```
\  *** ESP32forth - Timer test with interrupt ***
\ This test measures the BOOT switch depression time
\ by activating timer1 for the duration of the
\ key depression and shows the measure value.
\ The ESP32 has 4 64-bit timers: 0-3 clocked
\ by a 40Mhz clock in ESP32forth.
\ A timer value is represented by 2 32-bit values: lo hi.
\ At 40Mhz, lo and hi this could count for 14 623.5 years.
\ Lo alone will do half: 7 311.8 years.
\ For this test I will have enough of the lo count.
\ An action must be defined for the falling
\ and rising edge interrupts on swpin depression.
\ The default ISR is set to:
\ - initialize and start timer1 on the falling edge,
\ - stop timer1 and store it value to memory
\   on the rising edge.
\ The value of timer1 is the swpin depression duration in 1/40million sec.
\ A calculation will be performed to convert the time in ms.

\ *** Define VARIABLE, VALUE or CONSTANT
\ Define the timer1-lo VALUE for memory storage
\ of hardware timer1 lo section count.
0 value timer1-lo
\ Define the switch pin used in the test as a VALUE.
0 value swpin-val
```

```
\ *** Define a test word to print swpin value.
: .swpin cr ." The switch pin is: " swpin-val . ;

\ *** Define  words to configure the switch pin.
\ Ensure that INTERRUPTS vocabulary is active.
interrupts

\ These words set the switch pin as input with pullup.
\ These words use a pin number on the TOS.
\ In this test we use swpin-val
\ to provide this pin number. ( pin --- )
: pullup_en gpio_pullup_en drop ;
: pullup_dis gpio_pullup_dis drop ;
: set-pullup dup input pinmode pullup_en ;
\ Defining a word to setup our desired pin.
: sw-setup swpin-val set-pullup ;

\ Setting up our desired switch swpin.
sw-setup
\ The following words require TIMERS vocabulary.
timers
```

```
\ *** Define various words used to control hardware timer1.
\ Initialize t1. ( --- )
: t1-init 0 0 1 timer! ;
\ Start t1. ( --- )
: t1-on 1 1 enable! ;
\ Stop t1. ( --- )
: t1-off 0 1 enable! ;
\ Read timer1. ( --- lo hi )
: t1-rd 1 timer@ ;
\ Consert timer1 raw value to ms.
: t1-duration>ms t1-rd drop 40000 / ;
\ Store the timer value in timer1-lo memory VALUE.
: t1-mem! t1-rd drop to timer1-lo ;

\ Define words to control timer-lo VALUE in memory.
\ Reset timer1-lo to 0.
: 0>timer1-lo 0 to timer1-lo ;
\ Print timer1-lo unconverted rw value.
: .timer1 cr ." Timer1 value is: " timer1-lo . ;
\ Define a word to convert swpin depression duration to ms.
\ The drop is discarding the unused hi part of timer1.( --- ms )
: duration>ms timer1-lo 40000 / ;
\ Define a word that prints a swpin depression duration message
: .swpin-dur cr ." The switch depression lasted " duration>ms  . ." ms." ;
```

```
\ *** Define words for the falling and rising edge actions.
\ Initialize and start timer1on falling edge.
: H>Ldo t1-init t1-on ;
\ Stop timer1 and store its value in memory on rising .
: L>Hdo t1-off t1-mem! ;
\ Define the swpin-ISR.
: swpin-ISR swpin-val digitalread 0= if H>Ldo
  else L>Hdo then ;

\ The following words require the INTERRUPTS vocabulary.
interrupts

\ *** Define a word to set the swpin to generate
\ interrupts when depressed and released.
\ These interrupts then execute ISRs to control timer1
\ to measure the duration of the key depression.
: setISR-swpin ['] swpin-ISR swpin-val pinchange ;

\ Execute setISR-swpin to activate sw-pin interrupts.
setISR-swpin

\ Note: pinchange sets up a pin to cause an interupt
\ on any change of pin level: falling and rising
\ edge. A switch depression generates 2 interrupts.
\ *** Define words to print the swpin depression duration
\ each time the BOOT switch is depressed.
: swpin-test 0>timer1-lo begin timer1-lo 0 > if .swpin-dur then 0>timer1-lo key? until ;

\ Note: Using an external push button on pin 15
\ caused multiple interrupts due to switch bounce.
\ The built-in BOOT and EN switches
\ have a 0.1uF capacitor across the switch
\ to prevent the switch bounce effect.
\ Adding a 0.087uF capacitor across my bouncy switch
\ on pin 15 solved the bounce problem.
\ Since the definition vocabulary was left at FORTH,
\ all these words were defined in the FORTH
\ vocabulary. All these test words will be lost
\ on board reset.
\ If required, the ESP32forth can save these word
\ definitions in a flash memory blocks
\ or a file in flash.
\ For the moment, after a board reset,
\ I only use cut and paste from my editor
\ to keep this test simple for a newcomer like me.
\ It is finally working properly except for occasional switch bounce...
\ Start the test by entering the following words.
\ Depress the BOOT switch to see
\ the swpin depression duration.
\ To stop the test depress the ENTER key.
\ When out of the test, interrupts are still working.
\ Depress the BOOT switch and type .swpin-dur
\ to see the BOOT switch depression duration.
\ Various timer1 words can also be used on their own.
\ Ex: Measure 10ms delay with:
\ t1-init t1-on 10 ms t1-off t1-duration>ms cr .
\ Or make this a test word with:
\ : tst-10ms t1-init t1-on 10 ms t1-off t1-duration>ms cr . ;
\ For both the interpreted and the compiled versions,
\ the results fluctuate between 9 and 10 ms. Why?
\ Probably because the integer division
\ discards the remainder portion.
\ Also, the interpreted version includes word search time
\ which is not included in the compiled test version.
\ So the word search process would require less than 1 ms.
\ The following word starts test looping test that shows the
\  BOOT switch depression duration on each depression.
swpin-test
```