**// PROJECT  :RotaryEncoderPinChangeInt**

// PURPOSE  :ACES interpretation of Electronoobs tutorial of Pin Change Interrupts

// COURSE   :ICS4U

// AUTHOR   :Electronoobs/C. D'Arcy

// DATE     :2023 03 06

// MCU      :328P

// STATUS   :Working

// REFERENCE:<https://electronoobs.com/eng_arduino_tut132.php>

#define PCIEn  1<<PCIE2         //Use Port D as Pin Change Bank

#define PCPinA 1<<PD4           //Encoder pin A mask

#define PCPinB 1<<PD5           //Encoder pin B mask

#define RUNLENGTH 20            //Plotter visualization aid

uint8\_t pinAState;              //monitors the active pin A state

uint8\_t pinBState;              //monitors the active pin A state

volatile uint8\_t triggered = 0; //signals ISR activity

void setup() {

  **Serial**.begin(9600);           //support for Monitor/Plotter

  while (!**Serial**);              //wait for it...

  PCICR |= PCIEn;               //identify Pin Change Bank

  PCMSK2 |= PCPinA | PCPinB;    //identify Pin Change Interrupt trigger pins

  pinAState = PIND & PCPinA;    //acquire the starting pin A state

  pinBState = PIND & PCPinB;    //acquire the starting pin B state

}

ISR (PCINT2\_vect) {

  triggered = 1;

}

void loop() {

  if (triggered) {              //ISR was triggered

    triggered = 0;              //prepare for next one

    pinAState = PIND & PCPinA;  //acquire the current pin A state

    pinBState = PIND & PCPinB;  //acquire the current pin B state

    for (uint8\_t i = 0; i < RUNLENGTH; i++) {

      **Serial**.print(pinAState ? 4 : 3);

      **Serial**.print(',');

      **Serial**.println(pinBState ? 1 : 0);

    }

  }

}

