// PROJECT  :T1NormalOVFISR

// PURPOSE  :Timers, Interrupts, and Service Routines

// DEVICE   :Arduino

// AUTHOR   :C. D'Arcy

// DATE     :2019 02 13

// uC       :328p

// COURSE   :ICS4U

// STATUS   :Working

// REFERENCE:**Section 16.11**:  
// <https://mail.rsgc.on.ca/~cdarcy/Datasheets/ATmega328P.pdf>

// NOTES    :**AVR Timer Calculator**:

// <https://www.easycalculation.com/engineering/electrical/avr-timer-calculator.php>

#include "prescalers.h" //Arduino IDE **and** AS7

//volatile uint8\_t ovfCount;    //optional use

// Macro for Timer 1 Overflow Interrupt Service Routine with  
// predefined vector identifier

ISR(TIMER1\_OVF\_vect) {

 //  ovfCount++; //optional additional scaling

 //  if (!ovfCount)

 PORTB ^= (1 << PB5); //toggle UNO LED

}

int main() { //(KIS) avoid separate setup()&loop()

 DDRB |= (1 << PB5); //pin 13 set for output

 //ovfCount = 0; //optional

 cli(); //disable global interrupts

 TCCR1A = 0; //Normal Mode (start with simplest)

 TCCR1B = T1psNone; //no prescaler to start (full speed)

 TCNT1 = 0; //optional for this mode

 TIMSK1 = 1 << TOIE1; //enable Timer1 overflow interrupt

 sei(); //enable global interrupt system

 while (1); //enjoy the show; ISR responds to overflows

}

// FILE: prescalers.h

// RSGC ACES: ICS4U

// Prescale constants for ATmega328p Timers

#define T0Stopped 0b00000000    // Timer0 stopped

#define T0psNone  0b00000001    // T0:2^24/2^8  (no prescale)> 2^? ovf/s = ? Hz

#define T0ps8     0b00000010    // T0:2^24/2^3/2^8 (prescale)> 2^? ovf/s = ? Hz

#define T0ps64    0b00000011    // T0:2^24/2^6/2^8 (prescale)> 2^? ovf/s = ? Hz

#define T0ps256   0b00000100    // T0:2^24/2^8/2^8 (prescale)> 2^? ovf/s = ? Hz

#define T0ps1024  0b00000101    // T0:2^24/2^10/2^8(prescale)> 2^? ovf/s = ? Hz

#define T1Stopped 0b00000000    // Timer1 stopped

#define T1psNone  0b00000001    // T1:2^24/2^16  (no prescale)> 2^8 ovf/s > 128Hz

#define T1ps8     0b00000010    // T1:2^24/2^3/2^16 (prescale)> 2^5 ovf/s > 16Hz

#define T1ps64    0b00000011    // T1:2^24/2^6/2^16 (prescale)> 2^2 ovf/s > 2Hz

#define T1ps256   0b00000100    // T1:2^24/2^8/2^16 (prescale)> 1 ovf/s   > 0.5Hz

#define T1ps1024  0b00000101    // T1:2^24/2^10/2^16(prescale)> 0.25 ovf/s> 0.125Hz

#define T2Stopped 0b00000000    // Timer2 stopped

#define T2psNone  0b00000001    // T2:2^24/2^8  (no prescale)> 2^? ovf/s > ? Hz

#define T2ps8     0b00000010    // T2:2^24/2^3/2^8 (prescale)> 2^? ovf/s = ? Hz

#define T2ps32    0b00000011    // T2:2^24/2^5/2^8 (prescale)> 2^? ovf/s = ? Hz

#define T2ps64    0b00000100    // T2:2^24/2^6/2^8 (prescale)> 2^? ovf/s = ? Hz

#define T2ps128   0b00000101    // T2:2^24/2^7/2^8(prescale)> 2^? ovf/s = ? Hz

#define T2ps256   0b00000110    // T2:2^24/2^8/2^8(prescale)> 2^? ovf/s = ? Hz

#define T2ps1024  0b00000111    // T2:2^24/2^10/2^8(prescale)> 2^? ovf/s = ? Hz