```
;***************************************************************
;*                                                             *
;* ADC writes to STK500 LEDs on PORTB with sampling rate 200MHz.  *
;*                                                             *
;***************************************************************

.include "m32def.inc"


;***** Global register variables
.def    temp2   =R22
.def    temp    =R21
.def    tempL   =R20
.def    tempH   =R19
.def    tempADCL=R18
.def    tempADCH=R17

reset:                                          ;Main program entry point on reset

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

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

        ldi     temp,0b00000000         ;disable pull up resistor on each input pin of PORTA
        out     PORTA,temp

        ldi     temp,0b00000000         ;set PORTA as INPUT
        out     DDRA,temp

        cbi PORTA, PA0                  ;disable pull up resistor of PA0 (this instruction is not really needed given above instructions)
        cbi DDRA, DDA0                  ;set PA0 as input (this instruction is not really needed given above instructions)

        ldi temp, 0b01100000            ;REFS1bit7 and REFS0bit6 set to 01 for AVCC internal voltage reference i.e. 5V. Set ADLARbit5=1 for 10 bit Left Adjusted Output.
        out     ADMUX, temp             ;Set MUX4bit5-MUX0bit0 to 00000 for ADC) single ended input selection

timerctc:

        ldi temp2, 0b01000000           ; WGM for CTC mode (i.e. counter counting from 0x0000 to OC1RA) set to zero WGM10bit0 and WGM11bit1
        out TCCR1A, temp2               ; set COM1A1bit7 and COM1A0bit6 to 01 to enable toggling of OC1A (i.e. PD5) when OCR1A value is reached

        ldi temp2, 0b00001000           ; WGM for CTC mode (i.e. counter counting from 0x0000 to OC1RA) set WGM12bit3=1 and WGM13bit4=0
        out TCCR1B, temp2               ; Also, set CS12bit2, CS11bit1 and CS10bit0 to zero to stop counting before loading TCNT1 and OCR1A

        ldi tempH, high(41493-1)        ; high byte of upper limit to count upto is decimal 40000=5000us x 8 (since clock is 8MHz) for 200Hz fs (41493 to also compensate for error in CLK; see above)
        ldi tempL, low(41493-1)         ; low byte of upper limit to count upto is decimal 40000=5000us x 8 (since clock is 8MHz) for 200Hz fs (41493 to also compensate for error in CLK; see above)
        out OCR1AH, tempH               ; load OCR1A high byte
        out OCR1AL, tempL               ; load OCR1A low byte

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

```
                out TCNT1L, tempL               ; now that high byte is loaded, load timer low byte

                ldi temp2, 0b00010000           ; clear timer 1 overflow flag OCFR1A1bit4 by writing a logic 1 to it as the datasheet says pg 113
                out TIFR, temp2

                ldi temp2, 0b00001001           ; WGM for CTC mode (i.e. counter counting from 0x0000 to OC1RA) set WGM12bit3=1 and WGM13bit4=0
                out TCCR1B, temp2               ; 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 waittimer2                 ; loop while OCF1A flag is not set

;============================== ADC Conversion ================================================================

startConversion:
                ldi temp, 0b11000000            ;ADENbit7=1 to enable ADC, ADSCbit6=1 to start conversion, ADPS2bit2-ADPS0=000 for prescaler set to 1
                out     ADCSRA, temp            ;Set MUX4bit5-MUX0bit0 to 00000 for ADC) single ended input selection

waitadc:
                sbic ADCSRA, ADSC               ;ADSC bit = 0 after the ADC conversion is complete
                rjmp waitadc                    ;loop until the ADCS bit = 0


                in      tempADCL, ADCL          ;Must read FIRST ADCL and THEN ADCH otherwise new conversion does not start
                in      tempADCH, ADCH


                ldi temp, 0xff                  ;turn 0s into 1s and 1s into 0s since STK500 LEDs turn on when PBx is zero
                eor temp, tempADCH              ;turn 0s into 1s and 1s into 0s since STK500 LEDs turn on when PBx is zero

                out     PORTB, temp

                rjmp startConversion

;=============================================================================================================

                ldi temp2, 0b00010000           ; clear timer 1 overflow flag OCFR1A1bit4 by writing a logic 1 to it as the datasheet says pg 113
                out TIFR, temp2

                rjmp waittimer
```