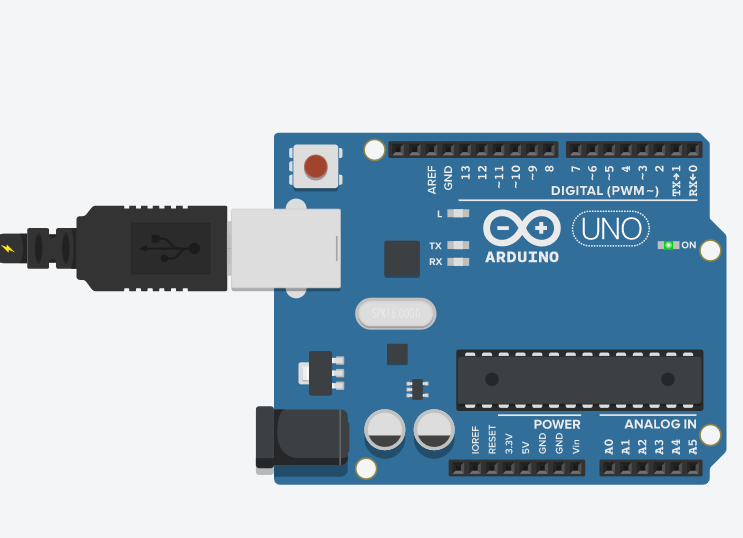
**LAB REPORT 3**

Ben Giftakis

8/13/20

Screenshot + Components:



* Arduino – single board computer

Summary:

* I created the code to first initialize a series of chars in memory addresses 0-7 with a checksum output going in memory address 8, then made a function that creates a checksum by XOR’ing the memory addresses 0-7 sequentially, then taking its compliment.

Results:

initial check

000 h

001 R

002 I

003 P

004 L

005 E

006 Y

007 O

008

EEPROM has been reinitialized

000 T

001 R

002 I

003 P

004 L

005 E

006 X

007 9

008 

Conclusions:

* Learned syntax for bitwise operations
* Found out that the serial monitor doesn’t always show the checksum char
* It was quite helpful to do my own initializing in the beginning to show that “corruption” that took place. It made things a little easier to visualize.

Code:

#include <EEPROM.h>

void setup()

{

  Serial.begin(9600);

    EEPROM.write(0,'T');//0

    EEPROM.write(1,'R');//1

    EEPROM.write(2,'I');//2

    EEPROM.write(3,'P');//3

    EEPROM.write(4,'L');//4

    EEPROM.write(5,'E');//5

    EEPROM.write(6,'Y');//6

    EEPROM.write(7,'O');//7

  // for test purposes only

  EEPROM.write(0, 'h');  // overwrite something to simulate data corruption

  // print initial state of EEPROM

  Serial.println("initial check");

  printEEPROM();

  // if checksum is not OK, reinitialize EEPROM to default values

  if((checkSum() & 0xff) != EEPROM.read(0x08)) {

    initialize();

    Serial.println("EEPROM has been reinitialized");

  }

  else {

    Serial.println("EEPROM checksum is OK");

  }

  printEEPROM();

}

void loop() {}

void printEEPROM() {

  for(int address = 0; address < 0x09; address++) {

    // read a byte from the current address of the EEPROM

    char value = EEPROM.read(address);

    if (address < 0x10)

      Serial.print("00");

    else if (address < 0x100)

      Serial.print("0");

    Serial.print(address, HEX);

    Serial.print("\t");

    Serial.print(value);

    Serial.println();

  }

  Serial.println();

}

void initialize() {

    //put some initial values in addresses 0x0 through 0x7

    EEPROM.write(0,'T');//0

    EEPROM.write(1,'R');//1

    EEPROM.write(2,'I');//2

    EEPROM.write(3,'P');//3

    EEPROM.write(4,'L');//4

    EEPROM.write(5,'E');//5

    EEPROM.write(6,'X');//6

    EEPROM.write(7,'9');//7

    EEPROM.write(8, checkSum()); //recalculate checksum and store it at 0x8

}

char checkSum() {

  char chksm = EEPROM.read(0);

  for (int i = 1; i<8;i++){

        chksm = chksm ^ EEPROM.read(i);

    }

  chksm = ~chksm;

  return chksm; //just a default value so the code compiles

}