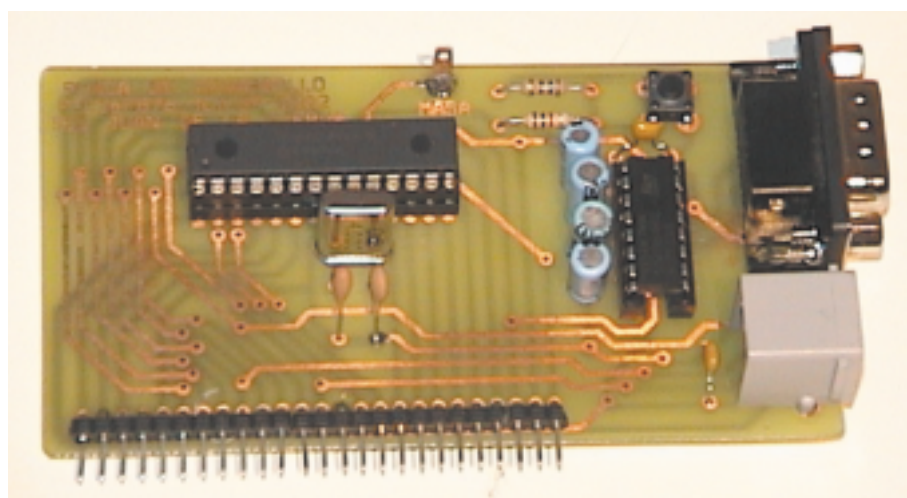


# CURSO DE MICROCONTROLADORES PIC16F87X (...y III)

Fernando Remiro Domínguez  
Profesor de Sistemas Electrónicos  
IES. Juan de la Cierva  
[www.terra.es/personal/fremiro](http://www.terra.es/personal/fremiro)



Programador para PIC16F87X con bootloader.

**S**i habéis utilizado un programador por el puerto serie del ordenador para programar los PIC16F87X, utilizando el maravilloso software libre IC-Prog, habréis comprobado que el tiempo de carga del PIC16F876, por ejemplo, que tiene 8K es de varios minutos que se hacen interminables, sobretodo cuando estamos programando y depurando programas.

Lo que presentamos en este curso es un circuito que a través de la USAR de los PIC 16F87X es capaz de ponerse en comunicación con ellos y cargar en la memoria de programa el código

hexadecimal del programa que queremos que se ejecute. Esto reduce considerablemente el tiempo de programación ya que solo carga los códigos del programa que queremos cargar. La placa que hemos diseñado además de utilizarla como programadora, se puede utilizar como placa de desarrollo, ya que se dispone de acceso a todos y cada uno de los pines del microcontrolador a través de la tira de postes para circuito impreso macho acodados y dispone de un circuito oscilador a cristal en nuestro caso a 4 MHz, tal y como se puede ver en el circuito de la Figura 1.

## DESCRIPCIÓN DEL CIRCUITO

El circuito consta de un microcontrolador PIC16F876 al que se le han conectado un cristal de cuarzo de 4MHz y los condensadores de desacoplo de 27 pF y un pulsador que cuando se activa pone un nivel bajo en la entrada MCLR para provocar un RESET del sistema. Además para poder conectar la USART del microcontrolador al ordenador, se ha utilizado un circuito adaptador de niveles MAX232, tal y como indica el fabricante en sus notas de aplicación, y por último se le ha dotado de un conector acodado DB-9 macho para poder conectar el circuito al puerto serie

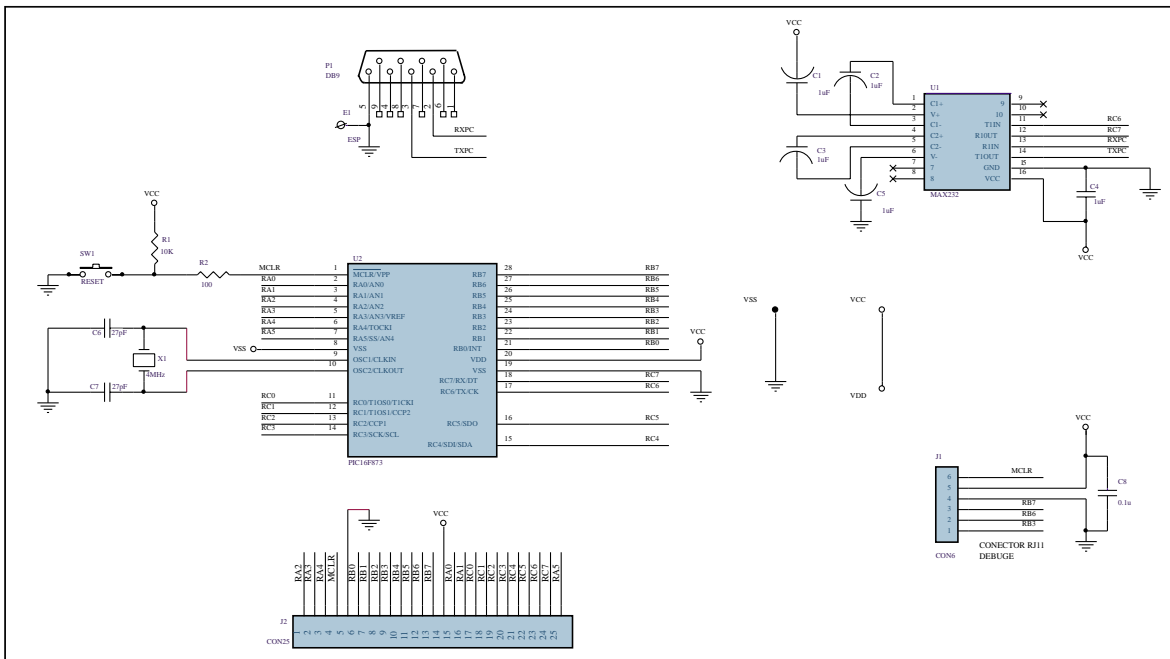


Figura 1.- circuito de la placa de desarrollo y de programación con el bootloaer

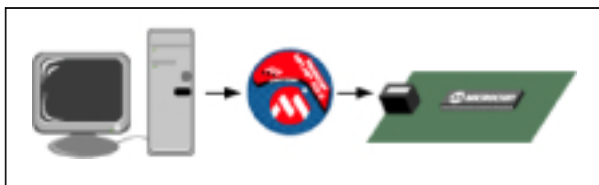


Figura 2.- Conexión del ICD a la placa de desarrollo.

del ordenador. El cable de conexión debe ser un modem nulo, es decir, un cable que solo necesita la línea de masa conectada al terminal 5 de los dos conectores y se cruzan las líneas 2 de un extremo (RxD) con la 3 del otro (TxD) y viceversa. Ver el diagrama de conexión de la Figura 3.

Además la placa dispone de un conector ARJ11, conectado a las líneas MCLR, RB3, RB6 y RB7 del microcontrolador, lo que permite realizar la conexión a un módulo ICD (Depurador en circuito), que como veremos es una herramienta de bajo costo

que conectada al PC y en el mismo entorno del MPLAB, permite la visualización de los registros del microcontrolador en los puntos de ruptura que se establezcan. Pudiéndose ejecutar los programas en tiempo real o en modo paso a paso.

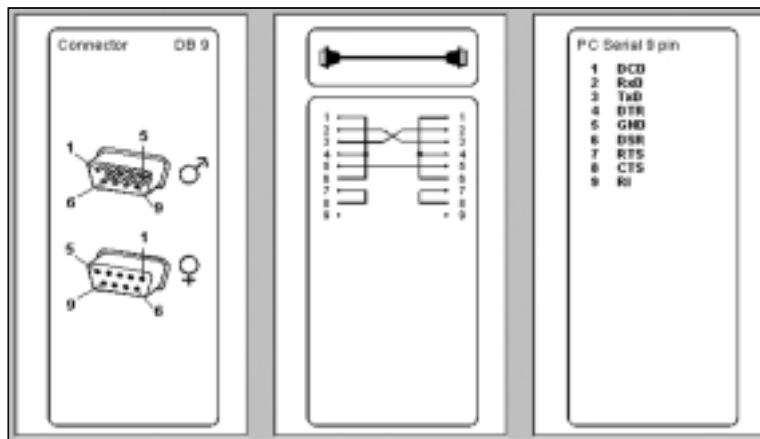


Figura 3.- Cable de conexión MODEM nulo para la conexión de la placa de desarrollo con el PC.

```

===== Sección fijada por el usuario =====
list    p=16f876    ; <<< fijar el tipo de microcontrolador

;Poner el mismo microcontrolador en el proyecto
#include <p16f876.inc> ; <<< poner el archivo *.inc del microcontrolador seleccionado

#define FOSC D'4000000' ; <<< Poner la frecuencia del cuarzo (Hz), max. 20 MHz
#define BAUD D'9600' ; <<< poner la velocidad de transmisión (Bd)
#define BAUD_ERROR D'4' ; <<< Poner la tasa de error (%)
#define TIME ; <<< Poner el método inicio del bootloader PIN/TIME/AUTO

;Disparo por puesta a nivel bajo de la señal de disparo
;TIME Inicia la recepción del byte de IDENT después el RESET
#define POR PORTB ; <<< solo para el PIN – del puerto en juego
PORTB/PORTC/PORTD/PORTE
#define PORT_BIT 7 ; <<< solo para el pin PIN – poner el bit de la línea
seleccionada 0/1/2/3/4/5/6/7
#define TIMEOUT D'3' ; <<< solo para el TIME – fijar el tiempo(0.1 s), max. 25.5 s

===== Fin de la sección fijada por el usuario =====
    
```

Tabla 1.- Sección del programa bootloader que debe fijar el usuario.

**COMO UTILIZAR EL PROGRAMA BOOTLOADER**

1. En primer lugar debemos obtener el programa en la siguiente web [http://www.ehl.cz/pic/pic\\_e.htm](http://www.ehl.cz/pic/pic_e.htm), donde también se puede obtener el PIC donwloader V1.8.
2. Abrir el proyecto bootldr.pjt utilizando el programa Microchip MPLAB y cambiar los parámetros de la sección de usuario marcados dentro del ensamblador con las marcas <<<, tal y como se muestra en la Tabla 1, en la que hemos fijado el microcontrolador PIC16F876, con un oscilador de 4MHz, una velocidad de transmisión de 9600 baudios y utiliza el pin 7 del PORTB como señal de disparo para ejecutar el bootloader cuando está a nivel bajo y correr el programa de usuario cuando está a nivel alto. Por tiempo entra en el programa de usuario después de haber pasado 0,3 segundos sin recibir la palabra de IDENTIFICACIÓN.

3. Si se modifica alguno de los parámetros anteriores, debemos de volver a compilar el programa bootldr.asm.
4. Programar el microcontrolador con el archivo bootldr.hex utilizando cualquier programador de los que ya se han visto en esta Revista, por ejemplo el que el que aparece en el módulo de aprendizaje de microcontroladores PIC que apareció en el número 189. Poniendo los bits de configuración para que el microcontrolador trabaje en la aplicación.
5. Recompilar el programa de la aplicación para que no borre el bootloader. El bootloader usa los 256 bytes de la parte alta de memoria de programa (0x1F04-0x1FFF). Llegado a este punto lo más

acaloratorio es ver un ejemplo con un programa que utiliza este método de carga del programa de usuario en el microcontrolador.

El programa TMR0\_1.asm es un programa ejemplo que utiliza una temporización utilizando TMR0 sin utilizar las interrupciones. Realiza una temporización de 32,77 ms que se emplea para hacer parpadear los diodos LED pares o impares respectivamente conectados al PORTB. En el código de este programa con separados por un ";" (es decir, como un comentario) se encuentran las instrucciones de salto a INICIO que utilizaríamos con un programador convencional. Y al final del programa, en la posición de memoria 0x1F00 hemos escrito:

```

ORG    0x1F00
bcf    PCLATH,4 ;Selecciona el banco 0 de memoria
bcf    PCLATH,3
goto   INICIO
    
```

```

;*****
; Programa          TMR0_1.ASM          Fecha : 7- Mayo-2002
; Este programa utiliza una temporización utilizando TMR0 sin utilizar las interrupciones.
; Se realiza una temporización de 32,77 ms que se emplea para hacer parpadear los diodos LED
; pares o impares respectivamente conectados al PORTB
; Programa preparado para cargar con el PIC downlader
; Revisión : 0.1          Programa para PIC16F87X
; Velocidad del Reloj: 4 MHz          Reloj Instrucción: 1 MHz = 1 uS
; Perro Guardián:      Deshabilitado      Tipo de Reloj : XT
; Protección del código: OFF
;*****
List          p=16F876          ;Tipo de procesador
include      "P16F876.INC"      ;Definiciones de registros internos

;ORG          0x00          ; Inicio del programa en
;goto        INICIO

ORG          0x05          ;salva el vector de Interrupciones
INICIO      bcf          STATUS,RP1
           bsf          STATUS,RP0          ; Banco 1
           movlw       b'11010111'        ; Valor a cargar en OPTION_REG
           movwf      OPTION_REG          ; predivisor 256 asignado a TMR0
           clrf       TRISB          ; PORTB como salida
           bcf       STATUS,RP0          ; Banco 0
           clrf       PORTB          ; Apaga los LEDS del PORTB

BUCLE      movlw       b'01010101'
           movwf     PORTB          ; Enciende los LED impares del PORTB
           call      RETARDO          ; Llamada a subrutina de RETARDO
           movlw     b'10101010'
           movwf     PORTB          ; Enciende los LED pares del PORTB
           call      RETARDO
           goto      BUCLE

RETARDO    clrf       TMR0          ; TMR0 = 0 y empieza su incremento
EXPLORA    btfss     TMR0,7          ; Es el bit 7 de TMR0=1?
           goto      EXPLORA          ; No ha llegado TMR0 a 128 Decial
           return          ; Ha llegado TMR0 al valor 128D y retorna
           ; al programa principal

ORG        0x1F00
           bcf       PCLATH,4          ;Selecciona el banco 0 de memoria
           bcf       PCLATH,3
           goto      INICIO          ;Salta a INICIO
END
    
```

Tabla 2.- Programa TMR0-1.asm para cargar con el bootlader.

Es decir, en las posiciones de memoria 0x1F00-0x1F03, escribimos un salto largo de 14 bits a la posición de memoria INICIO, que está en el banco cero de memoria de programa. Esto último debe verificarse si se utiliza un compilador en C.

6. Recompile el programa de la aplica

**COMO REALIZAR LA CARGA EN EL MICROCONTROLADOR**

1. En primer lugar se conecta la placa de desarrollo al PC mediante un cable de modem

nulo, tal y como se comento anteriormete.

2. Para realizar la carga del programa de usuario, vamos a utilizar un programa de "carga" freeware, se trata del PIC downloader de Petr Kolomaznik y que hemos con-

MICROCONTROLADORES  
 57  
 RESISTOR

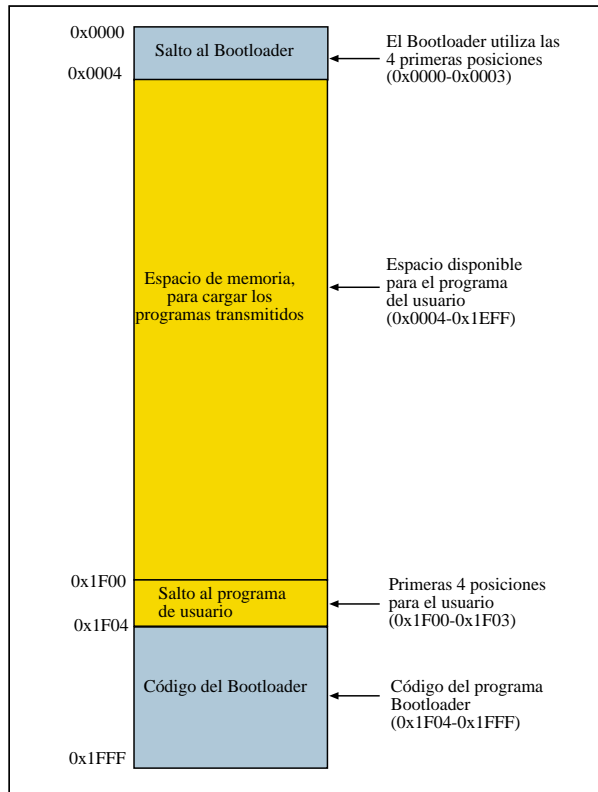


Figura 4.- Mapa de memoria del Bootloader para un microcontrolador de 8KBytes.

**Características del programa PIC downloader 1.8 :**

- Compatible con el bootloader de HI-TECH's o de Shane Tolmie para los microcontroladores PIC16F87x
- Funcionamientos bajo Windows 9x/ME/NT/2000
- Selección de la velocidad a partir del 2400 a 56000 Bd
- Selección del puerto serie entre el COM1-6
- Trabaja con datos de EEPROM en el archivo hexadecimal
- No envía la palabra de los configuración a la posición 0x2007 o palabra de la identificación desde el bootloader
- Mantiene las últimas configuraciones
- Se incluye el código de fuente
- El downloader del PIC se controla al realizar una señal de disparo del RESET en el microcontrolador
- Se trabaja en ensamblador para todos aquellos programadores que no utilizan el lenguaje C. El bootloader es muy fácil de recompilar para otros microcontroladores actualizando el pin/tiempo y la frecuencias del cristal de cuarzo. Solo se necesita el programa MPLAB de Microchip que también es gratuito.
- Se puede obtener en las siguientes webs: (<http://www.ehl.cz/pic/>) (<http://www.microchip.com/>)



Figura 5.- Apariencia de la pantalla que aparece al ejecutar el programa PIC downloader.

seguido en la web <http://www.microchip.com/>. Una vez instalado, al ejecutarlo, aparece una pantalla como la que se muestra en la Figura 5.

3. Seleccionamos la velocidad de transmisión en baudios, en nuestro caso 9600, el puerto serie de comunicación, en el ejemplo hemos seleccionado el COM2 y buscamos el nombre del programa en código hexadecimal que queremos cargar en el microcontrolador, TMR0-1.hex. Este programa tiene las siguientes teclas de función rápida:  
F2 - Búsqueda  
F4 - Escribir  
ESC - Cancelación  
F10 - Salida
4. Seguidamente se pulsa el botón de RESET y el programa se carga en el microcontrolador PIC. Según se va realizando la carga, aparece un indicador de carga (línea azul) y al finalizar la carga suena una señal acústica.
5. Si volvemos a actuar manualmente sobre el pulsador de RESET, se ejecutará el programa de usuario.