

## MANEJO DEL ENTORNO DE TRABAJO: PROTEUS ISIS Y MPLAB

Se propone la realización de un sencillo programa en ensamblador para el microcontrolador PIC16F84A que es el objetivo de estudio de este curso. El *datasheet* lo tienen en el CDROM.

### Objetivo

Se busca aprender a manejar las herramientas software y de desarrollo Proteus ISIS y MPLAB, así como poner en práctica el ciclo de diseño completo, que va desde la especificación del problema hasta la puesta en marcha y optimización del prototipo.

Existe una versión educacional de Proteus: Proteus Lite disponible en <http://www.proteuslite.com>. Se trata de un programa de diseño electrónico asistido por ordenador con captura de esquemas y simulación PSPICE-analógica, digital e híbrida, que permite la simulación de algunos microcontroladores.

El entorno integrado MPLAB es distribuido gratuitamente por el fabricante *Microchip* (<http://www.microchip.com>), e incluye un EDITOR, un ensamblador universal (MPASM), un enlazador (MPLINK) y un simulador (MPSIM). El *manual de usuario* de MPLAB lo tenéis en el CDROM.

### Especificaciones

El primer programa propuesto consiste en introducir un valor en binario por el PORTA del microcontrolador (5 bits) y sacar por el PORTB (8 bits) el valor leído incrementado en dos unidades. Se aconseja conectar los pines de entrada a los conmutadores del entrenador y las patillas de salida a los LED o al display de 7 segmentos del entrenador.

En esta ocasión proporcionamos el listado:

```
LIST p=16F84A
INCLUDE "P16F84A.INC"
RADIX DEC
ERRORLEVEL -302

ORG 0

bsf     STATUS,RP0      ; Selecciona Banco 1
movlw  11111111b       ; W = 0FFh (Todo entradas)
movwf  TRISA           ; Configuro PORTA
movlw  00000000b       ; W = 00h (Todo salidas)
movwf  TRISB           ; Configuro PORTB
bcf    STATUS,RP0      ; Selecciono Banco 0
Bucle  movfw  PORTA     ; Leo W = PORTA
      addlw  2          ; W = W + 2
      movwf  PORTB     ; PORTB = W
      goto  Bucle      ; Repito indefinidamente
END
```

## Para sacar nota

Prueba a cambiar el valor de salida en función de la entrada sacando el complemento del valor de entrada. ¿Sabrías codificar el valor mediante una operación O exclusiva?

Mnemonic, Operands	Description	Cycles	14-Bit Opcode			Status Affected	Notes		
			MSb		LSb				
<b>BYTE-ORIENTED FILE REGISTER OPERATIONS</b>									
<b>ADDWF</b>	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
<b>ANDWF</b>	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
<b>CLRF</b>	f	Clear f	1	00	0001	1fff	ffff	Z	2
<b>CLRW</b>	-	Clear W	1	00	0001	0xxx	xxxx	Z	
<b>COMF</b>	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
<b>DECf</b>	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
<b>DECFSZ</b>	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
<b>INCF</b>	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
<b>INCFSZ</b>	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
<b>IORWF</b>	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
<b>MOVF</b>	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
<b>MOVWF</b>	f	Move W to f	1	00	0000	1fff	ffff		
<b>NOP</b>	-	No Operation	1	00	0000	0xx0	0000		
<b>RLF</b>	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
<b>RRF</b>	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
<b>SUBWF</b>	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
<b>SWAPF</b>	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
<b>XORWF</b>	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
<b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>									
<b>BCF</b>	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
<b>BSF</b>	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
<b>BTFSC</b>	f, b	Bit Test f, Skip if Clear	1(2)	01	10bb	bfff	ffff		3
<b>BTFSS</b>	f, b	Bit Test f, Skip if Set	1(2)	01	11bb	bfff	ffff		3
<b>LITERAL AND CONTROL OPERATIONS</b>									
<b>ADDLW</b>	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
<b>ANDLW</b>	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
<b>CALL</b>	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
<b>CLRWDt</b>	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
<b>GOTO</b>	k	Go to address	2	10	1kkk	kkkk	kkkk		
<b>IORLW</b>	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
<b>MOVLW</b>	k	Move literal to W	1	11	00xx	kkkk	kkkk		
<b>RETFIE</b>	-	Return from interrupt	2	00	0000	0000	1001		
<b>RETLW</b>	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
<b>RETURN</b>	-	Return from Subroutine	2	00	0000	0000	1000		
<b>SLEEP</b>	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$	
<b>SUBLW</b>	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
<b>XORLW</b>	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

**Note 1:** When an I/O register is modified as a function of itself ( eg., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

- 2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.
- 3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

**Figura 1.** Conjunto de instrucciones del microcontrolador PIC 16F84A.