Mosaic Industries

Regenerative Timer

APPLICATION NOTE
MI-AN-033

Summary

The following demonstrates how to produce a regenerative timer.

Description

The following program uses the 68HC11 16 bit free running counter to generate a 125 msec interrupt. The interrupt service routine keeps track of the number of interrupts and when it gets to its set number, MULTIPLIER, it calls a user function. If the set number is 32, then the timer is 32 * 125 msec, which is 4 seconds. The timer also resets itself without losing any accuracy. For example, the user routine will be called on 4, 8, 12 ...etc. seconds until it has been disabled. See chapter 4, page 2 in the QED Hardware Manual for background on the counter.

This interrupt will call an external function, defined by the user. This routine should execute much faster than 125 msec. It is good practice to keep interrupt service routine execution time to a minimum.

The output compare function will actually generate the interrupt. This code uses output compare 3 (OC3). The free running counter with factory settings will overflow every 131.1 msec. The OC3 interrupt will occur every 125 msec but the user function will only be called every MULTIPLIER * 125 msec. The OC3 output pin, PA5, is not affected. Other output compares can be used instead of OC3, keep in mind that OC4 is used by serial2 and OC2 is used by the multitasker.

```
4 USE.PAGE
ANEW TEST.TIMER
HEX
10 WIDTH !

\ define the relevant control registers
800E  REGISTER:   TCNT              \ Timer counter register
801A  REGISTER:   TOC3              \ Output compare 3 register
8022  REGISTER:   TMSK1             \ Timer interrupt mask register
8023  REGISTER:   TFLG1             \ Timer interrupt flag register

20    CONSTANT    OC3.MASK          \ Isolates OC3 interrupt flag & mask bits

DECIMAL

62500 CONSTANT    125MSEC.CNT       \ Number of 2µsec counts in 125 msec

\ MULTIPLIER can be changed from 1 which is 0.125 sec to
\   65535(2hrs 16 min 31.875 seconds)
32    CONSTANT    MULTIPLIER        \ Number of 125msec in 4 seconds

\ By changing the 125MSEC.CNT and MULTIPLIER, the time interval can easily
\ be changed.  The 125MSEC.CNT can not be greater than 65535 (131msec)

VARIABLE    INTERRUPT.CNTR          \ Used to count number of interrupts
```

```
\ USER.ROUTINE is the routine that will be called every 4 seconds
\ Remember to keep it short since all other interrupts are disabled while this
\ routine is being executed.
: USER.ROUTINE
      ASCII . EMIT            \ Simply emit a period to show routine is working
                              \ Be careful, emitting a character at 9600 bps takes
                              \ about 1 ms
      \ enter your code here
;

\ DISABLE.OC3 will stop the interrupt-based timer.
: DISABLE.OC3
      OC3.MASK TMSK1 CLEAR.BITS       \ disables interrupts on OC3
;

\ OC3.SERVICE is an interrupt service routine that is entered every 125ms
\ The user defined routine is called from here.
: OC3.SERVICE
      OC3.MASK TFLG1 C!         \ clear the interrupt flag that got us here
      TOC3 @ 125MSEC.CNT + TOC3 !     \ set the next time for interrupt to occur
      1 INTERRUPT.CNTR +!             \ increment interrupt counter
      MULTIPLIER INTERRUPT.CNTR @ =
      IF                              \ is this the nth time interrupt occurred?
            0 INTERRUPT.CNTR !        \ if so, reset interrupt counter
            USER.ROUTINE              \ call user routine
      ENDIF
;


\ INIT.OC3 installs the interrupt handler, initializes the timer set point
\ and enables OC3 interrupt mask.
: INIT.OC3 ( -- )
      0 INTERRUPT.CNTR !              \ initialize variable to zero
      TCNT @ 125MSEC.CNT + TOC3 !     \ set time for next interrupt to occur
      DISABLE.OC3
      CFA.FOR OC3.SERVICE OC3.ID ATTACH
                                      \ install interrupt routine OC3.SERVICE
      OC3.MASK TFLG1 C!               \ resets the OC3 interrupt flag
      OC3.MASK TMSK1 SET.BITS         \ enables the OC3I interrupt flag
      ENABLE.INTERRUPTS               \ globally enable interrupts
      \ NOTE: if you want the Output Compare to change the output
      \ pin (Port A5), you need to also setup the following:
      \ Setup the timer control (TCTL1) register to toggle, clear or set
      \ on successful output compares. See Motorol MC68HC11F1 Technical Data
      \ Pages 9-10 and 9-11.
      \ You also should change the DISABLE.OC3 routine to set the TCTL1
      \ register to take no action on output compare OM3 = OL3 = 0
;
```

The information provided herein is believed to be reliable; however, Mosaic Industries assumes no responsibility for inaccuracies or omissions. Mosaic Industries assumes no responsibility for the use of this information and all use of such information shall be entirely at the user's own risk.