

MODIFICACIONES A LOS PROGRAMAS ENSAMBLADOR DE LA FAMILIA 16/ MIDRANGE PARA QUE FUNCIONEN EN LA FAMILIA 18/ HIGH PERFORMANCE.

En el amplio catálogo de microcontroladores que la empresa Microchip ofrece, existen entre otras, 2 familias: la llamada "Midrange" (por ejemplo, el 16F84, 16F628, 16F88, 16F877) y la "High performance" (por ejemplo, 18F2550, 18F4550).

Si usted está migrando de un microcontrolador de Microchip de la familia 16/ Midrange hacia la familia 18/ High Performance, puede usar sus antiguos programas escritos en lenguaje ensamblador **realizando solo modificaciones mínimas**.

Aquí tomaremos como ejemplo el caso de los microcontroladores 16F88 y 18F2550, pero estas modificaciones pueden hacerse extensivas a otros microcontroladores de las familias Midrange y High performance.

Las diferencias básicas entre ambos microcontroladores se muestran en la siguiente tabla:

	16F88	18F2550
Familia	Midrange	High Performance
Longitud de palabra	14 bits	16 bits
Memoria de código	4k palabras	16k palabras
Memoria de datos	368 bytes	2k bytes
Velocidad típica.	4 Mhz	48 Mhz
Set de instrucciones	35 instrucciones	75 instrucciones
Número de pines	18	28

Al modificar un programa del 16F88, es importante considerar que su ciclo de instrucción trabajando a 4 Mhz es de 1 us., mientras que en el 18F2550 funcionando a 48 Mhz es de solo 83.3 ns. Si el programa ensamblador contiene subrutinas de retrasos, éstas deben modificarse para compensar la diferencia de sus velocidades de operación.

Otra observación importante que el programador debe considerar en su archivo fuente (.asm), es programar **partiendo de una plantilla** para el microcontrolador y el sistema elegidos, en la cual ya estén incluidas la inicialización adecuada de los registros, tanto los SFR como los de configuración. En los siguientes links:

<http://www.puntofotante.net/BOLT-PLANTILLA-ENSAMBLADOR-2.asm>

<http://www.puntofotante.net/KEYBOLT.inc>

encontrará la plantilla utilizada por el sistema Bolt para el microcontrolador 18F2550 y su archivo auxiliar (.inc)

Y en estos otros links:

<http://www.puntoflotante.net/plantillaF88TINY2.asm>

<http://www.puntoflotante.net/key.inc>

encontrará la plantilla utilizada por el sistema EB88 para el microcontrolador 16F88 y su archivo auxiliar (.inc)

El set de 35 instrucciones del 16F88, es directamente compatible con el set del 18F2550 (el cual tiene 75 instrucciones), **con excepciones en solo 3 instrucciones, las cuales se detallan enseguida.**

Para modificar un programa escrito para el ensamblador del 16F88 y hacerlo compatible con el 18F2550, siga los siguientes pasos:

1. Localidades de memoria RAM para almacenar variables: en el 16F88 se tienen disponibles a partir de la localidad 20H. En el 18F2550, debe iniciarse el almacenamiento de variables a partir de la localidad 60H.
2. El registro especial FSR para direccionamiento indirecto, tiene 8 bits en el 16F88. En el 18F2550, tiene 16 bits y cambia su nombre a FSR0. Este registro se forma a su vez con los registros de 8 bits FSR0H y FSR0L, es decir la parte alta y la parte baja (ver ejemplos en el punto 3).
3. Ejemplos de manejo del registro FSR0. Reemplace las instrucciones en la izquierda de la tabla por su equivalente mostrado a la derecha.

16F88	18F2550
movlw 0x60 movwf FSR ;inicializa registro FSR	lfsr 0,0x60 ;inicializa registro FSR0
incf FSR,f ;incrementa registro FSR	incf FSR0L,f ;incrementa parte baja del registro FSR0

4. En las instrucciones de direccionamiento indirecto, debe modificarse el nombre del registro INDF (16F88), por INDF0 (18F2550), por ejemplo:

16F88	18F2550
incf INDF,w	incf INDF0,w

5. Las instrucciones de corrimiento ó rotación de bits que se usan en los microcontroladores Midrange tipo 16F88 (y que siempre se realizan incluyendo al CARRY), se reemplazan por las mostradas del lado derecho. Como observación

adicional, en el 18F2550, es posible también hacer rotaciones de bits en los registros **sin incluir al carry** con las instrucciones *rlncf*, *rrncf*.

16F88		18F2550	
rlf	REG1,f	rlcf	REG1,f
rrf	REG2,f	rrcf	REG2,f

6. Manejo de los bancos de registros: el microcontrolador 18F2550 incorpora un mecanismo automático para el direccionamiento de los registros especiales, de tal forma que **ya no es necesario modificar dentro del programa el valor del banco de registros**. Ésto simplifica considerablemente la programación en lenguaje ensamblador para el 18F2550. Observe en los ejemplos cómo el direccionamiento del registro TRISA requiere una inicialización del banco de registros en el 16F88, pero es inmediato en el caso del 18F2550.

16F88		18F2550	
bsf	STATUS,RP0	movlw	0x03
bcf	STATUS,RP1	movwf	TRISA
movlw	0x03	;inicializa registro TRISA, con	
movwf	TRISA	;direccionamiento automático del banco	
;inicializa registro TRISA		;de registros	

7. Manejo de tablas de datos con la instrucción *retlw*: el 18F2550 cuenta con una instrucción especial para el manejo de tablas de datos almacenadas en la memoria de código (instrucción *tblrd*). Sin embargo, también es compatible con la instrucción *retlw* utilizada para el mismo fin por los microcontroladores Midrange de la familia 16/.

Debe considerarse que en la familia 18/, el Program Counter –PC-, (a diferencia de la familia 16/), **direcciona la memoria de código por bytes** y no por palabras. Esto significa que, en cada instrucción, el PC avanza 2 unidades para realizar el ciclo fetch de la siguiente instrucción. Al trabajar con tablas de datos con la instrucción *retlw*, deberán entonces realizarse incrementos de PC+2 para direccionar correctamente a cada elemento consecutivo en la tabla (ver programa ejemplo).

Vea aquí un ejemplo de un programa realizado para el sistema Bolt 18F2550:

www.puntofotante.net/TABLAS-DE-DATOS-18F2550-LEDS.asm