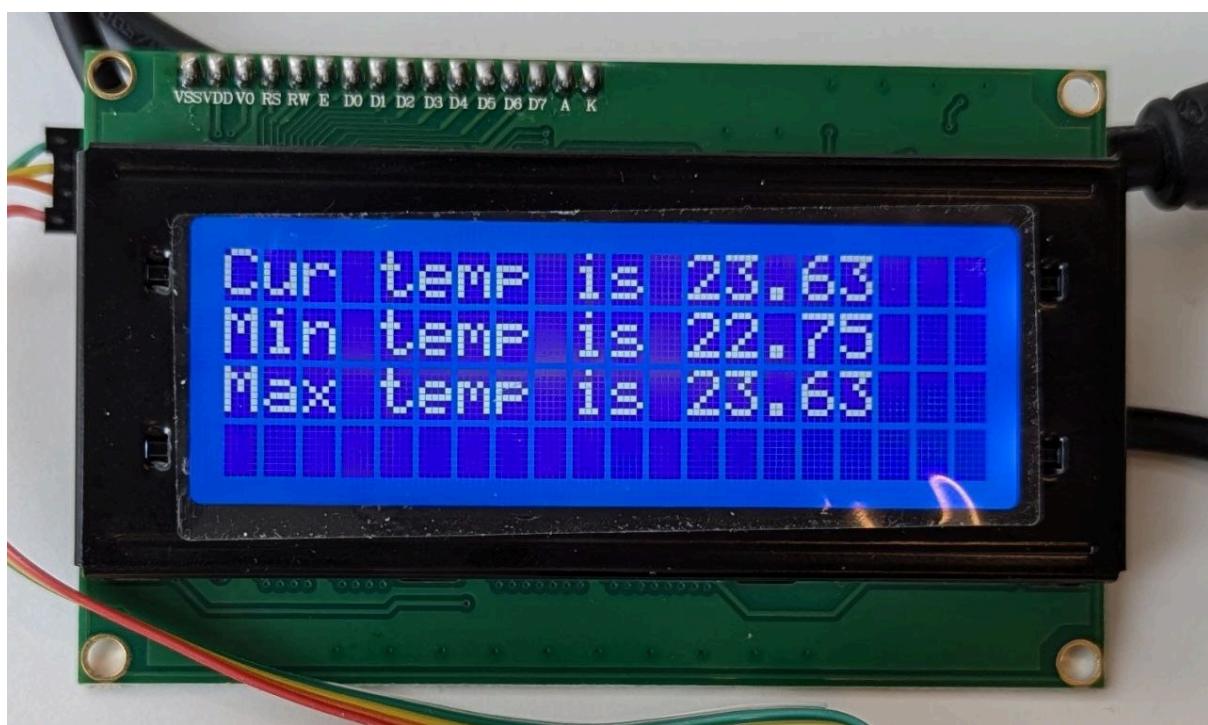
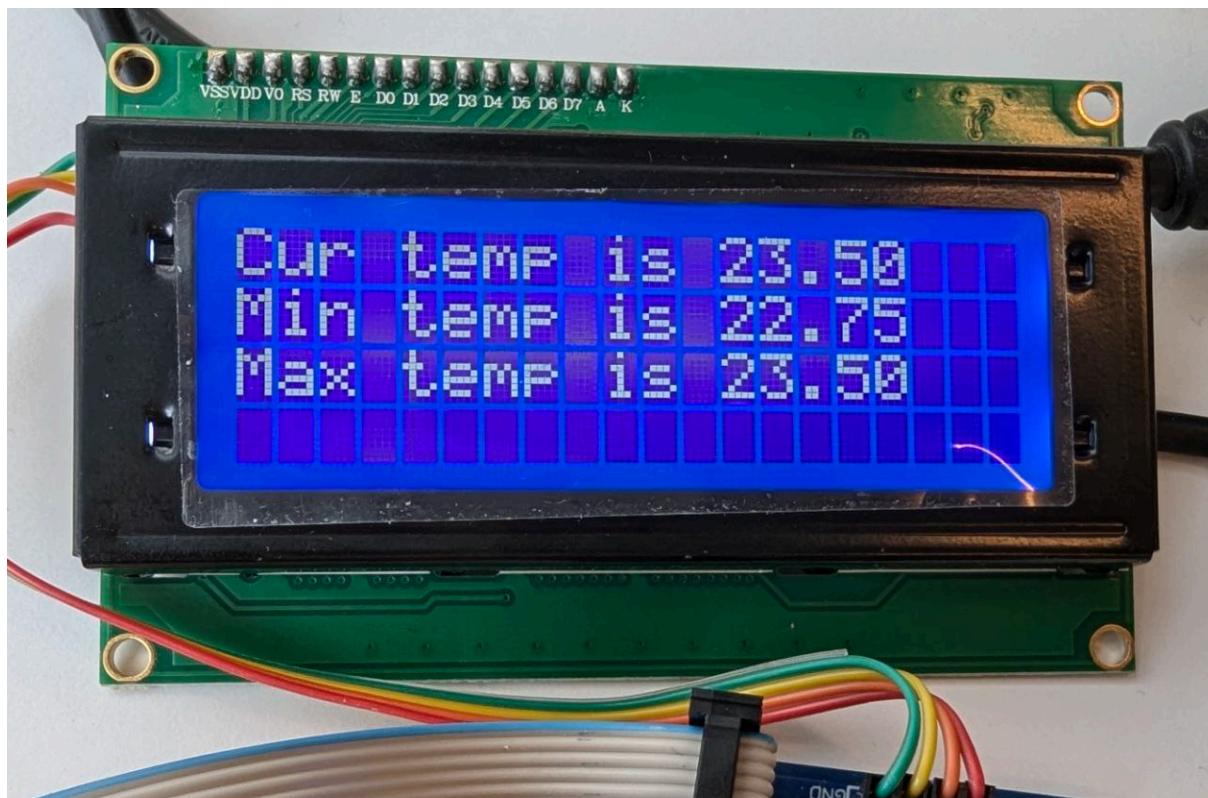


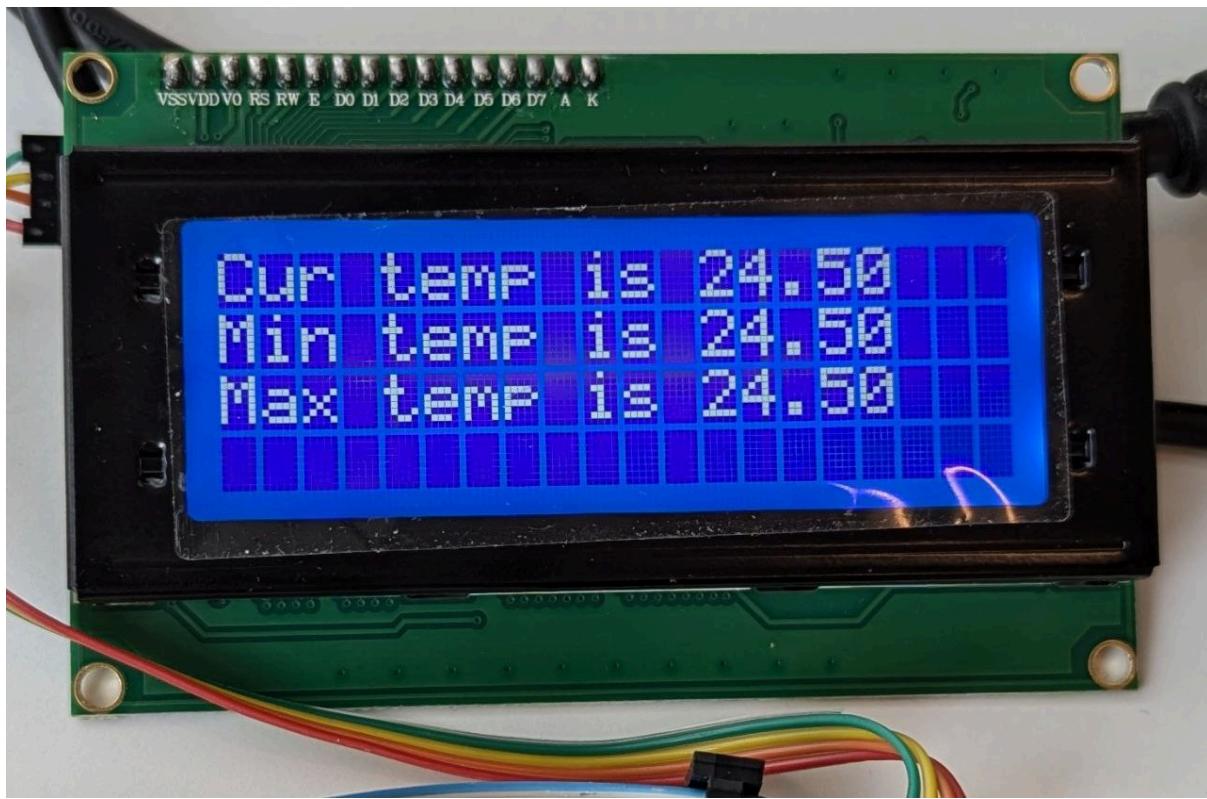
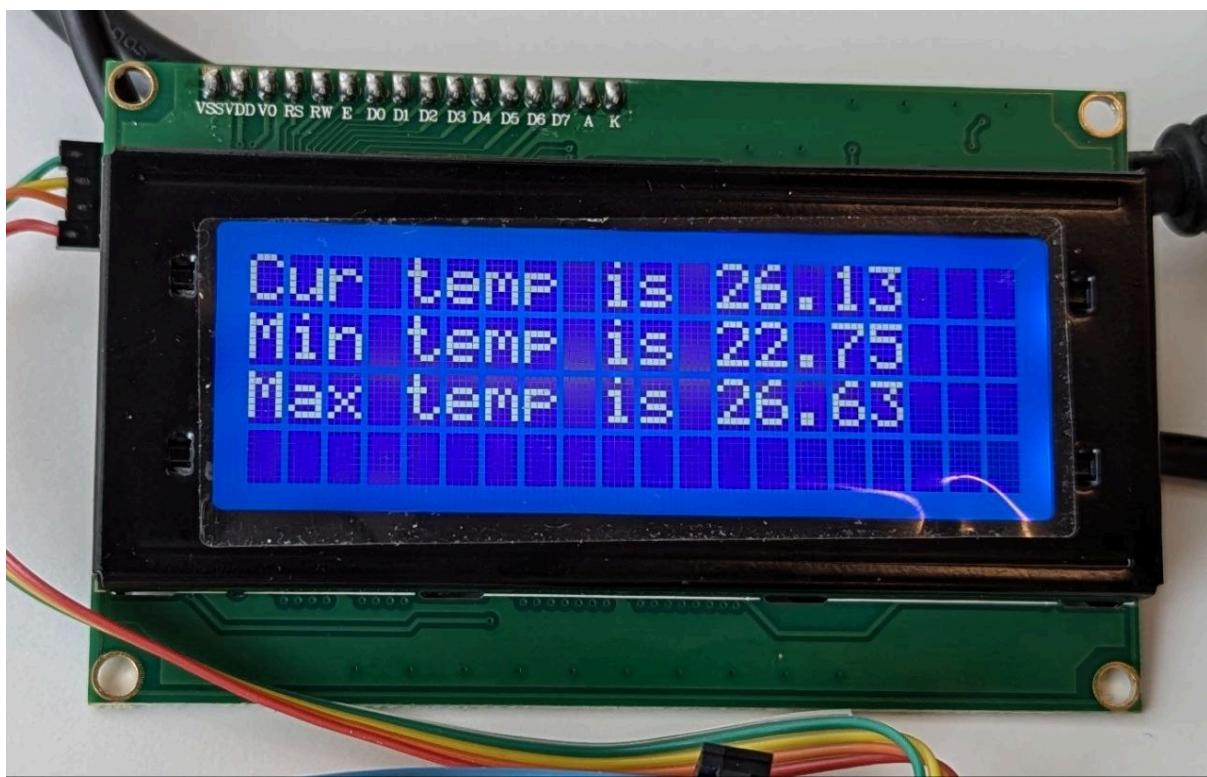
```
src\main.c

1  /*
2   * Practice Assignment 3 (PART C).c
3   *
4   * Created: 20/3/2025 10:43:27 AM
5   * Author : Sofiia
6   */
7
8
9 #include <stdio.h>
10 #include <avr/io.h>
11 #include <util/delay.h>
12
13 #include "uart.h"
14 #include "lcd.h"
15 #include "i2cmaster.h"
16 #include "lm75.h"
17 #include <avr/eeprom.h>
18
19
20 unsigned int address1 = 0; // Define an address to store minimum temperature at
21 unsigned int address2 = 4; // Define an address to store maximum temperature at
22
23 int main()
24 {
25     i2c_init(); // Initialise I2C communication
26     LCD_init(); // Initialise the LCD
27     lm75_init(); // Initialise the temperature sensor
28
29     float current_temp; // A variable that stores values of current temperature
30     float min_temp; // A variable that stores values of minimum temperature
31     float max_temp; // A variable that stores values of maximum temperature
32
33     /*Configuration for the Buttons*/
34     DDRC = 0xF0; // I/O board : PC0-PC3 configured as inputs for buttons
35     PORTC = 0x3F; // Enables internal pull at PC0-PC3 inputs
36
37     min_temp = eeprom_read_float((uint8_t*)address1); // Read a value that is stored at the
38     address
39     max_temp = eeprom_read_float((uint8_t*)address2); // Read a value that is stored at the
40     address
41
42     while(1)
43     { current_temp = (float)get_temperature(); // Current temperature is read from the
44       temperature sensor
45
46         if(current_temp < min_temp) // If the current temperature from the sensor is less than
47           the minimum temperature at the address
48         {
49             min_temp = current_temp; // Minimum temperature will be updated to be current = to
50             temperature
51             eeprom_write_float((uint8_t*)address1,(float)min_temp); // And its updated value is
52             written to the EEPROM address
```

```
47     }
48
49     if(current_temp > max_temp) //If the current temperature from the sensor is higher than
50     //the maximum temperature at the address
51     {
52         max_temp = current_temp; // Maximum temperature will be updated to be current = to
53         //temperature
54         eeprom_write_float((uint8_t*)address2,(float)max_temp); // And its updated value is
55         //written to the EEPROM address
56     }
57
58     LCD_set_cursor(0,0);
59     printf("Cur temp is %.2f", current_temp); // Current temperature value is printed on
60     //the LCD (from the beginnin gof the first row)
61     LCD_set_cursor(0,1);
62     printf("Min temp is %.2f", min_temp); // Minimum temperature value is printed on the
63     //LCD (from the beginnin gof the second row)
64     LCD_set_cursor(0,2);
65     printf("Max temp is %.2f", max_temp); // Current temperature value is printed on the
66     //LCD (from the beginnin gof the third row)
67
68     _delay_ms(1000);
69
70     if(PINC == 0b00110111) // If the FOURTH BUTTON is pressed
71     {
72         min_temp = current_temp; //Minimum temperature is reset to the current temperature
73         max_temp = current_temp; // //Maximum temperature is reset to the current temperature
74         eeprom_write_float((uint8_t*)address1,(float)min_temp); //And both values are stored
75         eeprom_write_float((uint8_t*)address2,(float)max_temp); //in the EEPROM memory after
76         //they are updated
77     }
78 }
79 }
```



If current temperature is higher than current maximum temperature >>
Maximum temperature updates



FOURTH BUTTON pressed resets all temperatures to the current one.