

// PROJECT  :MyPinChangeInterruptTest

// PURPOSE  :High and Register-Level presentation of Pin Change Interrupts.

// USE      :Hour:Minute buttons for CharlieClock (A5(PCINT13):A4(PCINT12))

// AUTHOR   :C. D'Arcy

// DATE     :2025 09 15

// uC       :328p

// COURSE   :ICS4U-E

// STATUS   :Working

// REFERENCE:

//<https://thewanderingengineer.com/2014/08/11/arduino-pin-change-interrupts/>

//<https://youtu.be/wIcC8-g9Lnw?t=1191>

//<http://darcy.rsgc.on.ca/ACES/TEI4M/Assembly/images/InterruptVectorTable.png>

//<http://darcy.rsgc.on.ca/ACES/TEI4M/CharlieClock/images/PCIntPrototype.png>

#define LEDHour PB4    //digital pin 12

#define LEDMinute PB1  //digital pin 9

#define DURATION 200   //arbitrary

volatile boolean triggered = false;

void setup() {

// START OF Register-Level preparation of the LED indicators (confirmation)

  DDRB = B00010010;     //DDRB |= 1 << LEDHour | 1 << LEDMinute; or pinMode(LEDHour, OUTPUT); and pinMode(LEDMinute,OUTPUT);

  PORTB &= ~B00010010;  //PORTB &= ~(1 << LEDHour | 1 << LEDMinute); or digitalWrite(LEDHour, LOW); and digitalWrite(LEDMinute, LOW)

  //Turn on LED Indicator pins 12 and 9 to simply test LEDs

  PORTB |= B00010010;  //PORTB |= 1<< LEDHour | 1<<LEDMinute; digitalWrite(12, HIGH) and digitalWrite(9,HIGH);

  delay(DURATION);

  //Turn off LED Indicator pins 12 and 9

  PORTB &= ~B00010010;  //PORTB &= ~(1<< LEDHour | 1<<LEDMinute); digitalWrite(12, LOW) and digitalWrite(9,LOW);

// END OF Register-Level preparation of the LED indicators (confirmation)

//Configure Pin Change interrupts on A5 (Hour) and A4 (Minute)

  // 0. Disable interrupts while configuring (Clears Global flag in SREG)

  cli();

  //1. Enable Pin Change Interrupts on PORTC (digital pins 14-19(aka A0-A5))

  PCICR = B00000010;   //PCICR = 1 << PCIE1;

  //2. Activate specific port pins. In this case A5:A4 (aka PCINT13:PCINT12)

  PCMSK1 = B00110000;  //PCMSK1 |= 1 << PCINT13 | 1 << PCINT12;

  //3. Set internal Pullups for button inputs (to monitor falling edge)...

  PORTC = B00110000;   //PORTC |= 1 << PC5 | 1 << PC4; or pinMode(19,INPUT\_PULLUP);

  //4. Reenable all interrupts to go live...

  sei();

}

void loop() {

  // Was a button pressed? (CHANGE from HIGH to LOW)...

  if (triggered) {

    triggered = false; // Prepare for the next one

    // DECODE which pin within the group changed to LOW...

    // Was it the HOUR button (A5) that was pressed?...

    if ((~PINC & 1 << PCINT13)) {  // bitClear(PINC,PCINT13) or !digitalRead(A5)

      PORTB |= 1 << LEDHour;       // digitalWrite(12, HIGH);

      delay(DURATION);

      PORTB &= ~(1 << LEDHour);    // digitalWrite(12, LOW);

    } else

    // Was it the MINUTE button (A4) that was pressed?...

if (~PINC & 1 << PCINT12) {   //  or bitClear(PINC,PCINT12) or !digitalRead(A4)

      PORTB |= 1 << LEDMinute;     // digitalWrite(9, HIGH);

      delay(DURATION);

      PORTB &= ~(1 << LEDMinute);  // digitalWrite(9, LOW);

    }

  }

}

// Use separate ISRs to handle each group

//PCINT0 (PORTB: digital pins 8-13)

ISR(PCINT0\_vect) {}  // handle pin change interrupt for D8 to D13 here

//PCINT1 (PORTC: digital pins 14-19(aka A0-A5) )

ISR(PCINT1\_vect)  // handle pin change interrupt for A0 to A5 here

{

  triggered = true;  // get out ASAP

}

//PCINT2 (PORTD: digital pins 0-7)

ISR(PCINT2\_vect) {}  // handle pin change interrupt for D0 to D7 here