// PROJECT  :JoystickMatrixMadeEZ

// PURPOSE  :Introduction to the Thumb Joystick for ICS3U

// COURSE   :ICS3U-E

// AUTHOR   :C. D'Arcy

// DATE     :2025 02 04

// MCU      :328P

// STATUS   :Working

// REFERENCE:http://darcy.rsgc.on.ca/ACES/TEI3M/Fritzing/JoystickMatrixMadeEZ.gif

//Input: Thumb Joystick

#define TJGND A0              //

#define TJSEL A1              //

#define TJHORZ A2             //

#define TJVERT A3             //

#define TJVCC A4              //

//Output: Matrix Made EZ

#define SHIFT\_DATA 8          //

#define SHIFT\_CLOCK 7         //

#define SHIFT\_LATCH 6         //

#define PIN\_DIM 3             //

#define DEBUG false           //

uint8\_t matrix[8];            //buffered image of the MatrixMadeEZ display

volatile boolean triggered = false;   //holds the ISR state

void setup() {

  if (DEBUG) {                //conditional support for the Serial Monitor

    Serial.begin(9600);

    while (!Serial);

  }

  //Input

  pinMode(TJGND, OUTPUT);

  pinMode(TJVCC, OUTPUT);

  digitalWrite(TJGND, LOW);

  digitalWrite(TJVCC, HIGH);

  //Output

  pinMode(TJSEL, INPUT\_PULLUP);

  pinMode(SHIFT\_DATA, OUTPUT);

  pinMode(SHIFT\_CLOCK, OUTPUT);

  pinMode(SHIFT\_LATCH, OUTPUT);

  pinMode(PIN\_DIM, OUTPUT);

  digitalWrite(PIN\_DIM, LOW);

  // Use of the external interrupt to trigger the clearing of the display

  attachInterrupt(digitalPinToInterrupt(2), ISRClear, FALLING);

}

//Interrupt Service Routine for the Joystick's Button press

void ISRClear() {

  triggered = true;

}

void loop() {

  if (triggered) {        //clear the display if requested

    triggered = false;    //prepare for the next interrupt

    for (uint8\_t r = 0; r < 8; r++)   //cleat the display

      matrix[r] = 0;

    delay(300);           //pause to allow to settle

  }

 //map joystick reading to a viable column number

  uint8\_t column = analogRead(TJHORZ) >> 7;

 //map joystick reading to a viable row number

 uint8\_t row = analogRead(TJVERT) >> 7;

  if (DEBUG) {

    Serial.print(column);

    Serial.print('\t');

    Serial.println(row);

  }

  matrix[row] |= 1 << column;               //Set the column  (or equals)

  for (uint8\_t r = 0; r < 8; r++) {         //row scanning update of display

    digitalWrite(SHIFT\_LATCH, LOW);

    shiftOut(SHIFT\_DATA, SHIFT\_CLOCK, LSBFIRST, matrix[r]);

    shiftOut(SHIFT\_DATA, SHIFT\_CLOCK, MSBFIRST, 1 << (7 - r));

    digitalWrite(SHIFT\_LATCH, HIGH);

  }

}