```
#include <xc.inc>

; Periodically blinks a LED connected to RB0 with
; a period of about 1 sec (999.975 ms to be precise).
; Assumes that the simulator instruction frequency is
; set to 1 Mhz.

counter1 SET 0x00
counter2 SET 0x01
counter3 SET 0x02
toggle SET 0x03

PSECT code

MAIN:
    clrf counter1
    clrf counter2
    clrf counter3
    movlw 0x0f
    movwf ADCON1 ; all pins digital (as opposed to analog)
    clrf TRISB   ; all pins output

LOOPFOREVER:
    call timer_task; task code
    call led_task; task code

    goto LOOPFOREVER

led_task:
    btfsc toggle, 0; we return if the toggle is clear
    goto changeLedState
    return

    changeLedState:
        movlw 0x01
        xorwf LATB; change the state of pin0
        bcf toggle, 0
        return

timer_task:
    ; increment a counter and check its value
    ; if the counter exceeds a thresold inform
    ; the led_task

    incf counter1
    bz overflow1; happens after 256 increments = 256 us
    return

    overflow1:
        incf counter2      ; counter1 is now zero, we increment counter2
        bz overflow2First  ; overflowed once
        btfsc counter3, 0
        goto overflow2AndSome
        return

    overflow2First:
        incf counter3     ; record that an overflow occurred
        return

    overflow2AndSome:
        movlw 4
        cpfseq counter2    ; if counter2 reached 4 skip return
        return
        clrf counter1      ; clear everything and toggle the led
        clrf counter2
        clrf counter3
        bsf toggle, 0; counter1 overflowed 256 x 256 + 4 x 256  times
        return

END
```