

LAB REPORT 03

Name: Giancarlos Marte

Date: 3/4/21

Screenshot + components:

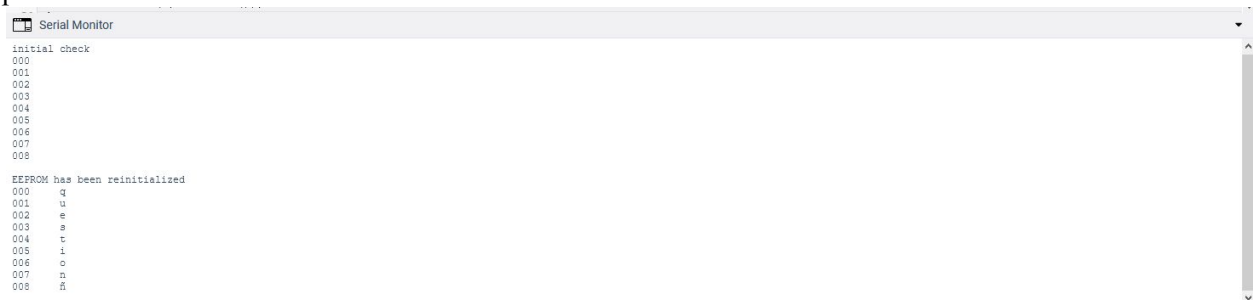
- ☐ No new components, just the arduino.

Summary:

In this lab I experimented with the EEPROM memory. Specifically, I stored a string into addresses 0 to 7 in the EEPROM of the Arduino using the given functions from the EEPROM library. I then implemented a checksum() algorithm by going through addresses 0 to 7, xor each value cumulatively and then return the complement of that. The char that is returned by the function is then used to make sure if the data was not corrupted. The purpose of this was to show how configuration data is stored and used in the real world.

Results:

part 1:



```
Serial Monitor
initial check
000
001
002
003
004
005
006
007
008

EEPROM has been reinitialized
000 q
001 u
002 e
003 a
004 t
005 i
006 o
007 n
008 n
```

part 2:

The character 'x' appears in address 7. Converting twobytes into hex you get 3778. In ASCII this becomes 7x. Since arduino is little endian, the least significant byte ('x') goes before the most significant byte ('7').

Conclusions:

2 things I learned:

1. EEPROM is usually used for configuration data in real world devices.
2. Algorithms like checksum() exist to validate the data in EEPROM, which makes your code less likely to run into errors.

Mistakes:

I had a difficult time understanding what was supposed to be printed in the serial monitor. Everytime I ran the code it would reinitialize everything. I thought after the first time running it, the serial monitor would show that the EEPROM checksum() was okay, but it never did. I don't know why this kept happening because the methods I implemented seemed correct.

Code:

```
#include <EEPROM.h>

void setup()
{
  Serial.begin(9600);
  // 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
  //Example: EEPROM.write(0, 'B'); //store character B at address 0x0
  char str[] = "question";
```

```

    for (int i = 0; i < 8; i++) {
        EEPROM.write(i, str[i]);
    }

    int twobytes = 14200; // For part 2

    // For part 2
    EEPROM.write(7, twobytes);

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

char checksum() {
    //implement a checksum algorithm
    char x = EEPROM.read(0);
    for (int address = 1; address < 0x8; address++) {
        x = x ^ EEPROM.read(address);
    }
    return ~x; //just a default value so the code compiles
}

```

Rubric:

Each lab is graded out of 10. Labs are due at midnight a week after they are assigned. Labs turned in late receive a max of 7 points:

Item	Points worth
Code correctness	3
Submission form correct	3
Report contains accurate information	2
Some effort put into report*	2

*No answer is too short to properly address the lab report section and I can tell you tried at least just a little.