// PROJECT  :BarsorDotsIncreasing

// PURPOSE  :Register-Level response to Nano Coding Companion Exercise

// COURSE   :ICS4U

// AUTHOR   :C. D'Arcy

// DATE     :2023 01 25. Updated 2023 02 13.

// MCU      :328P

// STATUS   :Working. Confirmed. Possible Memory Leak…

// REF:http://darcy.rsgc.on.ca/ACES/TEI3M/ArduinoCode/images/Animations/Bar0\_15.gif

// :http://darcy.rsgc.on.ca/ACES/TEI3M/ArduinoCode/images/Animations/Dot0\_15.gif

//    :https://github.com/arduino/ArduinoCore-avr/blob/master/cores/arduino/wiring\_digital.c

// NOTE     :Introduces C concepts of addresses and pointers (aka references)

#define BARS        //choose between BARS or ...

//#define DOTS        // ...DOTS

struct DIOPin {     //define data structure to hold GPIO pin/register details

 uint8\_t bit;

 uint8\_t pin;

 uint8\_t ddr;

 uint8\_t port;

};

// define an array of the 328P/NCC pins

const DIOPin NCCPins[] = {

 {PD0, &PIND, &DDRD, &PORTD},      //pin0: PIND, DDRD, PORTD

 {PD1, &PIND, &DDRD, &PORTD},

 {PD2, &PIND, &DDRD, &PORTD},

 {PD3, &PIND, &DDRD, &PORTD},

 {PD4, &PIND, &DDRD, &PORTD},

 {PD5, &PIND, &DDRD, &PORTD},

 {PD6, &PIND, &DDRD, &PORTD},

 {PD7, &PIND, &DDRD, &PORTD},      //pin 7: PIND, DDRD, PORTD

 {PB0, &PINB, &DDRB, &PORTB},      //pin 8: PINB, DDRB, PORTB

 {PB1, &PINB, &DDRB, &PORTB},

 {PB2, &PINB, &DDRB, &PORTB},

 {PB3, &PINB, &DDRB, &PORTB},

 {PB4, &PINB, &DDRB, &PORTB},

 {PB5, &PINB, &DDRB, &PORTB},      //pin 13: PINB, DDRB, PORTB

 {PC0, &PINC, &DDRC, &PORTC},      //pin 14(A0): PINC, DDRC,PORTC

 {PC1, &PINC, &DDRC, &PORTC}       //pin 15(A1): PINC, DDRC,PORTC

};

uint8\_t numPins = sizeof(NCCPins) / sizeof(DIOPin);

#define PAUSE 50                //speed...

//This declaration may result in a memory leak...

// ...Not sure why it dies after a minute or so

// Found that if I limit the number of LEDS to 10 it continues without issue

volatile uint8\_t \*reg;          //pointer to (holds address of) a GPIO register

uint8\_t i = 0;                  //current pin (index of the array element)

void setup() {

 // Eventually, I came up with the line below that actually worked

 DDRD = 0xFF;   // Why does this work?? Who knows? even though I will be using them as inputs...

 //set the direction of all NCC pins to output

 for (i = 0; i < numPins; i++) {

   reg = NCCPins[i].ddr;           //obtain the correct DDR

   \*reg |= 1 << NCCPins[i].bit;    //set I/O direction to Output

 }

}

void loop() {

 //Turn the next pin ON...

 reg = NCCPins[i].port;            //obtain the correct PORT

 \*reg |= 1 << NCCPins[i].bit;      //provide source voltage

 delay(PAUSE);

#ifdef DOTS //Turn the pin OFF...

 //  reg = NCCPins[i].port;

 \*reg &= ~(1 << NCCPins[i].bit);   //provide sink (ground)

#endif

 //prepare for next pin...which style do you prefer?...

 i = i == numPins ? 0 : i + 1;     //  numPins ? i=0 : i++;

#ifdef BARS            //For BARS, must turn off ALL pins at the end

 if (!i) {            // Are we at the end?  If so, turn them all off

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

     reg = NCCPins[i].port;

     \*reg &= ~(1 << NCCPins[i].bit);

   }

   delay(1000);      //hold for a second to confirm all LEDS are turned off.

 }

#endif

}