

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”

3era. Parte....

Hola amigos, continuando con nuestra serie de artículos referidos a como sacarle el mayor provecho a las herramientas disponibles para la familia HC908, tocaremos no solo aspectos de hardware a tener en cuenta, sino también, aspectos de los entornos de desarrollo disponibles para la familia HC908 FLASH de Freescale Semiconductor.

- **Uso de pocos pines para efectuar grabaciones en fábrica:**

Una duda que se nos plantea frecuentemente y sobre todo cuando trabajamos con sistemas que disponen de poco espacio para ubicar un conector de programación es..... ¿Cuántos pines necesita mi sistema para programarlo en circuito?

Entonces aquí nos surgen la dudas; ¿Necesito 2 x 10 pines como cuando utilizo los emuladores E-FLASH08 o FLASH_POD?, ¿Necesito menos? ¿Cuántos?

Bien, antes de responder a estas dudas, debemos recordar que en la familia HC908 para realizar las operaciones de borrado / grabación de la memoria flash de programa y también la depuración de código, hay que colocar al microcontrolador en un estado muy particular que se denomina “Modo Monitor”. En dicho estado, el microcontrolador ejecutará rutinas internas en ROM y otras alojadas temporalmente en RAM que permitirán realizar las funciones anteriormente comentadas bajo el control de una aplicación corriendo en una PC.

Para entrar en el “Modo Monitor” se deben cumplir una serie de condiciones en el entorno del microcontrolador que serán más o menos exigentes según el estado previo de la memoria flash de este. Como habíamos visto en artículos anteriores, debido a una posición conservadora de la gente de ingeniería de Freescale (en ese momento Motorola Semiconductor) se optó por utilizar unos cuantos pines de los microcontroladores de esta familia, más el agregado de una tensión elevada en el pin IRQ, cuando la memoria flash de estos se encuentra grabada (por un programa previo del usuario). Este modo se denomina “Modo Monitor Alta Tensión” y el mismo prescinde del estado de la memoria Flash.

Afortunadamente, existe un segundo modo denominado “Modo Monitor Forzado”, que solo está disponible cuando la memoria del microcontrolador está “Virgen” o Vacía (borrada), y que requiere muy pocos pines del MCU.

En el Modo Monitor Forzado, **solo 4 pines** son necesarios en nuestra placa para lograr la programación del sistema “en – circuito”, y ellos son:

- **+VDD**
- **GND (Vss del MCU)**
- **Reset**
- **PTA0 o PTB0 (según el MCU).**

Con el control de estas señales (ver en detalles la sección “Monitor ROM” del HC908 elegido) el usuario no necesita de las condiciones exigentes como en el primer modo para alcanzar igual objetivo. Sin embargo, todo esto es válido si y solo si la memoria flash está vacía o Virgen..... Entonces, ¿Que hacer ante una actualización del programa? De hecho, la memoria de nuestro MCU ya no es más “Virgen” y no está vacía por lo cuál no es válido lo anterior.

Por suerte, podemos echar mano a un “truco” que hará que nuestra memoria flash vuelva a ser “Virgen” asegurando de esta manera, la condición necesaria para ingresar nuevamente al modo monitor forzado. Para ello, utilizaremos las rutinas de borrado en masa (Mass Erase) de la memoria flash que disponen casi todos los HC908 en su memoria ROM interna y que solo invocaremos cuando alguna condición externa nos indique que estamos en presencia de comenzar con la operación “borrado total”.

La condición externa puede ser, por ejemplo, la conjunción de un pin de nuestro MCU forzado a “1” o “0” lógico, más tomar la salida del estado de Reset del mismo como momento para preguntar por el estado de ese pin, si ambas condiciones coinciden, entonces procederemos a ejecutar una rutina de borrado general (Mass Erase) y verificación del borrado de la memoria (este pequeño programa será parte del programa del usuario que reside en flash, pero deberá copiarse y luego ejecutarse desde la RAM para asegurar que el mismo se siga corriendo aun cuando se haya efectuado el borrado general de la memoria flash).

Con nuestro MCU en esas condiciones, al retirar la energía y volver a energizarlo, estará listo para ingresar en el Modo Monitor Forzado como si el mismo nunca se hubiera programado!!!.

Cualquiera de la herramientas para HC908 nos servirán para programar nuestro sistema a pocos hilos (E-FLASH08, FLASH_POD o un simple circuito de grabación) sin la necesidad de utilizar el conector completo en nuestra placa.

- **Debugging de código a pocos hilos:**

Entusiasmados por haber conseguido programar / reprogramar la memoria flash de nuestro HC908 nos preguntamos..... ¿Es posible Emular en Tiempo Real a pocos hilos?

La respuesta no es un “Si” universal, ya que es posible, pero dependerá de las condiciones y limitaciones que imponga nuestro sistema. Por esta razón se deberá tener especial cuidado en el diseño de la aplicación para que todas las condiciones estén presentes durante el debugging y no existan interferencias con las señales del sistema (sugiero leer en detalle la sección “Monitor ROM” de cada MCU en particular).

A modo de ejemplo, daré algunos puntos a tener en cuenta:

- **Puertos I/O utilizados en el Modo Monitor Alta Tensión.**

El Modo Monitor Alta Tensión de los HC908 (único modo útil para efectuar el debugging) necesita como condición complementaria forzar algunos puertos I/O a ciertos niveles lógicos (la cantidad, cuales y a que estado, dependen de cada dispositivo en particular) durante el estado de Reset del microcontrolador y durante algún tiempo posterior. Teniendo en cuenta ello, nuestro diseño deberá garantizar que dichos puertos tengan los niveles lógicos necesarios para garantizar el ingreso al modo monitor, por lo que deberemos asegurarnos que las señales de nuestra aplicación no interfieran con ello o mejor aun, contribuyan naturalmente a lograrlo. Como ejemplo podemos citar el uso de pulsadores con niveles lógicos favorables en esos pines, señales de circuitos externos que durante el “arranque” del sistema fuercen esos estados, etc. La idea aquí es poder utilizar esos puertos I/O para nuestra aplicación sin tener que dejarlos “exclusivos” para asegurar las condiciones a modo monitor.

- **Pin IRQ.**

La condición principal para ingresar en el Modo Monitor Alta Tensión es asegurar que en pin IRQ de los HC908 haya una tensión elevada (+7,5V a +9V) durante el estado de Reset del microcontrolador y según algunas condiciones, durante el debugging. Nuestro diseño deberá garantizar esa sobre tensión sin interferencias y sin dañar la circuitería externa de nuestra aplicación. La solución puede ser tan sencilla como el agregado de un diodo en serie con la resistencia de Pull-Up en este pin, hasta implementar un circuito “buffer” para separar la alta tensión, dependiendo de las condiciones de nuestra aplicación.

Cabe destacar que el pin de IRQ podrá disponerse en la aplicación del usuario en su función primitiva (entrada de interrupción de alta prioridad) teniendo en cuenta que la desaparición de la tensión elevada del pin, libera la inhibición por hardware del COP “Watchdog” que está implementada internamente en los HC908, para ello solo bastará con forzar a “1” lógico el bit “COPD” del registro de configuración “CONFIG 1” o bien mantener una tensión alta (+7,5V a + 9V) en el pin Reset cuando el dispositivo así lo permita (Ver sección “Monitor ROM” del manual de datos del dispositivo HC908 elegido).

❑ **Reloj del Sistema.**

Como la comunicación en el modo “monitor” se establece a un solo hilo y es del tipo NRZ asincrónica, para asegurar el diálogo con la PC se deberá utilizar una frecuencia de reloj que genere un Baud Rate estandar o muy cercano a el, ya que si bien los entornos de desarrollo como el WinIDE o el CodeWarrior disponen de cierta flexibilidad en el manejo del Baud Rate, no podemos afirmar lo mismo con algunas PC que disponen de puertos seriales muy limitados (sin controlador de comunicaciones, pobres en flexibilidad).

Por ejemplo, si con un microcontrolador determinado de la familia HC908 usando un cristal de 9,8304 MHZ obtenemos un Baud Rate de 9600 bps para nuestra comunicación en modo monitor, es perfectamente válido utilizar un cristal más común de 10,000MHZ, ya que el error cometido es de apenas 1,7% de la frecuencia ideal, y es perfectamente tolerable por cualquier UART (puerto COM) de una PC. Esto nos otorgará mayor flexibilidad a la hora de implementar nuestra aplicación y nos reportará mayor facilidad en la consecución de los materiales.

❑ **Pin RESET.**

Si bien el pin de Reset no es fundamental para poder ingresar al modo monitor, es conveniente tener control del mismo, ya que al hacerlo, nuestra herramienta sacará provecho de ello manejando los tiempos de reset del microcontrolador y mejorando la comunicación entre PC y sistema. Además al disponer de ese pin, como se explicó en el punto anterior, se podrá forzar una tensión elevada que mantenga inactivo al COP aun manejando la señal de IRQ de nuestro microcontrolador.

❑ **Pin PTA0 / PTB0 de comunicación a un solo hilo.**

Este pin es fundamental para mantener la comunicación a un solo hilo entre el microcontrolador y la PC. Es el único que no puede ser utilizado durante el Debugging ni antes, ni durante el “Modo Monitor”, ya que de hacerlo se perdería el control entre el MCU y la PC durante el proceso de depuración de código. Aquí sugiero utilizar este pin en aplicaciones menores como el encendido de un LED o la activación de un Buzzer, etc., ya que estas operaciones son de poca monta en todo proyecto y no será fundamental para nosotros perderlas durante la emulación.

El pin PTA0 / PTB0 (según corresponda en el MCU elegido) debe configurarse SIEMPRE como ENTRADA en el DDRA o DDRB respectivamente, se use o no el mismo en una aplicación, ya que el solo hecho de configurarlo como SALIDA durante nuestra emulación, contendría el manejo de dicho pin para nuestra emulación, imposibilitando la comunicación entre el MCU y la PC.

Luego de terminada la etapa de emulación, puede disponerse del pin sin problemas y configurarlo según nuestras necesidades.

□ Pines VDD y VSS.

Estos pines constituyen la alimentación física del microcontrolador y como es sabido, el “control del ciclo de alimentación” juega un papel importantísimo en el manejo del modo monitor de los HC908. Por ello es muy importante dejar disponible una conexión física con estos pines, que puede ser una simple conexión en paralelo con las mismas pero teniendo cuidado de no manejar grandes cargas de corriente (debido a grandes capacitores en VDD, o cargas de otros dispositivos) sobre todo, del tipo capacitivas, ya que retardarían los procesos de POWER – ON y POWER – OFF provocando errores en el ingreso al modo monitor, fundamental para efectuar la Emulación en Tiempo Real.

Bien, hasta aquí hemos tocado puntos referidos al hardware, en próximos artículos abordaremos aspectos referentes a los entornos de trabajo disponibles para esta familia y como sacarles el mayor provecho a los mismos.

Hasta la Próxima!!!!