// PROJECT  :SPIvsShiftOutV2

// PURPOSE  :Contrast software shiftOut and hardware SPI

// COURSE   :ICS3U

// AUTHOR   :C. D'Arcy

// DATE     :2020 03 24

// MCU      :328P

// STATUS   :Working

// REFERENCE:Hardware prototype to be as depicted on the right

// :Right click on image and Open hyperlink to expand

#define HARDWARE\_SHIFT false

#if HARDWARE\_SHIFT

#include <**SPI**.h>

#endif

#define DURATION  500

uint8\_t i = 1;

void setup() {

  pinMode(SS, OUTPUT);

#if HARDWARE\_SHIFT

  **SPI**.begin();

#else

  pinMode(SCK, OUTPUT);

  pinMode(MISO, OUTPUT);

  pinMode(MOSI, OUTPUT);

#endif

}

void loop() {

  digitalWrite(SS, LOW);

#if HARDWARE\_SHIFT

  **SPI**.transfer(i);

#else

  shiftOut(MOSI, SCK, MSBFIRST, i);

#endif

  digitalWrite(SS, HIGH);

  delay(DURATION);

  i = i ? i << 1 : 1;

}

// PROJECT  :MCP4231

// PURPOSE  :Demonstrates SPI Library manipulation of MCP4231 Digital Pot

// DEVICE   :Arduino + MCP4231 Digital Pot + DMM set to Voltage

// AUTHOR   :C. D'Arcy

// DATE     :2019 02 06

// uC       :328p

// COURSE   :ICS3U/ICS4U

// STATUS   :Working

// REFERENCE:<https://mail.rsgc.on.ca/~cdarcy/Datasheets/MCP4231.pdf>

// REFERENCE:<https://www.arduino.cc/en/Reference/SPISettings>

// NOTES    :MCP4231 Datasheet: up to 10MHz SCK

//          :Connect a DMM to the P0W Pin to watch potential (DCV) change

#include <**SPI**.h>            // include the SPI library of course

SPISettings mySettings(10000000, MSBFIRST, SPI\_MODE0);

//------------MCP4231 Parameters------------------------------

#define WRITE B00000000     //Set Command (MOSI)

#define INCR  B00000100     //Increment Command

#define DECR  B00001000     //Decrement Command

#define READ  B00001100     //Get Command (MISO)

#define REG0  B00000000     //Register 0 Write command

#define REG1  B00010000     //Register 1 Write command

#define LIMIT 127           //Maximum value for (7-bit) MCP4231

//------------Code Variables----------------------------------

uint8\_t data;               //value written to the MCP4231

uint8\_t res;                //value read back from the MCP4231

int8\_t  delta = -1;         //amount to change the value written (up/down)

void setup() {

  **Serial**.begin(9600);       //Serial monitoring useful

  **SPI**.begin();              //Initialize SPI (sets SCK, MISO & MOSI for output)

  **SPI**.beginTransaction(mySettings);     //(optional) tailoring of SPI session settings

  pinMode(SS, OUTPUT);      //set the Slave Select pin (to /CS pin of MCP4231) as an output

}

void loop() {

  SPIWrite(SS, REG0 | WRITE, data);       //sets and gets a new value

  **Serial**.print("SET:" + String(data));    //display the value written...

  res = SPIRead(SS, REG0 | READ);         //let's read it back to confirm...

  **Serial**.println("\tGET:" + String(res)); //display it for confirmation

  delay(64);                              //hmmmm...not to good...

  delta = (data == 0 || data == LIMIT) ? -delta : delta;  // change direction?

  data += delta;                          //update value to be written

}

void SPIWrite(uint8\_t slave, uint8\_t command, uint8\_t data) {

  digitalWrite(slave, LOW);   //Set the provided SS pin low

  **SPI**.transfer(command);      //Choose the register to write to

  **SPI**.transfer(data);         //Set the LED level [0,127]

  digitalWrite(slave, HIGH);  //Set the provided SS pin high again

}

uint8\_t SPIRead(uint8\_t slave, uint8\_t command){

  digitalWrite(slave, LOW);       //Set the provided SS pin low

  **SPI**.transfer(command);          //Choose the register to write to

  uint8\_t res = **SPI**.transfer(0);  //Send a dummy value to force a return

  digitalWrite(slave, HIGH);      //Set the provided SS pin high again

  return res;                     //Send the vlaue back to the caller

}

