// PROJECT  :analogWrite

// PURPOSE  :Register-Level implementation of analogWrite

// COURSE   :ICS4U-E

// AUTHOR   :C. D'Arcy

// DATE     :2024 12 13

// MCU      :328P

// STATUS   :Working

// REFERENCE:Secrets of Arduino PWM:

// :<http://www.righto.com/2009/07/secrets-of-arduino-pwm.html>

// REFERENCE:ATmega328P Datasheet: Timer1 Registers:

//  :[http://darcy.rsgc.on.ca/ACES/Datasheets/ATMEGA328P.pdf#page=131](http://darcy.rsgc.on.ca/ACES/Datasheets/ATMEGA328P.pdf%23page%3D131)

void setup() {

// High-Level PWM: 64 is 25% Duty Cycle (1.25 V?)

  analogWrite(9, 64);

// Register-Level PWM: 64 is 25% Duty Cycle (1.25 V?)

//analogWriteT1A(10);

}

void analogWriteT1A(uint8\_t val) {

// must explicity declare PWM digital pin for output

  DDRB |= (1 << PB1);  // aka pinMode(9, OUTPUT);

// Fast PWM Mode 5 is the closest fit with analogWrite() function

// Clear OC1A (Pin 9) on Compare Match, Set OC1A at BOTTOM

// (non-inverting mode)

  TCCR1A = (1 << COM1A1) | (1 << WGM12) | (1 << WGM10);

// Alternative to previous statement

//TCCR1A = \_BV(COM1A1) | \_BV(WGM12) | \_BV(WGM10);

  TCCR1B |= (1 << CS10);      // clk/1 (No prescaling) ~64 KHz.

  OCR1A = val;

}

void loop() {} // Nothing to do. Everything done in hardware