

## COMENTARIO TÉCNICO

# *Buceando en el HC908.....*



Por Ing. Daniel Di Lella  
Dedicated Distributor Field Application Engineer  
For Freescale Semiconductors Products & Technical Consult  
Dto. Técnico Electrocomponentes S.A.  
[fae@electrocomponentes.com](mailto:fae@electrocomponentes.com)  
[dilella@arnet.com.ar](mailto:dilella@arnet.com.ar)

## *“Como sacarle mayor provecho a las herramientas disponibles para la familia HC908”*

### **4to Capítulo.**

Como les había prometido en el capítulo anterior, ahora le toca el turno a los entornos integrados de trabajo más conocidos como IDE (Integrated Development Environment) y su interacción con las herramientas de hardware disponibles para la familia de microcontroladores FLASH HC908.

En este artículo, nuestros actores principales serán el nuevo sistema de desarrollo **FLASH\_POD** recientemente presentado en nuestro mercado por Electrocomponentes S.A. y el entorno integrado de trabajo (IDE) **CodeWarrior 5.0** de Freescale Semiconductor.

El sistema de desarrollo **FLASH\_POD** es una herramienta universal pensada para trabajar con toda la familia HC908 al igual que su predecesor, el sistema **E-FLASH08**. Esta nueva herramienta incorpora importantes ventajas en lo que refiere a flexibilidad de conexión con distintas PC's o Notebooks ya que dispone de un puerto serial universal (USB 2.0) y un puerto serial RS-232C (COM) que puede usarse en forma indistinta en la comunicación. Otras mejoras a destacar es la flexibilidad en las distintas fuentes de alimentación del sistema, el agregado de funciones de diagnóstico y otros “chiches” que facilitarán el trabajo del desarrollador.

Por otro lado, la nueva versión del entorno integrado de trabajo (IDE), el **CodeWarrior 5.0** ha sido pensada para simplificar el manejo de este poderoso entorno por parte de usuarios no habituados a los ambientes profesionales de desarrollo, cosa que no ocurría con las antiguas versiones de **CodeWarrior** para HC08 y HCS08.

Pues bien, hecha la presentación de las herramientas, nos adentraremos con un ejemplo práctico que nos facilitará la tarea de comprensión.

Nuestro ejemplo consistirá en implementar un pequeño programa en lenguaje ensamblador que efectue una interrupción en forma periódica cada “n” milisegundos basado en el uso del Timer en modo TOV (Timer Overflow). Este sencillo programa nos servirá como base para ejecutar un número de tareas más complejas en forma periódica de modo similar a como lo haría un sistema operativo más complejo.

Primeramente configuraremos al sistema **FLASH\_POD** para trabajar con el MCU **MC68HC908QY4CPE** que es el microcontrolador disponible en la placa **DEMO (Demo Board)** que viene con el kit del sistema.

Utilizaremos la conexión **USB** entre el sistema de desarrollo y nuestra PC, alimentando al microcontrolador y a la placa Demo con +5Vdc provistos por la propia herramienta.

De esta forma la configuración final será:

- Jumpers JP1A / JP1B / JP1C = **Posición 2-3 cerrado.**
- Jumpers JP3 / JP4 = **Posición 1-2 cerrado.**
- Jumpers JP5 / JP6 = **Posición Cerrado.**
- Jumper JP7 = **Posición 1-2 Cerrado.**
- Número de Configuración en display = **18**
- Baud Rate de la comunicación = **9600 bps**
- Frecuencia del Bus =  $OSC / 4 = 2,4576 \text{ MHz}$

Cabe recordar que como utilizaremos el puerto USB de nuestro sistema para comunicarnos, se deberá comprobar que se hayan instalado los drivers necesarios, provistos con la herramienta, para que la PC o Notebook utilizada interprete como un puerto “**COM Virtual**” al puerto USB físico, ya que en los entornos de trabajo integrados como el **CodeWarrior** o **P & E WinIDE** utilizaremos la opción “COM” como medio válido para comunicarnos (ver sección **10.0 “Poniendo en Marcha el sistema”** del manual de usuario de la herramienta).

Configurado nuestro sistema **FLASH\_POD**, procederemos a iniciar nuestro programa en el sistema **CodeWarrior 5.0** efectuando los siguientes pasos:

- 1) Al ejecutar el **CodeWarrior IDE**, se nos abrirá una ventana de opciones como se vé en la **figura 1**.

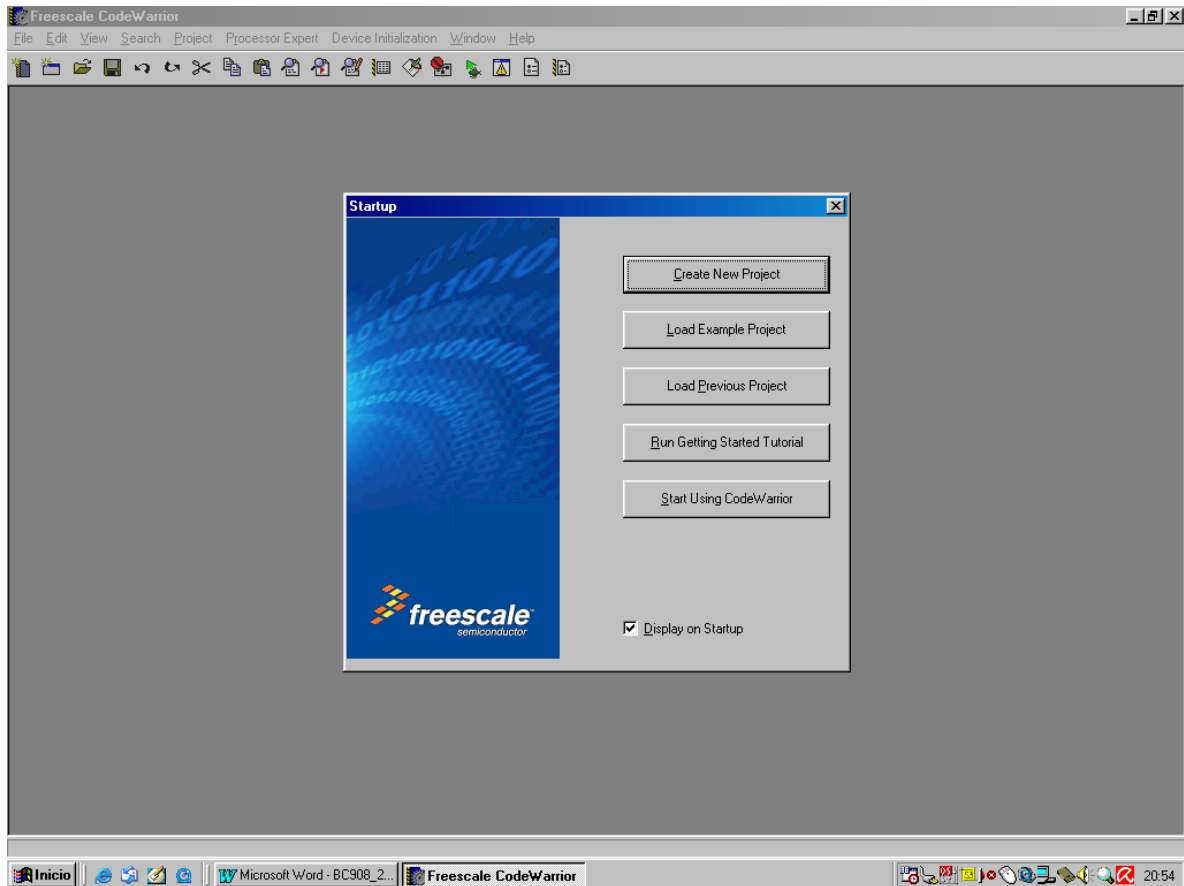


Figura 1.- Pantalla “Startup” con opciones de ayuda.

Eligiremos la opción “Create New Project” para armar nuestro nuevo proyecto .....

2) Se ingresará en la pantalla de configuración del proyecto donde elijeremos generación de lenguaje **ASSEMBLY**, y nombraremos a nuestro proyecto con el nombre “**proyecto1.mcp**”, según se puede ver en la **figura 2**.

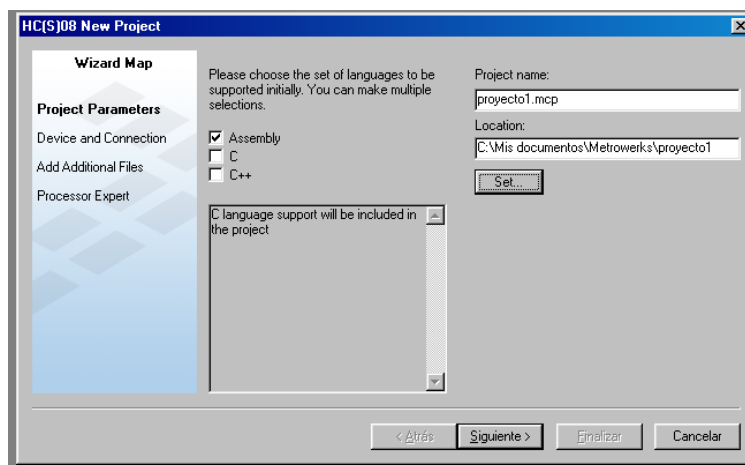
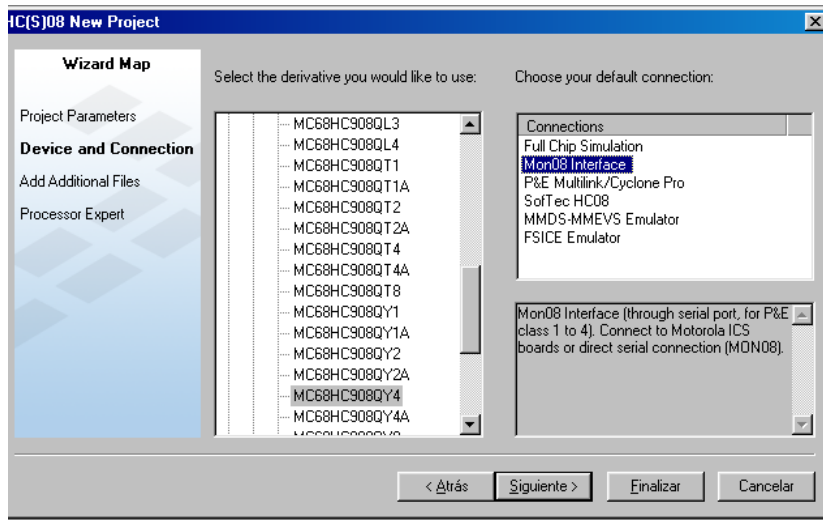


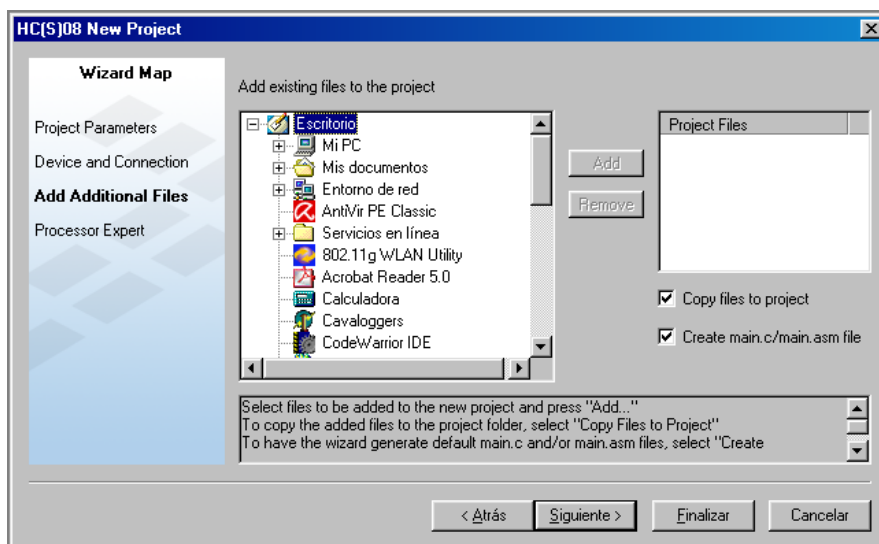
Figura 2.- Pantalla de configuración del proyecto.

- 3) En la siguiente pantalla configuraremos la familia y dispositivo en particular a utilizar para nuestro proyecto (**MC68HC908QY4CP**) según se puede ver en la **figura 3**.



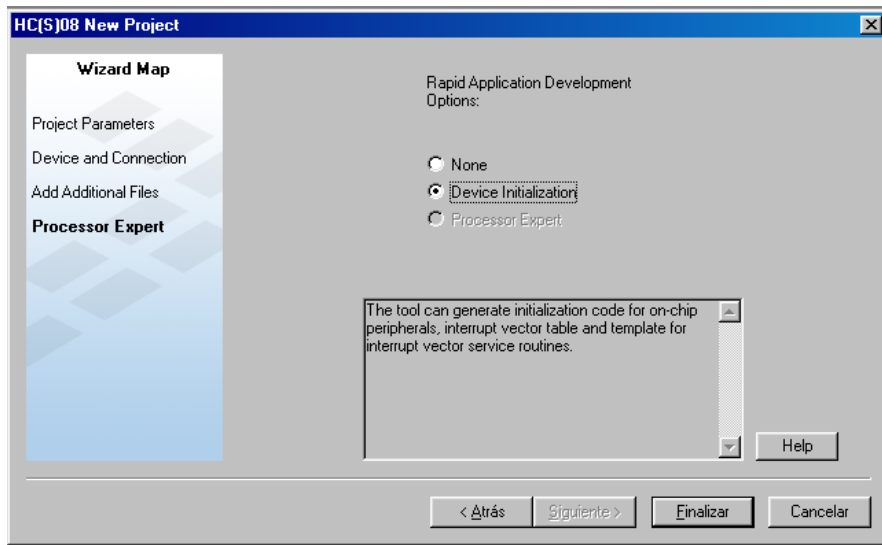
**Figura 3.- Pantalla de configuración de Familia, dispositivo y tipo de conexión.**

Como se observa en la **figura 3**, se ha elegido la familia HC08, dispositivo **MC68HC908QY4** y en cuanto a la conexión con la herramienta debe elegirse la opción **“Mon08 Interface”** ya que es la opción universal de conexión para los sistemas de desarrollo como los **EVAL08QTY**, **E-FLASH08**, **FLASH\_POD** y toda otra herramienta que no figure explícitamente en el listado de conexiones.



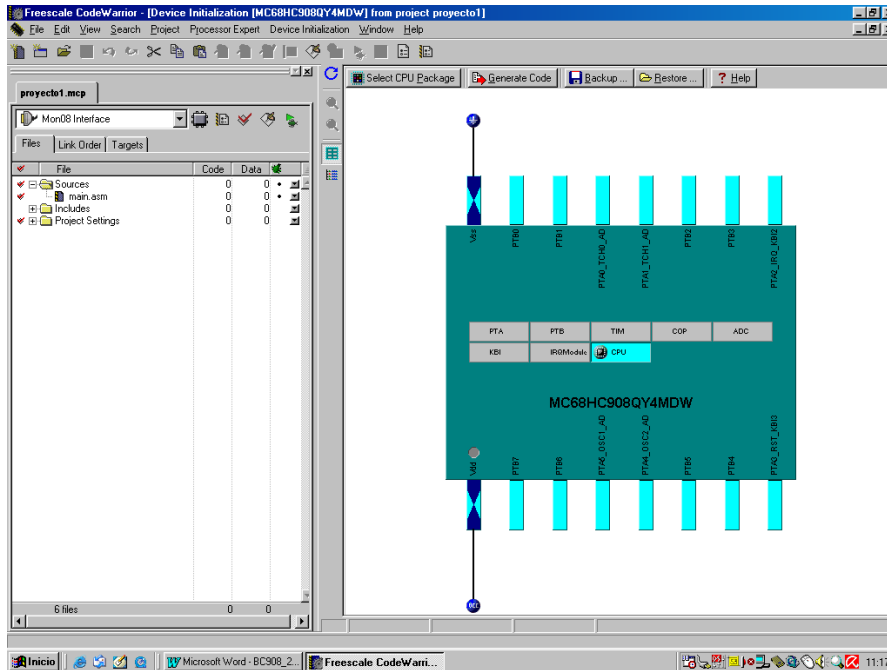
**Figura 4.- Pantalla de adición de archivos al proyecto.**

- 4) Al hacer click en el botón “**siguiente**” pasamos a una pantalla (**Figura 4.-**) que nos permite adicionar cualquier archivo al proyecto, para incluirlo en nuestro trabajo. En nuestro caso, saltaremos esta opción e iremos a la próxima pantalla.



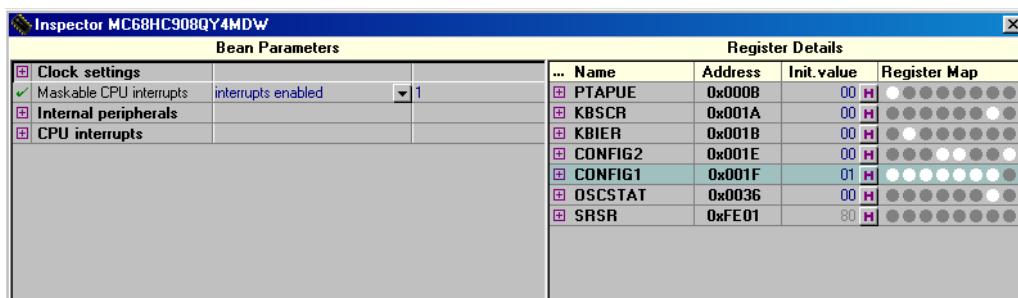
**Figura 5.- Pantalla de elección o no de generación de código asistido (Processor Expert).**

- 5) En la pantalla que se observa en la **figura 5.-**, se puede elegir la generación de código de inicialización de los distintos periféricos asistida o no. Nosotros elegiremos utilizar la generación de código asistida, por medio del aplicativo “**Processor Expert**”, que nos irá guiando paso a paso en la inicialización de los distintos periféricos del MCU elegido.
- 6) Al hacer click en el botón “**Finalizar**”, se generarán todos los archivos del proyecto, se lanzará la pantalla principal de trabajo del mismo y se podrá ver una interface gráfica con los pines y los distintos módulos que constituyen el MCU (**Figura 6.-**).



**Figura 6.-** Pantalla principal del proyecto e interface gráfica de generación de código (Processor Expert).

Ahora nos queda generar el código de inicialización del Timer para producir una interrupción periódica que será la base de nuestro sistema de disparo de tareas, inicializar los puertos I/O, los registros de configuración, etc., etc. Para hacer esto, usaremos el generador de código asistido “Processor Expert” haciendo click primeramente en el modulo CPU para configurar el Clock del sistema y otros aspectos como se observa en la **figura 7.-**



**Figura 7.-** Pantalla del módulo de CPU.

Se configura el **módulo de CPU** para:

- Clock ---- Externo ---- 9,8304Mhz (lo inyectará FLASH\_POD por pin OSC1).
- LVI ----- Habilitado ---- 3V trip point ----- LVI deshabilitado en modo STOP.
- Interrupciones Habilitadas.
- Vector de Reset apuntando a la etiqueta “\_Startup”
- Pin de Reset externo no disponible.

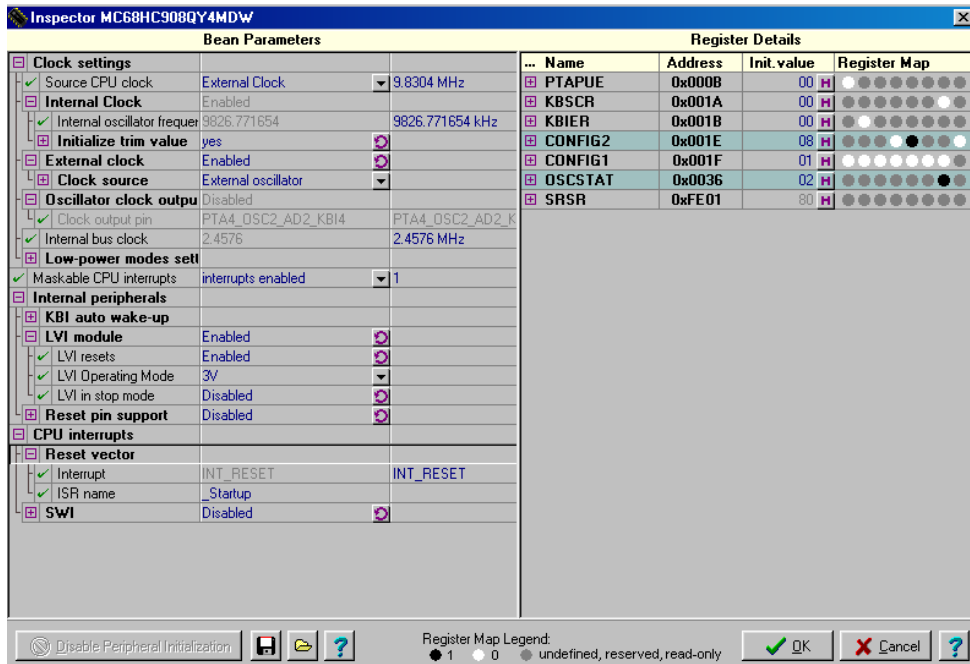


Figura 8.- Pantalla con los detalles de configuración del CPU.

- 7) A continuación procederemos a configurar el **módulo de Timer (TIM)** ingresando al mismo como muestra en la **figura 9.-**

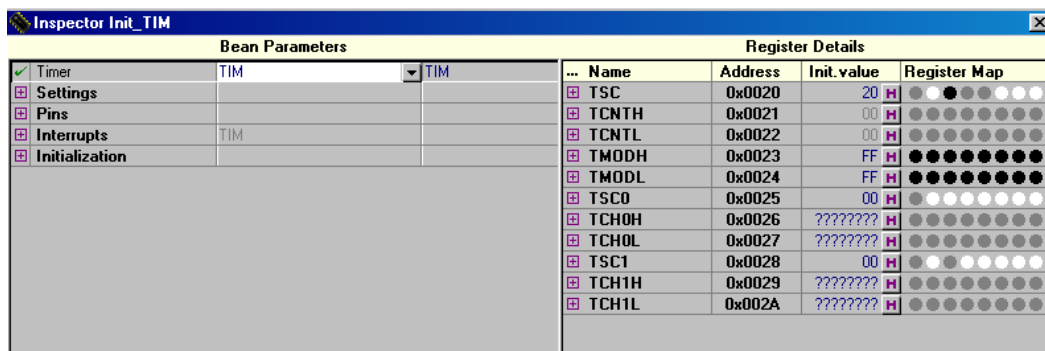


Figura 9.- Pantalla del Módulo de Timer.

Continuará.....