

COMENTARIO TÉCNICO

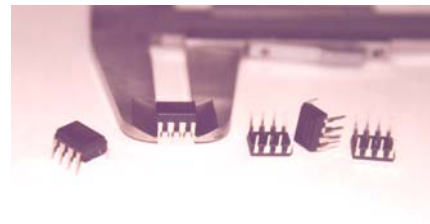
Buceando en el HC908.....

Por Ing. Daniel Di Lella
Dedicated Field Application Engineer
www.edudevices.com.ar
dilella@arnet.com.ar



www.edudevices.com.ar

Guía de Conversión de los MC908QT/QYx a MC908QTxA/QYxA.



1era Parte.

Mucha gente me ha preguntado en los últimos tiempos si existen diferencias entre la “**Vieja**” generación de los HC908QT / QY (QT1 / QT2 QY1 / QY4) y la **nueva generación** de los HC908QTxA/ QYxA (QT1A / QT2A QY1A / QY4A) La respuesta es **SI, hay diferencias, y todas ellas apuntan a tener en nuestras manos un producto con más prestaciones y a igual costo que los anteriores dispositivos.**

En esta “guerra” de productos que se viene experimentando en el mundo de los microcontroladores de 8 bits, el segmento de los MCUs “chiquitos” (de 6 a 16 pines) es uno de los más competitivos y **Freescale** está dando batalla al lanzar al mercado una nueva versión muy mejorada de los populares “QT / QY”.

Si bien en artículos anteriores ya les había presentado a los nuevos integrantes de la familia “Q”, el presente artículo pretende describir las diferencias entre la familia Q clásica y la nueva, aportando una guía de soluciones para adaptar estos nuevos “Q” en proyectos ya existentes o bien, aprovechar las mejoras en nuevos proyectos.

Pasemos entonces, a la descripción de cada una de las mejoras.....

Nuevo módulo conversor A/D de 10 bits.

Los nuevos QTxA / QYxA poseen un módulo A/D de 10 bits de resolución frente a los 8 bits de los QT / QY clásicos. Este módulo permite modos de conversión de 10 y 8 bits. Este incremento de la resolución puede ser útil en muchas aplicaciones como por ejemplo, medición de temperatura.

Las características del nuevo ADC de 10 bits incluyen:


- Dos nuevos canales A/D disponibles en PTB0 y PTB1 permiten mayor flexibilidad especialmente durante la depuración de código en modo monitor.
 - Una limitación que tienen los QT / QY clásicos es que muchos canales del conversor A/D están multiplexados y por ello el acceso a ellos está limitado durante el modo monitor. Al agregar canales extra en el puerto PTB se resuelve esta limitación.
- El A/D de los QTxA / QYxA puede operar mientras el MCU está en **modo STOP**, permitiendo ahorro de energía en aplicaciones que así lo requieran. Este modo de operación también permite obtener conversiones con bajo nivel de ruido, **fundamental para lograr mediciones estables y precisas.**
- 3 fuentes de referencia de clock:
 - FBUS (frecuencia de BUS).
 - ACS (Alternate Clock Source) (Clock Externo)
 - ACLK (Asynchronous Clock) (Clock Interno Dedicado).

El ACLK (clock asincrónico interno) permite que el módulo conversor funcione durante el modo STOP y solo está habilitado durante ese modo.

El A/D podrá ahora correr en modo STOP si el Bit ACLKEN está en “1”, habilitando de esta forma el clock interno dedicado (ACLK). Utilizando el modo STOP durante la conversión nos garantizará una operación más “tranquila” en cuanto a ruido se trate, logrando conversiones más precisas al no tener “ruido” generado por el propio consumo del microcontrolador. Este nuevo Bit, figura como “Reservado” en los QT / QY clásicos, por lo que no afectaría a rutinas escritas para los mismos.


Registros afectados:

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
Write:								
Reset:	0	0	0	1	1	1	1	1

 = Unimplemented


ADSCR – A/D Status and Control Register (\$003C).

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	AD9	AD8
Write:								
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

ADRH – A/D Data Register High (\$003D) (Nuevo registro).

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Write:								
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

ADRL – A/D Data Register Low (\$003E)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ADLPC	ADIV1	ADIV0	ADICLK	MODE1	MODE0	ADLSMP	ACLKEN
Write:								
Reset:	0	0	0	0	0	0	0	0

ADCLK – A/D Clock Register (\$003F)

Como se puede observar, solo **se agrega un nuevo registro (ADRH / \$003D)** que en los QT / QY clásicos figuraba como reservado, y es el que contiene los dos bits más pesados del resultado de la conversión A/D a 10 bits de resolución.

Además se ha agregado la opción de un tiempo largo de muestreo para conservar energía a expensas de un mayor tiempo de conversión. Para ello aparece el bit ADLSMP en el registro ADCLK (en los QT / QY clásicos es un bit reservado en dicho registro).

Bueno, pero que pasa si soy usuario de los “clásicos” QT / QY y tengo implementados programas que utilizan el conversor A/D..... ¿Tengo que modificar algo?

Si utilizo conversión a 8 bits, la respuesta es NO, ya que los nuevos QYxA / QTxA mantienen compatibilidad hacia “abajo”.

Por ejemplo, veamos una rutina de conversión que funcionaría en ambos chips.

```
TEMPE_LOOP!    MOV #$50,ADCLK      ;Clock Reg. -- A/D clock / 4 (1mhz aprox.) (8 BITS)
                MOV #$00,ADSCR      ;ADC -- ON -- A/D CH 0 (PTA0)
Coco_5?        BRCLR 7,ADSCR,Coco_5? ;si coco=1 -- puedo leer ADRL (8 BITS truncados)
                LDA ADRL            ;y guardo valor en TEMP_ACT
                STA TEMP_ACT        ; (TEMPERATURA ACTUAL)
                BRA TEMPE_LOOP!    ;En loop infinito....
```

En esta rutina, se configura al conversor A/D para que trabaje en modo 8 Bits justificados a la derecha, se utiliza FBUS como clock de referencia y se lo divide por 4 para lograr una frecuencia de trabajo del A/D de aproximadamente 1 Mhz, ya que por default la frecuencia del oscilador interno es de 12,8 Mhz, o sea 3,2 Mhz de FBUS. Si nos fijamos atentamente en los distintos registros de configuración del módulo conversor A/D, veremos que la configuración utilizada para los QT / QY clásicos es compatible con los nuevos QTxA / QYxA por lo que una rutina escrita para los primeros, es perfectamente funcional en los nuevos Q.

Ahora, si quisiéramos utilizar todo el potencial del conversor A/D y **efectuar conversiones a 10 Bits de resolución** la rutina debería modificarse de la siguiente forma:

```
ADQ_DATA!     MOV #$54,ADCLK      ;A/D Clock Reg. -- FBUS/4 -> APROX 1MHZ
                MOV #$00,ADSCR      ; A/D CH 0 (PTA0), NO INT'S, ONE CONV.
Conv_ok2?     BRSET COCO,ADSCR,Conv_ok2! ;si coco=1 -- puedo leer ADR
                BRA Conv_ok2?       ;Sino sigo en loop hasta fin conv.

Conv_ok2!     LDA ADRH0            ;Guardo los 10 bits leídos (ADRH0/ADRL0)
                STA CANAL1H        ;en CANAL1L y CANAL1H (no signados)
                LDA ADRL0          ;en el orden aquí mostrado ...
                STA CANAL1L        ; parte “alta” lero, luego “baja”
```

Lo único que se configuró aquí fueron los **bits “reservados”** en el registro ADCLK para los QT / QY clásicos, y que en los nuevos están disponibles para configurar los distintos modos de trabajo del conversor A/D.

Mejoras en el módulo Oscilador (OSC).

Los nuevos “QYxA / QTxA” tienen incorporadas mejoras notables en el módulo oscilador (OSC) con respecto a los clásicos QT / QY.

Las nuevas características son las siguientes:


- Los Bits ICFS1 / ICFS0 en el registro OSCSC (Oscillator Status and Control Register) Permiten configurar al Oscilador Interno para operar en 1 Mhz, 2 Mhz y 3,2 Mhz de frecuencia de Bus. En los QT / QY clásicos estos bits estaban reservados, por lo que las rutinas escritas para estos funcionan con un FBUS = 3,2 Mhz en los nuevos Q.
- Los Bits ECFS1 / ECFS0 en el mismo registro OSCSC, permiten configurar el rango de la frecuencia del cristal externo a utilizar. Con esta opción uno puede elegir el uso de cristales de BAJA FRECUENCIA (32 Khz a 100 Khz), MEDIA FRECUENCIA (1 Mhz a 8 Mhz) y ALTA FRECUENCIA (8 Mhz a 32 Mhz).

Otra mejora en el módulo Oscilador es que se puede conmutar entre el Oscilador Interno y el Oscilador Externo en cualquier momento. Esto es particularmente útil cuando uno quiere trabajar a muy baja velocidad y muy bajo consumo pero con muy buena precisión en la base de tiempo al utilizar un cristal externo de 32khz y cuando la aplicación necesita de alta velocidad de procesamiento, conmutar al oscilador interno, por ejemplo el de 3,2 Mhz de FBUS, sin que por ello el sistema tenga algún inconveniente. De igual forma se puede conmutar las distintas frecuencias disponibles del oscilador interno sin inconveniente alguno.

Registros Afectados:

El bit **OSCOPT** no está más disponible en el Registro **CONFIG2** y ahora reside en el registro **OSCSC**.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	OSCOPT1	OSCOPT0	ICFS1	ICFS0	ECFS1	ECFS0	ECGON	ECGST
Write:								
Reset:	0	0	1	0	0	0	0	0

 = Unimplemented

OSCSC – OSC Status and Control Register ()

OSCOPT1	OSCOPT0	Oscillator Modes
0	0	Internal oscillator (frequency selected using ICFSx bits)
0	1	External oscillator clock
1	0	External RC
1	1	External crystal (range selected using ECFSx bits)

ICFS1	ICFS0	Internal Clock Frequency
0	0	4.0 MHz
0	1	8.0 MHz
1	0	12.8 MHz — default reset condition
1	1	Reserved

ECFS1	ECFS0	External Crystal Frequency
0	0	8 MHz – 32 MHz
0	1	1 MHz – 8 MHz
1	0	32 kHz – 100 kHz
1	1	Reserved

Bueno amigos, en próximos artículos les iré detallando las otras mejoras implementadas en estos nuevos QTxA / QYXA.....

Continuará.....

..... Hasta la próxima!!!!

www.edudevices.com.ar

