```
;* TIMER1 TOGGLE LEDs on PORTB in NORMAL MODE
;* MUST use CLOCK 8MHz     (fs=200Hz, Ts=DW=5msec )
;* 5/12/16
;*********************************************************************
; BLOCK DIAGRAM
; start:
;   set timer in normal mode AND stop timer
;   set timer to 0xFFFF-40000 (63BF)
;   clear TOV1
;   start timer
;   waitloop:
;       is TOV1 set ?
;       if no goto waitloop
;       if yes change LEDs (i.e. execute main program that does things)
;   jmp start

.include "m32def.inc"

reset:
          ldi   R16,0b11111111        ;set PB0-7 as outputs (STK500 LEDs)
          out   DDRB,R16

          ldi R17,0b00000000          ; WGM for normal mode (i.e. counter counting up to FFFF) set to zero
                                      WGM10bit0 and WGM11bit1
          out TCCR1A, R17

forever:
          ldi R18,0b00000000          ; WGM for normal mode (i.e. counter counting up to FFFF) set to zero
                                     WGM12bit3 and WGM13bit4
          out TCCR1B, R18             ; Also, set CS12bit2, CS11bit1 and CS10bit0 to 000 to stop counting
                                     before loading TCNT1

          ldi R20, high(0xffff-40000); 5ms intervals i.e. 40000cycles at 1/8usec per cycle
          ldi R21, low(0xffff-40000)
          out TCNT1H, R20             ;load timer high byte FIRST since it is stored internally in a
                                     temporary location until the low byte is written
          out TCNT1L, R21             ; now that high byte is loaded, load timer low byte


          ldi R19, 0b00000100         ;clear timer 1 overflow flag TOV1bit2 by writing a logic 1 to it as
                                     the datasheet says pg 113
          out TIFR, R19
```

```
        ldi R22, 0b00000001     ; WGM for normal mode (i.e. counter counting up to FFFF) set to zero
                                WGM12bit3 and WGM13bit4
        out TCCR1B, R22         ; Also, set CS12bit2, CS11bit1 and CS10bit0 to 001 starts counting

waittimer:
        in R23, TIFR
        sbrs R23, TOV1          ;skip next instruction if TOV1 flag is set i.e. after the timer
                                overflows past FFFF
        rjmp waittimer          ;loop while TOV1 flag is not set


        ldi R24, 0b11111111     ;timer1 has reached ffff and TOV1 is set, so do something (LED toggle)
        eor R25, R24
        out PORTB, R25

        jmp   forever
```