

Programación y diseño de dispositivos mediante Microcontroladores PIC.

Dr. Eugenio Martín Cuenca

Ing. Jose María Moreno Balboa

Facultad de Ciencias. Universidad de Granada.

E-mail:emartin@goliat.ugr.es

¿ QUE NECESITAMOS PARA COMENZAR ?

En el número anterior se introdujo el concepto de microcontrolador, haciendo una breve referencia a los temas que se tratarán en esta sección. Se prestará especial interés a la programación de los microcontroladores en lenguajes de alto nivel como son el BASIC y el C, para facilitar la introducción al tema del mayor numero de lectores. No por ello se dejará totalmente de lado la programación en ensamblador.

Comenzamos inicialmente con los microcontroladores PIC. ¿ Pero con cual o cuales de entre ellos ?. Hemos elegido el microcontrolador PIC16F84, por las características que se explicaran mas adelante.

Muy bien ya disponemos de nuestro microcontrolador PIC16F84 (Figura 1), que es un chip de dieciocho pines, y ahora, ¿ Que hacemos con él?, ¿Como lo hacemos funcionar?, ¿Como introducimos el programa dentro del mismo?.

DESCRIPCIÓN PINES PIC16F84 / 16C84				
Nombre	Nº	Tipo I/O/P	Tipo Buffer	Descripción
OSC1 / CLKIN	16	I	ST / CMOS	Entrada oscilador a cristal / fuente externa de reloj
OSC1 / CLKOUT	15	O	--	Salida oscilador a cristal. Conectar al cristal de cuarzo
MCLR	4	I / P	ST	Reset a nivel 0. Entrada voltaje de programación
RA0	17	I / O	TTL	El Puerto A es bidireccional El pin RA4 del Puerto A, si se programa como salida es de colector abierto. Como entrada puede seleccionarse en funcionamiento normal o como entrada del contador / temporizador TMR0 es decir TOCKI.
RA1	18	I / O	TTL	
RA2	1	I / O	TTL	
RA3	2	I / O	TTL	
RA4/TOCKI	3	I / O	ST	
RB0/INT	6	I / O	TTL / ST	El pin 0 del Puerto B, RB0 puede programarse como entrada de interrupciones externas INT.
RB1	7	I / O	TTL	
RB2	8	I / O	TTL	El Puerto B es bidireccional. Mediante software y cuando se programa como entrada pueden activarse resistencias de polarización internas.
RB3	9	I / O	TTL	
RB4	10	I / O	TTL	Los pines RB4 a RB7 pueden programarse para que respondan a interrupción por cambio de estado.
RB5	11	I / O	TTL	
RB6	12	I / O	TTL / ST	
RB7	13	I / O	TTL / ST	Modo programación entrada RELOJ
Vss	5	P	--	Modo programación, DATOS
Vdd	14	P	--	Pin de masa GND
				Pin de alimentación positiva

Figura 1.- Breve descripción y disposición de patillas del PIC 16F84 y 16C84.

Las respuestas a estas preguntas son al menos dos. Por un lado necesitamos un dispositivo que permita grabar el programa elaborado dentro del PIC16F84, es decir un **grabador** o programador. Además se necesita un compilador que produzca el código ejecutable de dicho programa. Una vez introducido el programa en la memoria del microcontrolador, este no funcionará así porque sí, necesita un mínimo de componentes para poder hacerlo (Figura 2).

Los autores hemos pensado en ustedes y desde estas páginas se proporcionarán al lector las herramientas imprescindibles para poder realizar, no solo los dispositivos que se presentan, sino que también podrán finalizar con éxito sus propios proyectos.

Para dicho fin se ha diseñado y realizado un sistema de aprendizaje modular, que consta como mínimo del grabador **MultiPIC** (foto 2), el **Módulo-01**(foto 1) que contiene el microcontrolador y el **Módulo de aprendizaje** (foto 3) . Además el software se ha implementado para su uso tanto bajo MS-DOS,

lo que permite que el lector pueda emplear cualquier ordenador anticuado que aun conserve y bajo Windows 95/98. Todo este material se irá presentando a lo largo de la serie de artículos.

Las herramientas imprescindibles son:

- Si trabaja bajo MS-DOS, un editor de texto ASCII como el **EDIT**. Si trabaja con Windows 95/98, el programa del grabador MultiPIC ya incorpora un editor ASCII.
- Ensamblador **MPASM** para MS-DOS o el **MPLAB** para Windows que proporciona gratuitamente la empresa Microchip.
- Un compilador PicBasic o C.
- Grabador o programador **MultiPIC**.
- Software de grabación **MultiPIC** ya sea en su versión para DOS o Windows 95/98. La versión de Windows incluye un editor ASCII así como el compilador ensamblador MPLAB que proporciona Microchip.
- Un microcontrolador PIC16F84 junto con la electrónica necesaria para hacerlo funcionar como por ejemplo el **Módulo-01**.
- Opcionalmente el *Módulo de aprendizaje*.

Se colocarán en la página Web los programas ensamblador MPASM y MPLAB. Es intención de los autores atender la mayor cantidad de consultas o dudas, pero dada la imposibilidad de una atención individualizada y personal, se creará una pagina en la Web donde se responda a las preguntas mas frecuentes. No dude en mandar un e-mail con cualquier duda o consulta.

EL PIC16F84. MÓDULO - 01.

Comencemos por el principio. El microcontrolador **PIC 16F84** pertenece a la gama media del fabricante y posee unas características optimas para su empleo en el aprendizaje.

Este microcontrolador dispone de memoria FLASH y está fabricado con tecnología CMOS. Su consumo es muy reducido siendo un circuito completamente estático por lo que su reloj puede detenerse completamente sin pérdida de los contenidos de los registros. El consumo típico es de 2 mA a 4 MHz y unos 40 μ A funcionando en modo Sleep.

El encapsulado de 18 pines, así como la descripción del patillaje puede verse en la **Figura 1**. Existe un modelo de 4 MHz el PIC16F84-04/P y otro de 10 MHz el PIC16F84-10/P. Para el desarrollo de las prácticas se ha diseñado el Módulo-01, que se muestra en la foto 1, mientras en la figura 2, se presenta el esquema eléctrico del mismo para aquellos aficionados a la electrónica y al soldador.

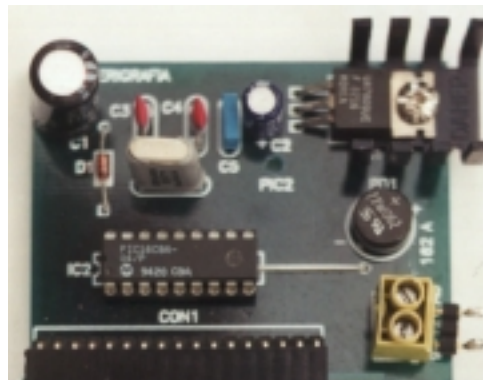


Foto 1.-Imagen del KIT del Módulo – 01, montando un PIC16C84. Se puede apreciar el conector CN1.

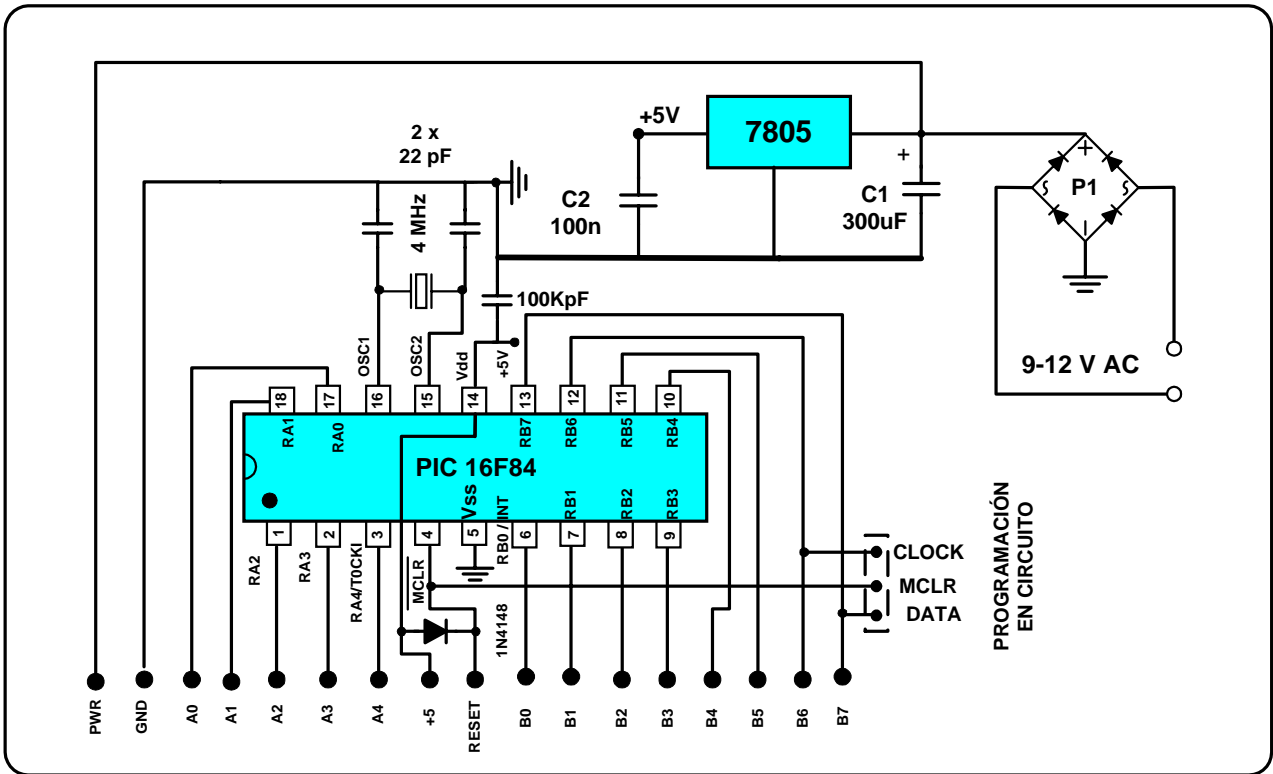


Figura 2.- Esquema eléctrico del Módulo – 01.

Se propone en primer lugar, el Módulo - 01 ó comodín, es decir este módulo se emplea en casi todas las experiencias a realizar. Contiene sólo y exclusivamente el microcontrolador PIC16F84 así como la circuitería necesaria para su alimentación partiendo de una entrada en alterna de 9-12 V A.C. o desde una pila de 9V. Incluye además un conector de 17 pines (CN1) donde se encuentran todas las líneas útiles del microcontrolador así como las señales de masa, +5V y el voltaje rectificado y filtrado del transformador PWR.

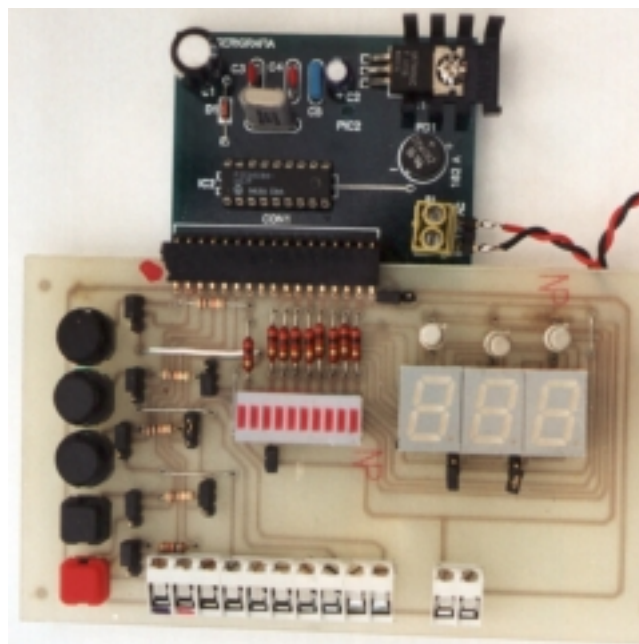


Foto 3.- Módulo – 01 y Módulo de aprendizaje interconectados a través del conector CN1.

Con la sola sustitución del cristal de cuarzo y dos condensadores puede hacerse funcionar el microcontrolador a 4 o 10 MHz. Los módulos empleados en los ejemplos funcionan a 4 MHz, por lo tanto si el lector los usa a una velocidad de 10 MHz, debe tener en cuenta esta mayor velocidad a la hora de escribir sus programas.

ARQUITECTURA INTERNA (Colocar este apartado dentro de un Recuadro)

Las altas prestaciones de los microcontroladores PIC derivan de las características de su arquitectura (Figura 3). Están basados en una Arquitectura tipo **Hardvard** es decir, estos microcontroladores disponen de dos memorias independientes: una, que contiene sólo instrucciones y otra, sólo datos. Ambas poseen sus respectivos sistemas de buses y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas. Esto los hace más rápidos que los microcontroladores basados en la arquitectura tradicional de Von Neumann, permitiendo además que las instrucciones tengan un tamaño 14 bits. Otra característica es su juego de instrucciones reducido (35 instrucciones) **RISC** que en su mayoría se ejecutan en un solo ciclo de reloj excepto las instrucciones de salto que necesitan dos ciclos.

Los microcontroladores PIC y por tanto el **16F84** poseen dos bloques de memoria separados, la *memoria de programa* y los *bancos de registros*. (Figura 3).

La *memoria de programa* está organizada con palabras de 14 bits con un total de 1K, del tipo EEPROM, y en funcionamiento es de sólo lectura. Únicamente se ejecutará el código contenido en esta memoria, pudiéndose además almacenar en ella una cantidad limitada de datos como parte de la instrucción **RETLW**. En una sola palabra se agrupan el código de la instrucción y el operando o su dirección.

Los *bancos de registros* RAM son de 8 bits (byte) a excepción del contador del programa que es de 13 bits. Como puede verse en la Fig. 3, las direcciones bajas contienen los registros especiales y los registros de recursos, encontrándose a continuación los registros de propósito general. El **PIC16F84** posee 68 registros que van de la dirección \$00 a la \$4F, aunque no todas estas direcciones se usan. El **PIC16C84** solo posee 36 registros RAM; esta es una de las pocas diferencias entre el PIC16C84 y el PIC16F84.

El **vector de reset** se encuentra en la posición 0000h y el de **interrupciones** en la 0004h, por lo que la memoria de usuario propiamente dicha se extiende desde la dirección 0005h a la 03ffh.

La *pila* es de 8 niveles. No existe ninguna bandera que indique que esté llena, por lo que será el programador el responsable de controlar que no se produzca su desbordamiento.

Posee además una **memoria EEPROM de datos** de 8 bits. Esta no forma parte del espacio normal direccionable, y solo es accesible en lectura y escritura a través de dos registros para los datos el **EEDATA** que se encuentra en la posición 0008h del banco de registros RAM y para las direcciones el **EEADR** en la 0009h. Para definir el modo de funcionamiento de esta memoria se emplean dos registros especiales el **EECON1** dirección 0088h y el **EECON2** en 0089h. Esta memoria EEPROM no emplea ningún recurso externo de alimentación. Su programación dura unos 10 ms, y se controla mediante un temporizador interno.

Mencionar por ultimo que el microcontrolador dispone de dos Puertos de entradas y salidas E/S. El **Puerto A** con 5 líneas de la RA0 a la RA4, en la que hay que distinguir la A4 o **T0CKI** (*Timer 0 Clock Input*). Como puede ver en la breve descripción de la figura 1, esta línea es compartida con la entrada para el timer 0 (TMR0) a través de un trigger de Schmitt y que cuando se configura como salida es de colector abierto.

El **Puerto B** dispone de 8 líneas de entradas / salidas, que van de la RB0 a la RB7. En este, cabe mencionar que la línea RB0 ó INT es compartida con la entrada de interrupción externa (Ver figura 1).

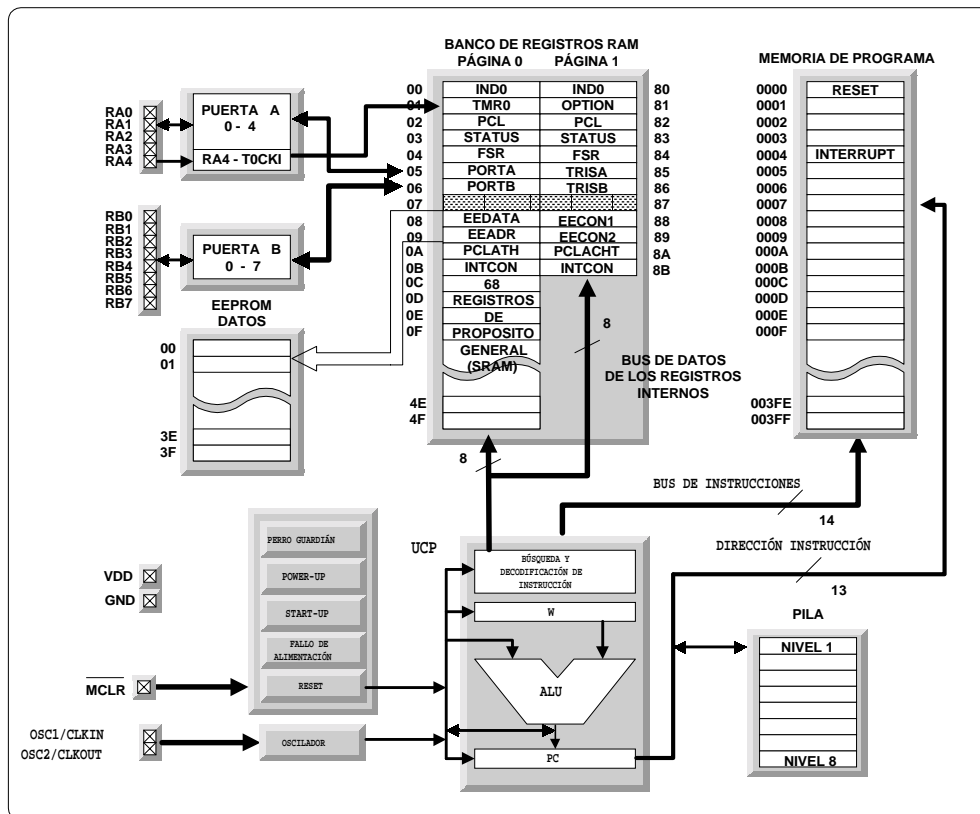


Figura 3.- Diagrama esquemático de la arquitectura interna del PIC16F84.

EL PROGRAMADOR MULTIPIC.

En la foto 2, puede ver el nuevo programador *MultiPIC* que permite grabar la mayoría de los microcontroladores PIC abarcando desde los de la gama baja de 8 pines como el 12C508 que se encuentra en el sistema anticopia de la PlayStation, hasta los de 40 pines como el 16C74 de la gama media.

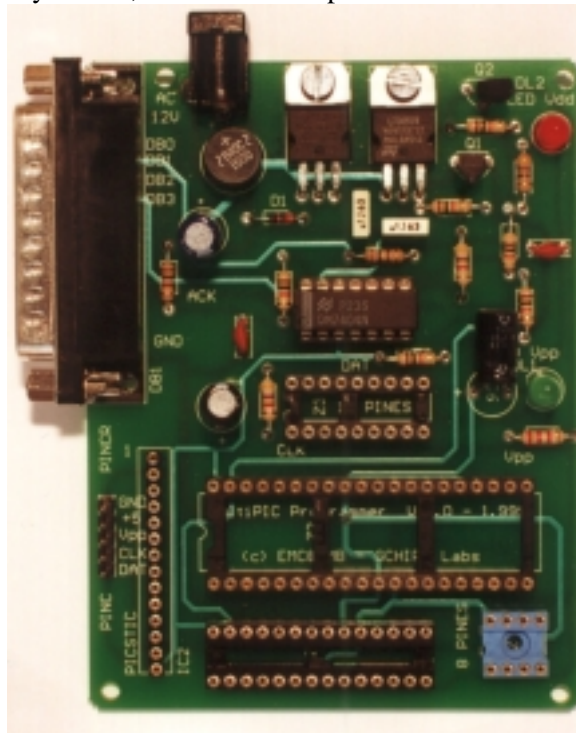


Foto 2.- Se presenta el programador *MultiPIC* diseñado para la grabación de múltiples modelos de microcontroladores PIC.

El grabador se conecta al puerto paralelo de cualquier PC, no importa el tipo; puede ser un antiguo XT o un moderno PetiumII. En el frontal, además del interruptor de encendido/apagado y de los zócalos para la programación se dispone de dos diodos LEDs, para evidenciar las señales de **POTENCIA** (V_{dd}) y **ACTIVIDAD** (V_{pp}). El LED *amarillo* indica que el programador se encuentra bajo tensión de alimentación V_{dd} mientras que el LED *verde* de **ACTIVIDAD** indica que está siendo aplicada al PIC la tensión de programación V_{pp} (12.6 V). Estas tensiones se encuentran bajo control del programa.

Recomendaciones también en un recuadro

¡¡ Atención !! solo debe insertar o retirar el PIC del grabador, cuando el LED de actividad está apagado. No inserte nunca el PIC con el LED de ACTIVIDAD encendido.

También es conveniente si se han de realizar repetidas grabaciones durante el aprendizaje, colocar el PIC en un zócalo de patitas torneadas, que soportará el esfuerzo de las repetidas inserciones y retiradas del zócalo de programación, evitando daños en las patitas del propio microcontrolador.

EXPERIENCIAS

Aunque pueda parecer una incoherencia, el primer programa que se realizará, estará escrito en ensamblador, dado que debe realizar menor cantidad de pasos y esto le ayudara posteriormente con los lenguajes de alto nivel. Necesitará el Módulo - 01 o un montaje similar. Realice el montaje que se presenta en la figura 4 conectando las líneas RB0 - RB7 (conector CN1) del Módulo - 01, con una serie de resistencias de 330 ohmios además de un diodo LED rojo.

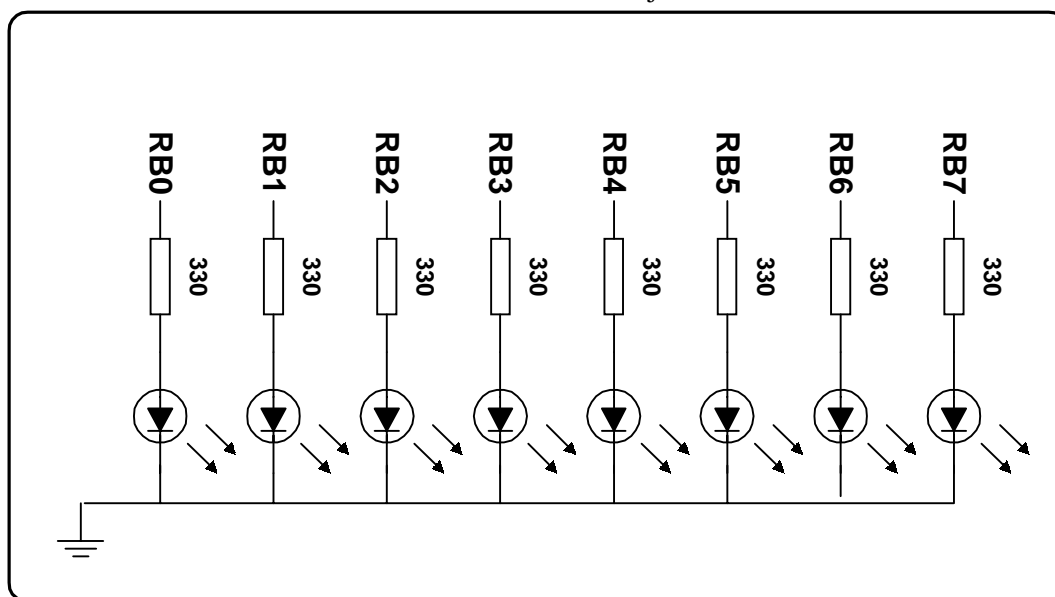


Figura 4.- Montaje necesario para la realización de la experiencia.

La experiencia va a consistir en la escritura y posterior grabación del microcontrolador PIC16F84 con el programa **CONTAR.ASM**, cuyo listado se presenta en el Apéndice - A. La finalidad del mismo consiste en contar en binario a través del Puerto B (patitas RB0 - RB7), iluminando los correspondientes diodos LEDs. Los pasos a seguir están esquematizados en la figura 7.

I.- El primer paso consiste en la escritura del programa **CONTAR.ASM** que se encuentra al final del artículo, haciendo uso de un editor de textos ASCII o del incorporado en la versión Windows del grabador MultiPIC. Una vez escrito el programa debe salvarse al disco.

II.- El segundo paso consiste en el ensamblado de este fichero (CONTAR.ASM) empleando el compilador-ensamblador **MPASM.EXE** que distribuye libremente el fabricante. El ensamblador produce varios ficheros; de entre ellos, el que necesitamos es el fichero **CONTAR.HEX**.

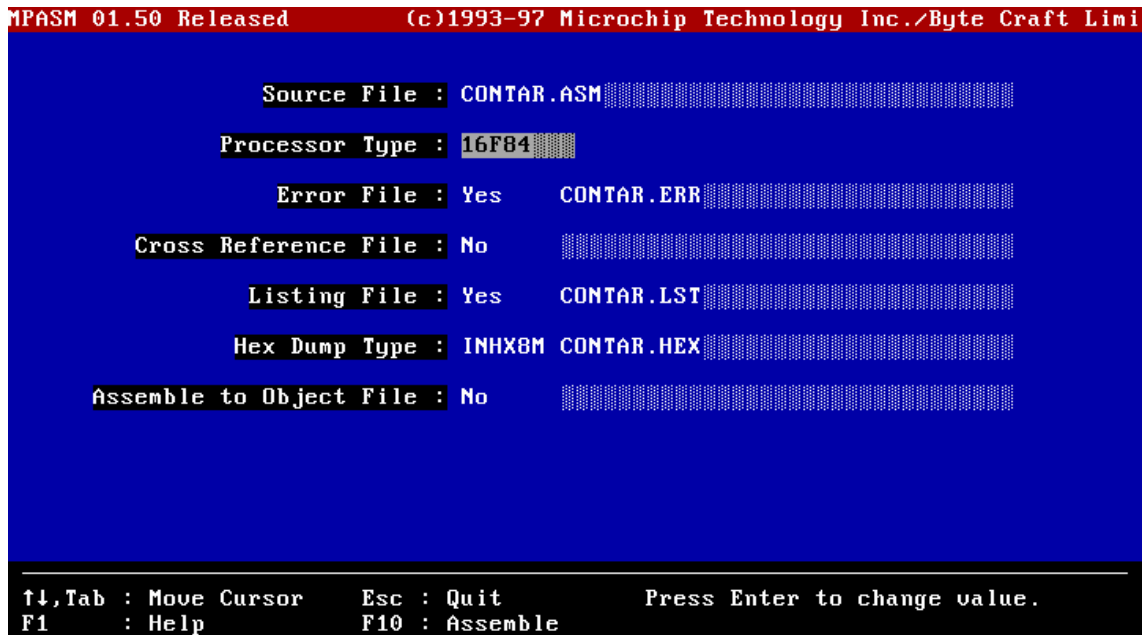


Figura 5.- Aspecto que presenta la pantalla del ordenador cuando se emplea el programa ensamblador MPASM. Como se observa en el ejemplo, en la línea **Source File** se indica el nombre del fichero a ensamblar **CONTAR.ASM**, en la **Processor Type** **16F84** y en la línea **Hex Dump Type** el fichero que se creará **CONTAR.HEX** en formato **INHX8M**.

III.- Llegado este punto es el momento de la grabación del microcontrolador. Conecte el programador a su computador, ejecute el programa **MultiPIC** y a continuación grabe en el microcontrolador el programa **CONTAR.HEX**.

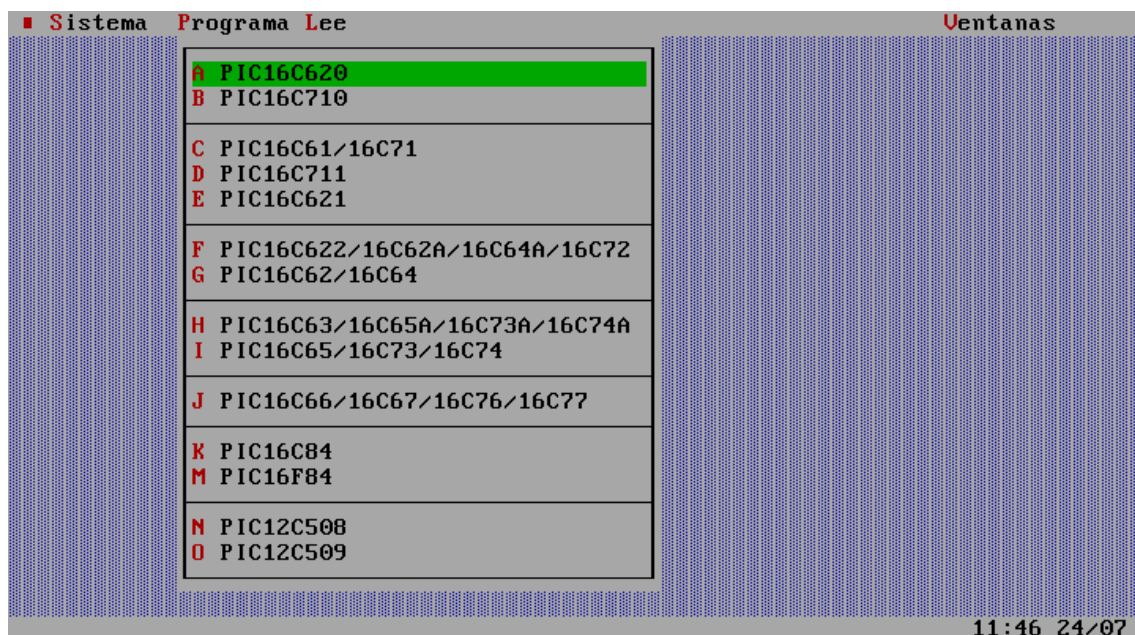


Figura 6.- Aspecto de la pantalla del ordenador durante el uso del programa de grabación **MULTIPIC.EXE**.

Retire el microcontrolador del grabador e insértelo correctamente en el Módulo-01. A continuación conecte el Modulo - 01 con el Módulo de aprendizaje tal como se muestra en las fotografía 3. En caso de no

disponer del mismo, realice ahora las conexiones que se le han indicado en la figura 4. Conecte la alimentación y ¡ Ahá ! su primer programa comienza a funcionar.

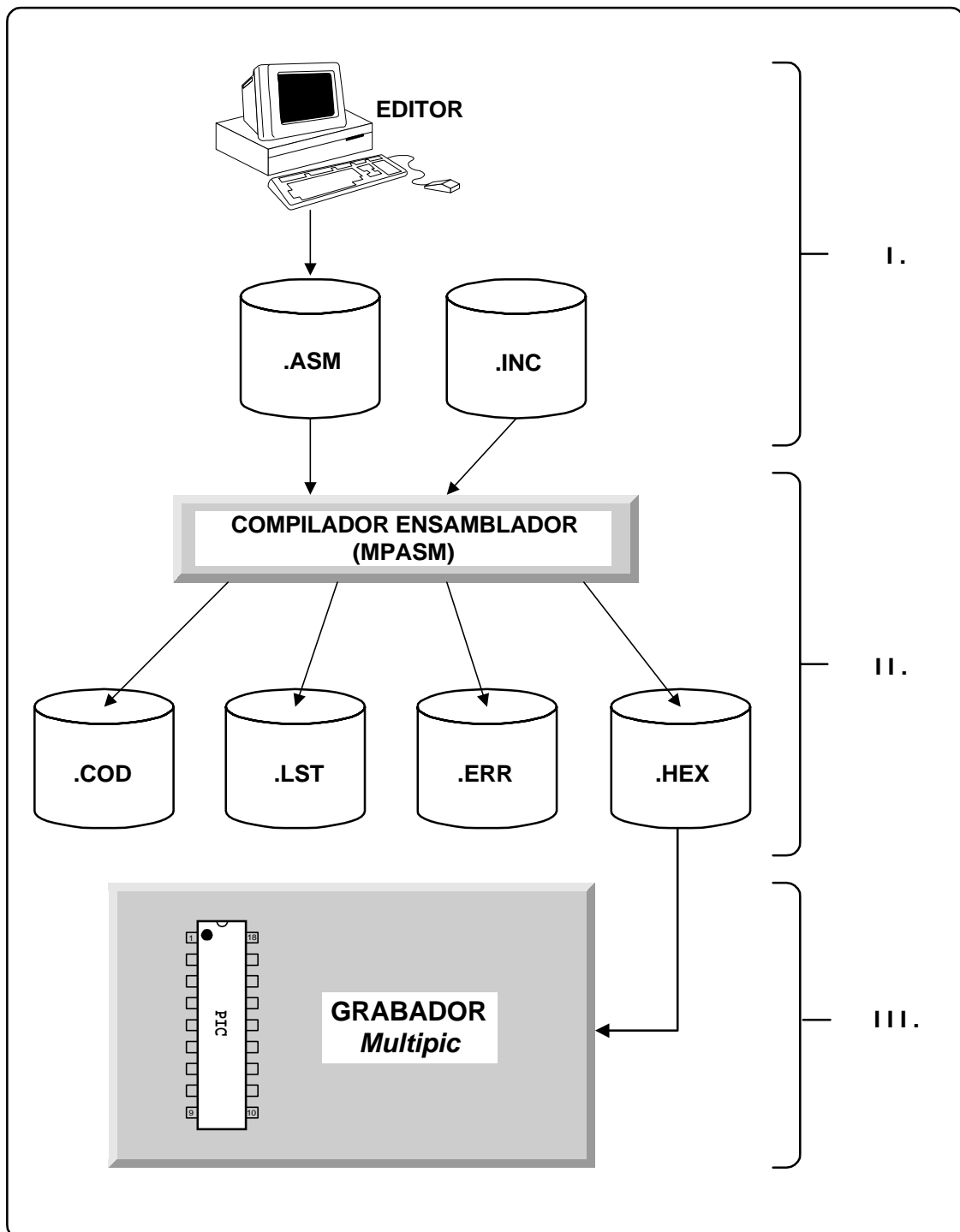


Figura 7. Esquema de la secuencia de pasos para la grabación del microcontrolador.

BIBLIOGRAFÍA

- ♦ Martín Cuenca, E., Angulo J.M., y Angulo, I. (1998). “*Microcontroladores PIC. La solución en un CHIP*”. 2ª Edición. Paraninfo-ITP.
- ♦ Martín Cuenca, E. y Moreno Balboa, J.M. “*Diseño y Realización de Aplicaciones Industriales con Microcontroladores PIC*”. (en preparación).
- ♦ Martín Cuenca, E. y Moreno Balboa, J.M. (1998). “*Fundamentos de Electrónica Moderna. Teoría y Diseño de Circuitos.*”

APÉNDICE - A : PROGRAMA CONTAR.ASM

```
.*****
;
; Programa CONTAR.ASM                      Fecha : *
; Este programa cuenta de 0 a 255 sacando el dato binario a los LED *
; conectados a la Puerta B introduciendo un retardo durante la cuenta *
; Revisión : 0.0                          Programa para PIC16C84/16F84 *
; Velocidad del reloj: 4 MHz                Reloj Instrucción: 1 MHz = 1 uS *
; Perro guardián : Deshabilitado           Tipo de Reloj : XT *
; Protección del código : OFF *
.*****
;
; ***** IGUALDADES *****
; ***** Igualdades que designa los destinos *****
w      EQU      0
f      EQU      1

.***** Igualdades de la UCP y del Mapa de memoria *****
;
PORTA  EQU      05h  ; Puerta A
PORTB  EQU      06h  ; Puerta B
TRISA  EQU      05h  ; Registro triestado Puerta A. Página 1
TRISB  EQU      06h  ; Registro triestado Puerta B. Página 1
STATUS EQU      03h  ; Registro STATUS
RP0    EQU      05h  ; Bit 5 registro STATUS
Contador1 EQU 0Ch  ; Registro para primer contador
Contador2 EQU 0Dh  ; Registro para segundo contador

; ***** Sección Código de reset *****
;
      ORG      00h  ; Dirección del Vector de reset
      GOTO    Inicializa ; Comienza el programa detrás
                        ; del Vector interrupción
      ORG      05h  ; Una posición a continuación
                        ; del vector de interrupción

; ***** Sección Inicializa *****
;
Inicializa  BSF      STATUS,RP0 ; Selecciona página 1
            CLRF     TRISB      ; Programa Puerta B todo salida
            BCF      STATUS,RP0 ; Vuelve a página 0
            CLRF     PORTB      ; Apaga los LEDs borrando la Puerta B
            CLRF     Contador1  ; Inicializa (borra) Contador1 (0Ch)
            CLRF     Contador2  ; y Contador2 (0Dh)

; ***** Sección Principal *****
;
Principal  INCF     PORTB,f     ; Incrementa en 1 la Puerta B
; ***** Sección Bucle *****
;
Bucle      DECFSZ   Contador1,f ; Decrementa Contador1 y chequea si = 0
```

Programación en BASIC de los Microcontroladores PIC

```
GOTO      Bucle      ; Si no 0, decrementa Contador1
DECFSZ   Contador2,f ; Si es 0, decrementa Contador2
GOTO      Bucle      ; Si Contador2 no es 0, decrementa
                        ; Contador1 otras 256 veces
GOTO      Principal   ; Actualiza los LED
                        ; después de 65536 iteraciones
END
```

