

Revisiones	Fecha	Comentarios
0	13/08/07	

Dado que tanto la implementación de multiplexado para displays de 7-segmentos como la de software timers utilizan el timer de 8-bits, en aplicaciones que requieran ambos conceptos, éstos utilizan dicho timer de forma diferente. La presente nota de aplicación se basa en CAN-079, proveyendo un esquema de multiplexado para hasta cuatro dígitos de 7-segmentos, y agregando a la misma el esquema desarrollado en CAN-071, que provee un sencillo y eficiente esquema de software timers de 8-bits.

La frecuencia de interrupción del timer viene dada por la cantidad de dígitos del display. No lo desarrollamos en CAN-079, pero la idea es tener una cantidad de parpadeos superior a la mínima detectada por el ojo. Si tomamos esta frecuencia como de 50Hz, entonces, para un display de cuatro dígitos el timer de 8-bits interrumpe cada 5ms. Esta interrupción nos controla los timers cortos, que cuentan 'ticks' en vez de centésimas de segundos como en CAN-071. Estos 'ticks' se dividen por software hasta lograr un segundo, base de los timers largos.

```

clockrate    equ 4000000          ; clock
muxfreq      equ 50              ; frecuencia de actualización por dígito
prescaler    equ 256             ; prescaler (configurar los bits !)
TMRCTRL      equ 10010111b      ; prescaler=256 (4MHz/256=>64us)
TICKS        equ muxfreq*DIGITS
TMRCONST     equ 256-(clockrate/(prescaler*TICKS))

```

El tiempo de demora de esta implementación es de $\left\lceil x \cdot \frac{1}{2} \right\rceil \oplus \frac{1}{2}$ cuentas. Por ejemplo: 28 en `_TK_TIMERS` (cuentas de 5ms) es en realidad $137,5 \oplus 5 \text{ ms}$ y 5 en `_SEC_TIMERS` es $4,5 \oplus 0,5 \text{ seg}$.

Para mayor información, observe las notas de aplicación CAN-071 y CAN-079. La forma de utilización es equivalente y puede apreciarse en el listado adjunto. La diferencia principal es que los timers cortos ahora son `_TK_TIMERS` (por 'ticks' en vez de centésimas de segundo).