



Summary

The following software uses a free running counter to generate a 250msec interrupt until it is disabled.

Description

This program uses the 68HC11 16 bit free running counter to generate a 250 milli-second interrupt 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. The output compare function will actually generate the interrupt. This code uses output compare 4 (OC4). The free running counter, set with a prescale factor of 4 for the 8MHz board, will overflow every 131.1 ms. The OC4 interrupt will occur every 125 ms but the user function will only be called every 250 ms. The OC4 output pin, PA4, is not affected.

```
HEX
10 WIDTH !

\ MI-AN-020
\ 250 milli-second timer
\ define the relevant control registers
800E REGISTER: TCNT          \ Timer counter register
801C REGISTER: TOC4          \ Output compare 4 register
8022 REGISTER: TMSK1         \ Timer interrupt mask register
8023 REGISTER: TFLG1         \ Timer interrupt flag register

10  CONSTANT  OC4.MASK          \ Isolates OC4 interrupt flag & mask bits
F424 CONSTANT  125MS.CNT        \ Number of 2 s counts in 125 ms
2   CONSTANT  MULTIPLIER        \ Number of 125ms in 250ms
\ By changing the 125MS.CNT and MULTIPLIER, the time interval can easily
\ be changed. The 125MS.CNT should not be greater than FFFF (131ms)

VARIABLE  INTERRUPT.CNTR        \ Used to count number of interrupts

\ USER.ROUTINE is the routine that will be called every 250 ms
USER.ROUTINE
    \ enter your code here
;

\ DISABLE.OC4 will stop the interrupt-based timer.
DISABLE.OC4
    OC4.MASK TMSK1 CLEAR.BITS    \ disables interrupts on OC4
;
```

```

\ OC4.SERVICE is an interrupt service routine that is entered every 125ms
\ The user defined routine is called every other time this service routine is
\ entered.
OC4.SERVICE
  OC4.MASK TFLG1 C!          \ clear the interrupt flag that got us here
  TOC4 @ 125MS.CNT + TOC4 !  \ set the next time for interrupt to occur
  1 INTERRUPT.COUNTER +!    \ increment interrupt counter
  MULTIPLIER INTERRUPT.CNTR @ =
  IF                          \ is this the 2nd time interrupt occurred?
    0 INTERRUPT.CNTR !      \ if so, reset interrupt counter
    USER.ROUTINE           \ call user routine
  ENDF
;

\ INIT.OC4 installs the interrupt handler, initializes the timer set point
\ and enables OC4 interrupt mask.
: INIT.OC4 ( -- )
  0 INTERRUPT.CNTR !          \ initialize variable to zero
  TCNT @ 125MS.CNT + TOC4 !  \ set time for next interrupt to occur
  DISABLE.OC4
  CFA.FOR OC4.SERVICE OC4.ID ATTACH
                                \ install interrupt routine OC4.SERVICE
  OC4.MASK TGLG1 C!          \ resets the OC4 interrupt flag
  OC4.MASK TMSK1 SET.BITS    \ enables the OC4I interrupt flag
  ENABLE.INTERRUPTS          \ globally enable interrupts
;

```

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.

Mosaic Industries

5437 Central Ave Suite 1, Newark, CA 94560

Telephone: (510) 790-8222

Fax: (510) 790-0925