// **PROJECT  :MatrixMadeEZInversion**

// PURPOSE  :Demonstrates simple TimerOne interrupt-driven inversion animation

// FOR      :2019/2020 ICS3U

// AUTHOR   :C. D'Arcy

// DATE     :2019 12 05

// uC       :328P

// STATUS   :Working

// REFERENCE:<https://cdn-shop.adafruit.com/datasheets/1079datasheet.pdf>

**#include <TimerOne.h>       //requires use of pins 9 and 10**

#define SPEED 5        //frame flipping speed (in uS)

#define DMLEVEL 200         ///G on Power TPIC:  bright:0, dim:250ish

uint8\_t dimmer = 3;         //PWM (or external voltage divider, eg pot, LDR)

uint8\_t data = 11;          //595 Shift Register Support

uint8\_t latch = 12;         //          "

uint8\_t clk = 13;           //          "

//<http://darcy.rsgc.on.ca/ACES/LEDMatrixUtility.html>

**volatile** uint8\_t image[] = { 255, 153, 255, 231, 231, 189, 195, 255};

uint8\_t row = 0;

void setup() {

 pinMode(dimmer, OUTPUT);        //declare pin states

 pinMode(data, OUTPUT);          //      "

 pinMode(clk, OUTPUT);           //      "

 pinMode(latch, OUTPUT);         //      "

 analogWrite(dimmer, DMLEVEL);   //brightness control on the /G pin of the TPIC

 Timer1.initialize(SPEED);       //instantiate timer and interval

 Timer1.attachInterrupt(invert); //identify the Interrupt Service Routine

}

//Timer ISR (Interrupt Service Routine)

//Note: Keep as short as possible!

void invert() {

 for (uint8\_t r = 0; r < 8; r++)

   image[r] = ~image[r];         //invert (complement) the column data

}

void loop() {

 shiftOut(data, clk, MSBFIRST, image[row]);    //Note: **MSBFIRST!**

 shiftOut(data, clk, LSBFIRST, 1 << row);      //LSBFIRST

 digitalWrite(latch, LOW);                     //create a rising edge on latch

 digitalWrite(latch, HIGH);                    // to store latched data on Q pins

 row = row == 7 ? 0 : row + 1;           //max efficiency == No division!

}