

## COMENTARIO TECNICO

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



Por Ing. Daniel Di Lella  
Dedicated Field Application Engineer  
[www.edudevices.com.ar](http://www.edudevices.com.ar)  
[dilella@arnet.com.ar](mailto:dilella@arnet.com.ar)



[www.edudevices.com.ar](http://www.edudevices.com.ar)

## *“Cuando el consumo es un problema...”*

### Segunda Parte.

En el artículo anterior, vimos como “apagar” o “encender” el módulo supervisor de tensión de alimentación, conocido como LVI en familia HC908, para ahorrar energía en nuestro sistema portátil. Otro artilugio muy utilizado cuando queremos reducir el consumo de energía en un MCU es orientar al sistema a trabajar la mayor parte del tiempo en “modo dormido” o “modo latente”, siempre y cuando la aplicación lo permita. El truco radica en forzar al sistema a un estado de muy baja o nula actividad, la mayor parte del tiempo posible, de forma tal que la relación sistema “Activo / Dormido” tenga valores de 1: 5 , 1:10 , o a aún mejores. De esta forma la corriente en modo activo se verá notoriamente promediada con la corriente del estado de bajo consumo, dando por resultado una corriente unas 5, 10 o “n” veces menor a la normal en el modo activo del MCU.

Para ello, en muchos HC908 contamos con un módulo de temporización denominado TBM (Time Base Module), que por medio de un cristal de baja frecuencia (generalmente 32,768 KHZ) permite generar interrupciones periódicas tanto si el MCU está en modo activo o bien en algún estado de bajo consumo. De esta forma, por ejemplo, si configuro los registros de este módulo para generar un pedido de interrupción cada 1 segundo, puedo orientar al programa para que ejecute todas las tareas que sean necesarias lo más rápido posible y al termino de ello, colocar al MCU en el estado de STOP (es el estado de menor consumo típico de cualquier MCU).

Previamente a ello, apago todas las fuentes de generación de consumo posibles y habilito solo las interrupciones necesarias (entre ellas las del TBM.... claro), para que al final del tiempo (1 segundo) se genere un pedido de interrupción por el TBM y con ello se “despierte” el sistema desde su estado de “latencia” (STOP). De esta forma, se obtienen dos cosas:

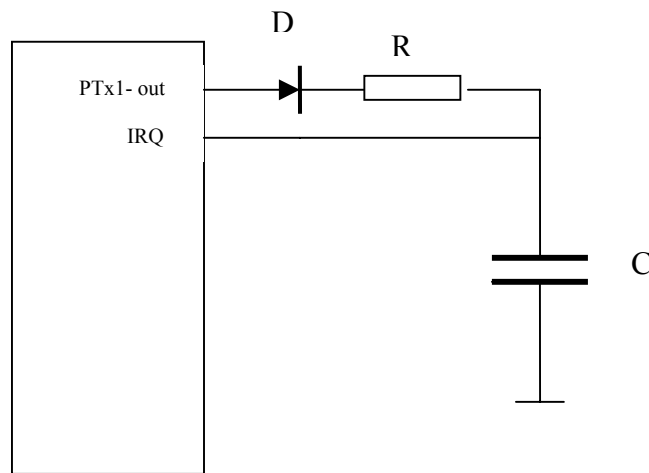
- 1) Una reducción notable en el consumo del sistema sin perder capacidad de control del mismo.
- 2) Una “base de tiempo” muy estable, como para generar, por ejemplo un RTC (Real Time Clock) y con ello mantener un “tracking” del tiempo y de procesos relacionados con el.

Bien, pero no siempre disponemos de un módulo tan a la medida como el TBM, entonces.... Que hacemos?

Utilizar los Timers internos que poseen los HC908 no parecería una buena idea, ya que nuestra solución debería ser de muy bajo consumo, y mantener encendido el módulo del timer implicaría utilizar la instrucción WAIT (esta instrucción solo bloquea el clock al CPU, pero no al resto del MCU) en lugar de la STOP para poner en bajo consumo al sistema, siendo la primera de mayor consumo que la última.

Sin embargo, sería muy interesante lograr una acción “temporizada” que garantice de algún modo despertar al MCU de su letargo a un tiempo determinado a intervalos regulares, pero ... Como hacerlo?

Una solución posible es emplear la entrada de interrupción Externa (IRQ o IRQ1, según el MCU) que poseen todos los HC908 e implementar un circuito mínimo externo que permita “temporizar” la acción de “despertador” del sistema en sus estados de letargo. Un circuito posible podría ser el de la fig. 1.-



- **Fig. 1.- Circuito Externo Mínimo de Temporización.**

El principio de funcionamiento es el siguiente:

El capacitor “C” se irá cargando de forma rápida a un valor máximo cercano a  $V_c = (+V_{DD} - V_{\text{diodo}})$ , por medio de un diodo y un resistor en serie conectados a un puerto de salida del MCU (con un “1” lógico en su salida), donde el tiempo de carga estará limitado por los valores de R y de C, teniendo en cuenta que la corriente máxima de carga no debe superar la corriente máxima del puerto usado como salida.

Una vez cargado el capacitor, se procede a habilitar el pin de IRQ, teniendo en cuenta de hacerlo para que reconozca interrupciones por “nivel”. De aquí en más se procede con la ejecución de las tareas que el MCU debería realizar cuando está en modo “activo”, una vez terminadas todas las tareas se fuerza el puerto de salida utilizado para la carga del capacitor a “cero” lógico y se coloca al sistema en un modo de bajo consumo como STOP. De esta forma, el capacitor irá perdiendo carga e irá bajando su tensión hasta alcanzar el umbral de disparo de la IRQ externa.

Tres, son las fuentes de descarga del capacitor:

- 1) Corriente de pérdida del propio capacitor.
- 2) Corriente de “sangría” externa formada por la juntura del diodo polarizada en inversa y el resistor de carga en serie (este último puede obviarse en el cálculo debido al poco valor de la corriente inversa del diodo).
- 3) Corriente de Drenaje de entrada del pin de IRQ, que es un HC MOS y ese dato está en el manual del HC908 elegido.

Una vez generada la interrupción, debemos atenderla y tener en cuenta que deberemos volver a “recargar” a nuestro capacitor externo para una nueva temporización.

Los tiempos logrados con este esquema, lamentablemente no son para nada exactos, y por lo general su estabilidad térmica no será de las mejores, y además involucra el uso de la interrupción externa y de otro puerto adicional, pero es un circuito lo bastante sencillo y económico como para tenerlo en cuenta cuando se quiera lograr esquemas de temporización (para obtener bajos consumos) sin recurrir a timers o módulos especiales.

Es todo por el momento, hasta nuestro próximo encuentro!!

[www.edudevices.com.ar](http://www.edudevices.com.ar)

