Summary

This application note shows an alternative to frequent use of the smartwatch.

Description

The basic idea is to use the built-in elapsed time clock that is associated with the timeslicer (it uses the OC2 interrupt). The elapsed time is added to a "base time" which is read from the smartwatch only at system startup or at very infrequent intervals (say once per day).

The advantage is that the READ.ELAPSED.TIME and READ.ELAPSED.SECONDS routines disable interrupts for less than 14 microseconds, compared to a total 1 msec interrupt-disable time for READ.WATCH.

The disadvantage of this technique is that it is actually a bit slower than READ.WATCH; UPDATE.TIME executes in 1.5 msec compared to about 1 msec for READ.WATCH. Thus you wouldn't want to call UPDATE.TIME from inside an interrupt routine!

```
ANEW ELAPSED.TIME.CODE
DECIMAL                              \ compile in decimal base
8 WIDTH !

CODE READ.WATCH   ( -- 100ths.sec\sec\min\hrs\day\date\month\yr )
\ re-define it to disable interrupts, making it re-entrant.
\ Will generate a non-unique warning at compile time
      TPA
      PSHA
      SEI                            \ disable interrupts
      CALL READ.WATCH
      PULA
      TAP                            \ restore interrupt flag to prior state
      RTS
END.CODE

VARIABLE    HUNDREDTH.SECONDS.BASE  \ these variables hold the base time
VARIABLE    SECONDS.BASE
VARIABLE    MINUTES.BASE
VARIABLE    HOUR.BASE
VARIABLE    DAY.BASE
VARIABLE    DATE.BASE
VARIABLE    MONTH.BASE
VARIABLE    YEAR.BASE
```

```
: INIT.BASE.TIME  ( -- )
\ call this at powerup and once per day to re-establish the base time.
\ Note that this routine uses variables and so is not re-entrant;
\ it should be called from only 1 location in the application.
      START.TIMESLICER        \ also globally enables interrupts
      READ.WATCH              ( -- 100ths.sec\sec\min\hrs\day\date\month\yr )
      INIT.ELAPSED.TIME       \ set TIMESLICE.COUNT = 0\0
      YEAR.BASE !             \ now save base time in the variables
      MONTH.BASE !
      DATE.BASE !
      DAY.BASE !
      HOUR.BASE !
      MINUTES.BASE !
      SECONDS.BASE !
      HUNDREDTH.SECONDS.BASE !      ( -- )
;

: +MOD      ( u1\u2\u3 -- sum\carry )
      \ does u1+u2 subject to rollover value u3; u3 = 1+ max.allowed.value
      >R + R> U/MOD
;

: UPDATE.TIME     ( -- u1\u2\u3\u4\u5 )
\ meaning:        ( -- 100ths.sec\sec\min\hrs\new.day? )
\ execution time appx 1.5 msec.
\ This routine can be used in conjunction with the
\ variables YEAR.BASE MONTH.BASE DATE.BASE and DAY.BASE
\ to provide the same information as READ.WATCH.
\ new.day? flag is true if sum of base.time and current elapsed.time
\ falls on a different day than the base.time;
\ this flag can be used to signal a new call to INIT.BASE.TIME.
\ Note that timeslicer must be running to use this routine;
\ call START.TIMESLICER at system startup.
      READ.ELAPSED.TIME       ( -- msec\sec\min\hrs\days)
      4 ROLL 10 /             ( -- sec\min\hrs\days\hundredth.sec)
      HUNDREDTH.SECONDS.BASE @ 100 +MOD   ( --sec\min\hrs\days\new.100ths\sec.carry)
      5 ROLL +                ( -- min\hrs\days\new.100ths\sec+carry)
      SECONDS.BASE @ 60 +MOD  ( -- min\hrs\days\new.100ths\new.sec\min.carry )
      5 ROLL +                ( -- hrs\days\new.100ths\new.sec\min+carry)
      MINUTES.BASE @ 60 +MOD  ( -- hrs\days\new.100ths\new.sec\new.min\hr.carry)
      5 ROLL +                ( -- days\new.100ths\new.sec\new.min\hr+carry)
      HOUR.BASE @ 24 +MOD     ( -- days\new.100ths\new.sec\new.min\new.hr\day.carry)
      5 ROLL + BOOLEAN        ( -- new.100ths\new.sec\new.min\new.hr\new.day?)
;


\ routines used to benchmark the execution time:
: SEE 1000 0 DO UPDATE.TIME 5 NDROP LOOP ;
\ 1.58 ms;

: SEE2 1000 0 DO READ.ELAPSED.TIME 5 NDROP LOOP ;
\ 0.84ms without rolls, extra math, etc.

: SEE3 1000 0 DO
>ASSM DEY DEY DEY DEY DEY DEY DEY DEY DEY DEY >FORTH
5 NDROP LOOP ;                \ subtract 35 usec looping and dropping overhead
```

Mosaic Industries

5437 Central Ave Suite 1, Newark, CA 94560 Telephone: (510) 790-8222 Fax: (510) 790-0925