

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

dilella@arnet.com.ar

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

Gracias al avance de la tecnología, hoy en día nos resulta común trabajar con herramientas de hardware y software que nos permiten una mayor comodidad a la hora de depurar código y encontrar problemas más rápidamente.

Sin ir más lejos, la emulación en tiempo real, era prohibitiva hasta que aparecieron nuevas familias de microcontroladores como la HC908, que incorporan modos de depuración como el “modo monitor” que permiten contar con herramientas económicas y con prestaciones similares a las de mayor costo.

Sin embargo, para aprovechar estas ventajas, estas herramientas de menor costo necesitan que el usuario tome algunas precauciones para evitar problemas durante el trabajo de desarrollo.

Como resultado de las preguntas, que periódicamente he recibido, he confeccionado un resumen de las dudas más frecuentes y sus respuestas que espero les sean útiles en su trabajo diario.

Aquí van algunas....

**- Que diferencia existe entre "Simulación Pura", "Simulación En - Circuito" ,
y "Emulación en Tiempo Real" ??**

Estos tres modos de "debugging" o depuración, están disponibles en los sistemas EVAL08QTY y E-FLASH08 por medio de entornos integrados de desarrollo bajo Windows denominados "WinIDE" (de P & E Microcomputers System) o el CodeWarrior (de Freescale Semiconductors) que permite disponer de los mismos por medio de "iconos" o "botones de acción rápida", además de otras aplicaciones.

Por "**Simulación Pura**", se entiende al modo de "simulación" de funcionamiento del software o programa escrito para una aplicación en el MCU, estrictamente en el entorno del PC Host, SIN que medie circuito o hardware alguno. Por lo tanto, todo el proceso de depuración de código (ver como funciona "lógicamente" el programa), se efectúa bajo el entorno del sistema operativo del PC Host, por lo que el método **NO** es de **TIEMPO REAL** y la "velocidad" de corrida del programa dependerá fuertemente del PC Host utilizado.

En este modo de depuración, NO es necesario el uso del hardware de los sistemas EVAL08QTY o E-FLASH08, y suele usarse en los "primeros pasos" de la depuración de un programa de un MCU.

Por "**Simulación En - Circuito**", se entiende al modo de "simulación" de funcionamiento del software o programa escrito para una aplicación en el MCU, en el entorno del PC Host, pero con el uso de la placa de hardware (Pod) propia del sistema EVAL08QTY o E-FLASH08 elegido, con el PC. Esto permite vincular el programa a depurar con el mundo exterior en forma física (niveles lógicos en un puerto I/O, niveles de tensión en un canal A/D, etc.), pero bajo el control del sistema operativo del PC Host, por lo que el método **NO** es de **TIEMPO REAL** y la "velocidad" de corrida del programa dependerá fuertemente del PC Host utilizado.

Este modo de depuración suele usarse en los "primeros pasos" de la depuración de un programa de un MCU y cuando las señales eléctricas involucradas NO son críticas en el tiempo (eventos muy lentos).

Por "**Emulación en Tiempo Real**", al modo de funcionamiento que permite la emulación de las funciones y funcionalidad de un microcontrolador determinado **a la velocidad del clock del sistema**, esto significa que en este modo de funcionamiento, **la depuración del programa de un MCU se hace "corriendo" a la velocidad real del chip** SIN intervención del sistema operativo del PC Host, ya que este último solo se limita a recibir o enviar comandos de acción al sistema emulador (solo como un "administrador" de las novedades).

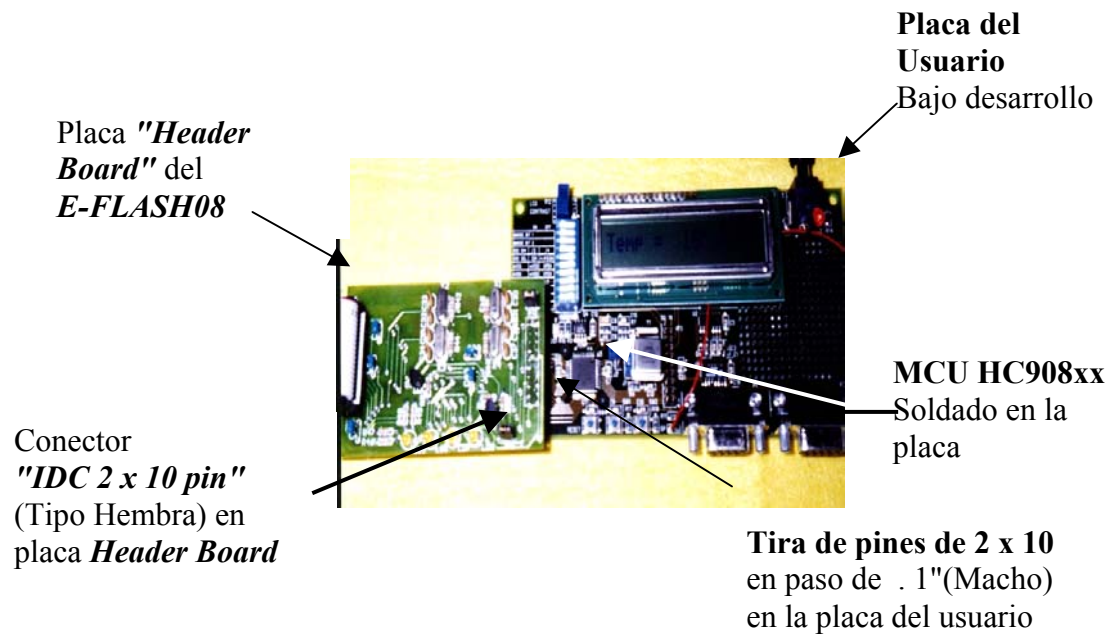
Este modo de depuración es el más preciso y real de todos los anteriores y suele usarse en las instancias finales del proceso de depuración de un programa o cuando se tiene gran seguridad en el uso de estas herramientas. Este modo es el menos conocido ya que estuvo por mucho tiempo solo disponible para herramientas de alto costo.

- Que características presenta el "conector de vinculación" entre el sistema ¿E-FLASH08 y la placa del usuario a emular?

El conector de vinculación entre Emulador - Placa bajo desarrollo, es del tipo 2 x 10 pines (**tira de pines 2 x 10 a paso de .1"**) "**header**" en la placa, y del tipo **IDC 2 x 10 (hembra) en el emulador (Header Board)**.

Este conector (el de la placa del usuario bajo desarrollo) debe intercalarse entre el MCU y el resto del circuito, de esta forma el emulador "**abre**" el camino natural de las señales **entre el MCU y el resto de la circuitería**.

Cuando el emulador, ya no es necesario, este conector de vinculación de la placa de usuario (tira de pines) es puesto en "**cortocircuito**" entre pines opuestos por medio de jumpers estándar.



En la **figura 1**, pueden verse algunos detalles de conexión entre MCU y resto de circuito, para un derivativo de la línea **HC908**.

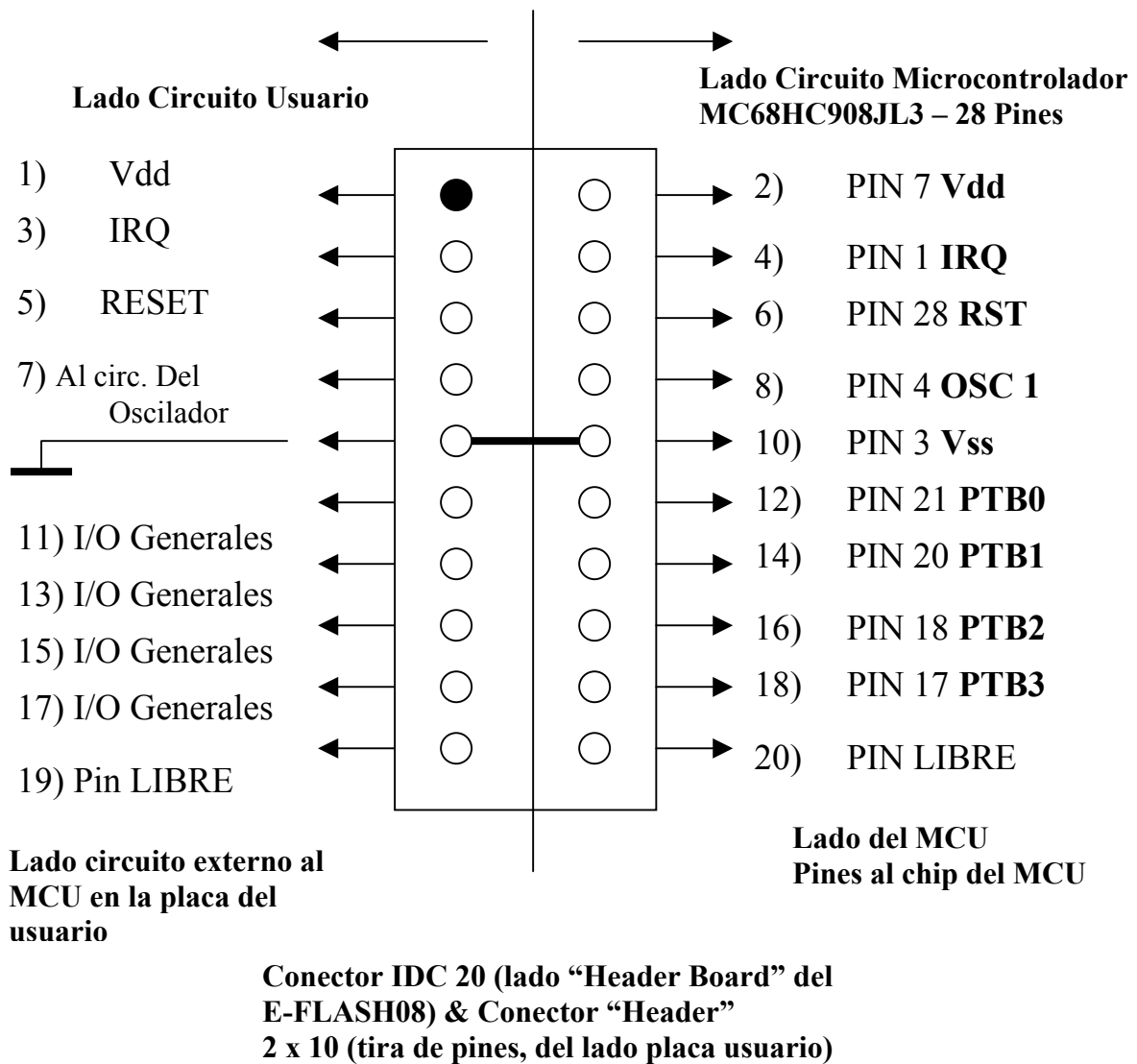


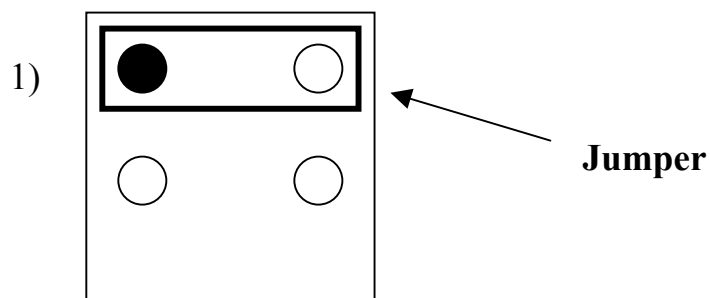
Figura 1. – Detalle de conexión del conector IDC 20 y MCU – Resto del circuito.

NOTA 1: En el Pin 7 del conector IDC 20 se debe conectar el circuito del oscilador, es decir una de las patas de cristal con su correspondiente capacitor de 22 PF y la resistencia de 10 Mohms. La otra pata del cristal debe ir directamente conectada al **PIN 5 del Chip (OSC 2)**.

NOTA 2: En los pines donde se conectarán los circuitos correspondientes a IRQ y RST (reset), así como en los pines de I/O generales (uso I/O general), el usuario conectará el dispositivo que considere necesario (demás C.Is, transistores, etc.) de acuerdo con su diseño original.

NOTA 3: El concepto del conector tipo “tira de 2 x 10 pines en la placa del usuario” (Header), es como si el mismo fuera “intercalado” entre las 10 señales más importantes entre el MCU y el resto del circuito externo al mismo. Estas señales son utilizadas por el sistema E-FLASH08 para “dialogar” con el MCU bajo desarrollo en el modo “monitor” y luego de liberar las mismas en forma automática, permitir de esta forma una emulación en tiempo real.

Una vez finalizada la etapa de depuración de la placa del usuario, el conector formado por una tira de pines de 2 x 10 en la placa bajo desarrollo (Conector “Header”), debe ser PUENTEADO por “jumpers” que permitan al flujo normal de las señales en cuestión. (Ejemplo, se puentea PIN 1 con PIN 2, PIN 3 con PIN 4..... PIN 19 con PIN 20.)



¿Que señales o pines no están disponibles en el emulador E-FLASH08?

El E-FLASH08 es un Emulador En - Sistema, que por su concepción, permite disponer libremente de los pines de los distintos MCUs a emular. Un circuito exclusivo **permite que todos los pines del MCU estén disponibles** durante el Modo de Emulación **sin limitaciones de uso** (I/O, ADC, Timers Channels, IRQ, Reset, etc.). Solamente **1 (un) pin** del MCU **NO está disponible** durante la emulación, porque es necesario un puerto de comunicación a un solo hilo entre el MCU y la PC (1-WIRE-M). El pin 1-WIRE -M, Generalmente, es **PTA0 ó PTB0** según el dispositivo elegido (ver manual de datos del MCU respectivo) y presenta las mismas limitaciones de uso que en los sistemas EVAL08QTY.

Si se desea usar este pin **SOLO** se lo puede hacer en el **modo Usuario disponible por Medio del "setup" del sistema** (ver **Modo Usuario en el manual del sistema**).

- Cuales son las señales / pines del sistemas E-FLASH08, con las que se debe tener algún cuidado durante su uso ?

Por lo explicado en el punto anterior, el sistema E-FLASH08, permite operar con todas las señales / pines del MCU a emular, sin mayores cuidados en cuanto a su uso. Sin embargo, cabe aclarar que, la señal de **IRQ** (External IRQ), NO es físicamente la misma señal o pin del MCU, ya que la misma está separada ("Buffered") del resto del circuito exterior. Esto se debe a que en el pin propio del MCU existe una tensión elevada (8 Vdc aprox.) para colocar al mismo en modo "monitor" y garantizar así la emulación. Por lo tanto, el pin IRQ - C (lado Circuito externo), se encuentra separado por un circuito "buffer" especial de aislación, del pin IRQ-M (lado MCU), aunque funcionalmente son idénticos.

Otra señal a tener en cuenta es **RESET (RST)**, pues el sistema E-FLASH08, permite configurar a la misma para tensiones de trabajo igual a **VDD (+ 3Vdc a + 5Vdc)** o igual a **Vhigh (+8 Vdc)**. Esta característica del sistema, le da al mismo, la posibilidad de trabajar libremente con el pin de IRQ, sin necesidad de deshabilitar el sistema de COP watchdog por software, durante la emulación de un programa determinado. Para mayor información acerca de cómo funciona el pin de RESET con Vhigh durante el modo monitor, consultar con el manual de datos respectivo del chip, en la sección "**Monitor ROM**".

- **Los sistemas EVAL08QTY / E-FLASH08 utilizan alguna frecuencia de Clock especial durante la emulación o simulación en circuito / sistema?**

SI, debido a que estos sistemas basan su funcionamiento en el "Modo Monitor" del MCU HC908, necesitan de una frecuencia de clock especial para comunicarse con el mundo exterior (PC Host) durante la emulación. Dicha frecuencia estará en relación con una velocidad estándar de comunicación o Baud Rate (4800, 9600, 14.400 Bps, etc...) que pueda generar el sistema, a partir de una frecuencia de cristal dada.

El valor de la frecuencia del cristal, dependerá del sistema **EVAL08QTY** o **E-FLASH08** utilizado. Ver las frecuencias disponibles para cada uno de ellos, en sus hojas de datos.

En el sistema E-FLASH08, el clock es generado por la placa "**Header Board**", que posee cuatro (4) frecuencias distintas, e inyectado en la placa del usuario bajo desarrollo, aunque la placa tenga su propio cristal. Esto es posible, gracias a la disposición de pines del conector "header" 2 x 10 de la placa del usuario que permite "abrir", en forma natural, el camino de la señal del cristal utilizado en la misma placa, anulando su funcionamiento durante la Emulación y facilitando la inyección de señal del oscilador externo de Clock.

Continuará