// PROJECT  :AT28C16Write

// PURPOSE  :Writes data to the AT28C16 (2Kx8) EEPROM IC

// COURSE   :ICS4U

// AUTHOR   :B. Eater. adapted for ACES CHUMP use by C. D'Arcy

// DATE     :2019 11 13-16

// MCU      :Nano/328

// STATUS   :Working

// REFERENCE:B. Eater Github...

// :https://github.com/beneater/eeprom-programmer/blob/master/eeprom-

// programmer/eeprom-programmer.ino

// REFERENCE:B. Eater Videos

//          1. Using an EEPROM to replace combinational logic

//          https://www.youtube.com/watch?v=BA12Z7gQ4P0&feature=emb\_logo

//          2. Build an Arduino EEPROM programmer

//          https://www.youtube.com/watch?v=K88pgWhEb1M&feature=emb\_logo

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

// CHUMP Basic Program Example

//  0000: 10000010  READ    2   ;addr<-2

//  0001: 00010000  LOAD    IT  ;accum<-mem[addr]

//  0002: 00100001  ADD     1   ;accum++

//  0003: 01100010  STORETO 2   ;mem[2]<-accum

//  0004: 10100000  GOTO    0   ;pc<-0000

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

// populate array below with machine instructions above

byte code [] = {

0b????????,

0b????????,

0b????????,

0b????????,

0b????????,

};

#define PROG\_SIZE sizeof(code)

#define EEPROM\_D0 5

#define EEPROM\_D7 12

#define EEPROM\_WE 13

#define EEPROM\_A0 ??    //  |PC0

#define EEPROM\_A1 ??    //  |PC1

#define EEPROM\_A2 ??    //  |PC2

#define EEPROM\_A3 ??    //  |PC3

#define EEPROM\_OE ??    //  |PC4

// Tie /CE to ground as permanently enabled

void setup() {

 pinMode(EEPROM\_A0, OUTPUT);

 pinMode(EEPROM\_A1, OUTPUT);

 pinMode(EEPROM\_A2, OUTPUT);

 pinMode(EEPROM\_A3, OUTPUT);

 digitalWrite(EEPROM\_WE, HIGH);

 pinMode(EEPROM\_WE, OUTPUT);

 digitalWrite(EEPROM\_OE, HIGH);

 pinMode(EEPROM\_OE, OUTPUT);

**Serial**.begin(9600);

//Display the code to be flashed...

**Serial**.println("Here's the data to be flashed to EEPROM...");

 for (int address = 0; address < PROG\_SIZE; address++) {

**Serial**.println(code[address], HEX);

 }

 // Write the code to EEPROM...

**Serial**.println("Writing " + String(PROG\_SIZE) + " bytes of code to EEPROM...");

 for (int address = 0; address < PROG\_SIZE; address++) {

   writeEEPROM(address, code[address]);

 }

**Serial**.println("Done");

 // Confirm the write by reading and echoing the code to the Serial Monitor...

**Serial**.println("Reading EEPROM");

 printContents();

}

// Read the contents of the EEPROM and print them to the serial monitor.

void printContents() {

 for (int address = 0; address < PROG\_SIZE; address++) {

**Serial**.println(readEEPROM(address),HEX);

 }

}

// Output the address bits and outputEnable signal using shift registers.

void setAddress(int address) {

 for (int pin = EEPROM\_A3; pin >= EEPROM\_A0; pin--) {

   digitalWrite(pin, address & 0x08);

   address <<= 1;    //destructive

 }

}

/\*

  Read a byte from the EEPROM at the specified address.

\*/

byte readEEPROM(int address) {

 digitalWrite(EEPROM\_WE, HIGH);

 for (int pin = EEPROM\_D7; pin >= EEPROM\_D0; pin--) {

   pinMode(pin, INPUT);

 }

 byte value = 0;

 setAddress(address);

 digitalWrite(EEPROM\_OE, LOW);

 delayMicroseconds(1);

 digitalWrite(EEPROM\_OE, HIGH);

 for (int pin = EEPROM\_D7; pin >= EEPROM\_D0; pin--) {

**Serial**.print(digitalRead(pin));

???????????????????????

 }

**Serial**.print(" ");

 return value;

}

//  Write a byte to the EEPROM at the specified address.

void writeEEPROM(int address, byte data) {

 digitalWrite(EEPROM\_OE, LOW);

 digitalWrite(EEPROM\_WE, HIGH);

**Serial**.println("[" + String(address) + "]" + String(code[address], HEX));

 //Set the address

 setAddress(address);

 digitalWrite(EEPROM\_OE,HIGH);

 //prepare to write the data...

 for (int pin = EEPROM\_D0; pin <= EEPROM\_D7; pin++) {

   pinMode(pin, ??????);

 }

 //write the data...

 for (int pin = EEPROM\_D7; pin >= EEPROM\_D0; pin--) {

   digitalWrite(pin, data & 0x80);

   data ??????;     //destructive....

 }

 digitalWrite(EEPROM\_WE, LOW);

 delayMicroseconds(1);

 digitalWrite(EEPROM\_WE, HIGH);

 delay(10);

}

void loop() {

 // nothing left to do...

}

