EEPROM and LCD

Performance Check

_____ (40) EEPROM Write and Read functions (procedure 3 and 4)

_____ (50) Read from EEPROM, save UART message to EEPROM, then display in LCD upon a push button press or startup (7.d)

Objective:

The purpose of this lab is to use a 2x16 character LCD to display text, read and write to EEPROM, and display message sent from serial port to LCD.
Reference: For reference use the data sheet for the **Atmega 2560** controller that was supplied in Bb.

Read before writing your code :

1. Read through this entire lab before you attempt any of the prelab or lab procedures.
2. Complete your LCD and USART lab.
3. Wired up the LCD display, send a message from Coolterm / Serial Terminal. This message should show in the LCD display. This is from your previous lab.
4. Debugger will not work for this lab

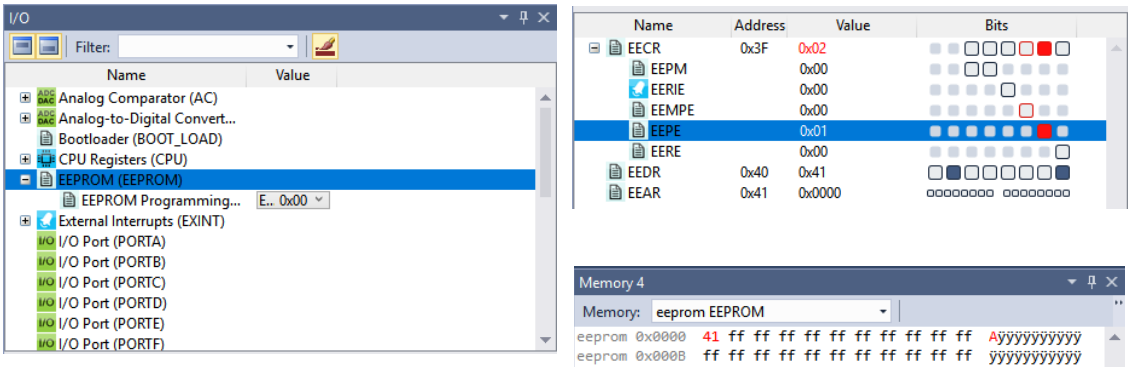
Lab Procedures: EEPROM

1. Set the LCD setup aside, do not dissemble. You will need it later during the lab.
2. **The following code should be placed above the while (1) loop.**
3. Write data to EEPROM. We will begin by writing a character to EEPROM at address 0x0000
    a. Start a new project with a primary module called main and a secondary module called EEPROM
    b. Write a function in the EEPROM module called EEPROM_write_one_char. This function should have the following function prototype.
    void EEPROM_write_one_char(uint16_t uiAddress, uint8_t ucData);
    c. EEPROM_write_one_char function should have the same code as that in p.25 of the datasheet. C Code example of EEprom in section 8.3.1
    d. In the main function, write a character “A” to address starting at 0x0000.
    e. Verify that you can write to EEPROM by using the simulator. Timing is very important in the EEPROM operation, Do Optimize the Compiler in this exercise. **Optimization level should remain at Optimize (-O1)**
        i. You should have the following simulator windows:

1. EEPROM memory
2. I/O window → EEPROM

ii. Run the simulator, it will seem to be frozen because there is only one line of code. After running the simulator, press CTRL+F5 or the ⏸ button.

iii. The following screen shots show you where the EEPROM window is located



iv. Notice the red number 41 in eeprom memory window. This shows that the character 'A' is stored in address 0x0000

f. Once you have accomplish above, you can start writing to more EEPROM locations. Increase the address uiAddress to the next location and write another character. Then verify that it works by downloading the memory again. To verify, use **Breakpoint** to simulation, instead of Step Into. Timing is critical in this operation, using Breakpoint will ensure proper timing for these operations.

g. Make a message array called Message, it should contain this "Type something"

h. Use a while loop to save the information inside the array Message to EEPROM. Your function should have the following function prototype (ucData points to the address of Message:

```
void EEPROM_write_string(uint16_t uiAddress, char *ucData);
```

Watch video online to see this operation.

4. **Read from EEPROM**

a. Add in the EEPROM module:

i. An EEPROM read function inside this module. Follow the code as illustrated in the datasheet in Chapter 8, this code read one character at a specified address

ii. Write another function that read several characters (string), until the EEPROM data equals to 0xFF. The value will be stored in an array (EEPROM_buf), which address is pointed by a pointer EEPROM_buf_ptr. Its function prototype :

```
void EEPROM_read_string(uint16_t uiAddress, char *EEPROM_buf_ptr);
```

b. Use the simulator to verify operation of the EEPROM_read_string function. Remember to use Breakpoint, not Step Into (F11) operation.

c. Watch the video online on how this works.

Instructor Check off

1. Illustrate EEPROM_read_string function operation in simulator
2. Explain the code operation to instructor
3. All code are commented and contains an EEPROM module

5. **Inside the while(1) loop**
    a. when a push button (you decide the connection) is pressed, read the EEPROM data from address 0x0000 and display to LCD.
6. **Serial Terminal Interface**
    a. Add to previous procedures, the UART and LCD modules from prelab
    b. Display EEPROM data from address 0x0000 in LCD upon startup, you will write to EEPROM after serial terminal sends data.
    c. Comment out the EEPROM write statements at the beginning of the code.
    d. Load the code and you should see the same message in LCD.

7.
    a. Add to above procedure, upon startup, Transmit this message (from the EEPROM) to Serial Terminal. Check your connection between Serial port and Mega 2560, to ensure both Rx and Tx are connected.
    b. Upon startup, LCD shows the information saved in EEPROM. LCD also shows information in EEPROM upon press of a push button
    c. Send a message (less than 16 characters) from Serial Terminal to the Mega 2560. This value should be saved to EEPROM and also send back to Serial Terminal.
    d. Watch video online on this operation.

Instructor Check Off

1. Store a message to EEPROM via serial terminal
2. Press restart button, serial terminal and LCD display message from EEPROM. The message that was stored from #1
3. The message displayed should not contain any other characters besides what was saved in #1.

Deliverables/Check-offs:

1. Listings of your commented final codes.