```c
/*
 * FileName: main.c
 * Version: 1
 *
 * Created: 11/8/2022 11:50:01 PM
 * Author: Ethan Zeronik
 *
 * Operations: basic eeprom
 *
 * Hardware:
 *   Atmega2560          micro controller
 *   PORTG0:2            control for the lcd
 *   PORTl               data buss for the lcd
 */

/* NOTE: Includes */
#include <avr/io.h>

#include "Eeprom.h"

/* NOTE: Custom Macros */
// TODO: None

/* NOTE: Global Variables */
uint16_t addr = 0x0100;

/* NOTE: Function prototypes */
// TODO: none

/* NOTE: Application implementation */
// the main loop of the function, provided to us
int main(void)
{
    char const * const result = EEPROM_readString(addr);

    while(1)
    {
        // do nothing
    }
}

/* NOTE: Function implementations */
// TODO: none
```

```c
/*
 * FileName: main.c
 * Version: 1
 *
 * Created: 11/12/2022 8:19:38 PM
 * Author: Ethan Zeronik
 *
 * Operations: basic eeprom
 *
 * Hardware:
 *   Atmega2560          micro controller
 *   PORTG0:2            control for the lcd
 *   PORTL               LCD data
 *   PORTG.0             LCD RS
 *   PORTG.1             LCD R/W
 *   PORTG.2             LCD E
 */

/* NOTE: Includes */
#include <avr/io.h>
#include <avr/interrupt.h>

#include "Eeprom.h"
#include "Serial.h"
#include "LiquidCrystalDisplay.h"

/* NOTE: Custom Macros */
// TODO: None

/* NOTE: Global Variables */
// address for the stored value
uint16_t addr         = 0x0100;
// buffer for uart
char     message[16]  = {0};
uint8_t  messageIndex = 0;
uint8_t  readFlag     = 0;

/* NOTE: Function prototypes */
// handler
void asyncGetHandler(char c);

/* NOTE: Application implementation */
// the main loop of the function, provided to us
int main(void)
{
    LCD_init(&DDRG, &PORTG, &DDRL, &PORTL);

    SERIAL_uartInitAsync(serialUsart0, 9600);
    SERIAL_uartAsyncGetHandler(serialUsart0, &asyncGetHandler);

    // clear, home, and move cursor
    LCD_sendInstruction(0x01);
    LCD_sendInstruction(0x02);
    LCD_sendInstruction(0x80);

    LCD_sendString("Last Saved:");

```

```
58      char const * const result = EEPROM_readString(addr);
59
60      // move to last line and print previous eeprom
61      LCD_sendInstruction(0xc0);
62      LCD_sendString(result);
63
64      sei();
65
66      SERIAL_uartSend(serialUsart0, "atmega booted!\n\r");
67      SERIAL_uartSend(serialUsart0, result);
68
69      while(1)
70      {
71          if(readFlag)
72          {
73              EEPROM_writeString(message, addr);
74
75              SERIAL_uartSend(serialUsart0, "eeprom updated to:\n\r");
76              SERIAL_uartSend(serialUsart0, message);
77
78              readFlag = 0;
79          }
80      }
81  }
82
83  /* NOTE: Function implementations */
84  void asyncGetHandler(char c)
85  {
86      if(c != 0x0d && c != 0x0a && c != '\0')
87      {
88          if(messageIndex < 15)
89          {
90              // add to array
91              message[messageIndex]     = c;
92              message[messageIndex + 1] = '\0';
93
94              messageIndex++;
95          }
96      }
97      else
98      {
99          // set update flag
100         readFlag = 1;
101
102         // reset message
103         messageIndex = 0;
104     }
105 }
106
```

```h
/*
 * FileName: Eeprom.h
 * Version: 1
 *
 * Created: 11/8/2022 11:54 PM
 * Author: Ethan Zeronik
 *
 * Operations: eeprom definition
 */

#ifndef Eeprom_h_INCLUDED
#define Eeprom_h_INCLUDED

#if defined(__cplusplus)
extern "C" {
#endif

#include <stdint.h>

/* NOTE: Function prototypes */
// write a charater to the given address
void               EEPROM_writeChar(char in, uint16_t addr);
// reads a character
char               EEPROM_readChar(uint16_t addr);
// write a string to the given address
void               EEPROM_writeString(char const * const in, uint16_t addr);
// reads a string from the address
char const * const EEPROM_readString(uint16_t addr);

#if defined(__cplusplus)
} /* extern "C" */
#endif

#endif // Eeprom_h_INCLUDED
```

```c
/*
 * FileName: Eeprom.c
 * Version: 1
 *
 * Created: 11/8/2022 11:54 PM
 * Author: Ethan Zeronik
 *
 * Operations: eeprom implementation
 */

/* NOTE: Includes */
#include "Eeprom.h"

#include <avr/io.h>
#include <stdlib.h>

/* NOTE: Local declarations */
// reads blob from the given address
uint8_t read(uint16_t address);
// writes blob to the given address
void    write(uint8_t data, uint16_t address);

/* NOTE: Function implementations */
void EEPROM_writeChar(char in, uint16_t addr)
{
    write(in, addr);
}

char EEPROM_readChar(uint16_t addr)
{
    return (char)read(addr);
}

void EEPROM_writeString(char const * const in, uint16_t addr)
{
    uint16_t i = 0;

    while(in[i] != '\0')
    {
        write(in[i], addr + i);

        i++;
    }

    // add in a null terminator
    write('\0', addr + i);
}

char const * const EEPROM_readString(uint16_t addr)
{
    char * buf = malloc(sizeof(char) * 256);
    uint16_t     i   = 0;

    while(read(addr + i) != '\0')
    {
        // read into buffer
        buf[i] = read(addr + i);
```

```
58
59          i++;
60      }
61
62      // add in the null terminator and resize
63      buf[i] = '\0';
64      buf    = realloc(buf, sizeof(char) * (i + 1));
65
66      return buf;
67  }
68
69  /* NOTE: Local function implementations */
70  uint8_t read(uint16_t address)
71  {
72      // wait for read completion
73      while(EECR & (1 << EEPE))
74      {
75          // do nothing
76      }
77
78      // set address
79      EEAR = address;
80
81      // read and return
82      EECR |= (1 << EERE);
83      return EEDR;
84  }
85
86  void write(uint8_t data, uint16_t address)
87  {
88      // wait for write completion
89      while(EECR & (1 << EEPE))
90      {
91          // do nothing
92      }
93
94      // set address and data
95      EEAR = address;
96      EEDR = data;
97
98      // enable master program and start
99      EECR |= (1 << EEMPE);
100     EECR |= (1 << EEPE);
101 }
```