```
;* TIMER1 CHANGES LEDs on PORTB (CTC Mode)
;*
;* MUST use CLOCK 8MHz      fs=200Hz, Ts=DW=5msec
;*
;***************************************************************************
.include "m32def.inc"

;***** Global register variables
.def      temp     =R20
.def      tempL    =R19
.def      tempH    =R18
.def      leds     =R17

.org 0x000

reset:                                          ;Main program entry point on reset

                ldi temp, high(RAMEND)          ;Set Stack Pointer to top of SRAM (for ATmega32 it is $85f)
                out SPH, temp
                ldi temp, low(RAMEND)
                out SPL, temp

                ldi      leds,0b11111111        ;turn OFF all STK500 LEDs (do this before DDRB to avoid LEDs turning on for 2usec)
                out      PORTB,leds

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

; TIMER 1 in CTC Compare Match Mode with polling (Timer 1 will count from 0x0000 to OCR1A and toggle OC1A on OCF1A flag set)  (this DOES create accurate durations)

                sbi DDRD, PD5                   ; PD5 i.e OC1A set as output

                ldi temp, 0b01000000            ; WGM for CTC mode (i.e. counter counting from 0x0000 to OC1RA) set to zero WGM10bit0 and WGM11bit1
                out TCCR1A, temp                ; set COM1A1bit7 and COM1A0bit6 to 01 to enable toggling of OC1A (i.e. PD5) when OCR1A value is reached

                ldi temp, 0b00001000            ; WGM for CTC mode (i.e. counter counting from 0x0000 to OC1RA) set WGM12bit3=1 and WGM13bit4=0
                out TCCR1B, temp                ; Also, set CS12bit2, CS11bit1 and CS10bit0 to zero to stop counting before loading TCNT1 and OCR1A
```

```
            ldi tempH, high(40000)  ; high byte of upper limit to count upto is decimal 40000 for 200Hz  (41493 to also compensate for error in CLK; see above)
            out OCR1AH, tempH       ; load OCR1A high byte
            ldi tempL, low(40000)   ; low byte of upper limit to count upto is decimal 40000 for 200Hz  (41493 to also compensate for error in CLK; see above)
            out OCR1AL, tempL       ; load OCR1A low byte

            ldi tempH, 0x00         ; Timer 1 will count upwards from 0x0000
            out TCNT1H, tempH       ; load timer high byte FIRST since it is stored internally in a temporary location until the low byte is written
            ldi tempL, 0x00
            out TCNT1L, tempL       ; now that high byte is loaded, load timer low byte

            ldi temp, 0b00010000    ; clear timer 1 overflow flag OCFR1A1bit4 by writing a logic 1 to it as the datasheet says pg 113
            out TIFR, temp

            ldi temp, 0b00001001    ; WGM for CTC mode (i.e. counter counting from 0x0000 to OC1RA) set WGM12bit3=1 and WGM13bit4=0
            out TCCR1B, temp        ; Also, set CS12bit2, CS11bit1 and CS10bit0 to 001 starts counting


waittimer:
            in temp, TIFR
            sbrs temp, OCF1A        ; skip next instruction if OCF1A flag is set i.e. after the timer reaches OCR1A
            rjmp waittimer          ; loop while OCF1A flag is not set

            ldi temp, 0xff
            eor leds, temp
            out PORTB, leds

            ldi temp, 0b00010000    ; clear timer 1 overflow flag OCF1A (bit4 of TIFR) by writing a logic 1 to it as the datasheet says pg 113
            out TIFR, temp

            rjmp    waittimer
```