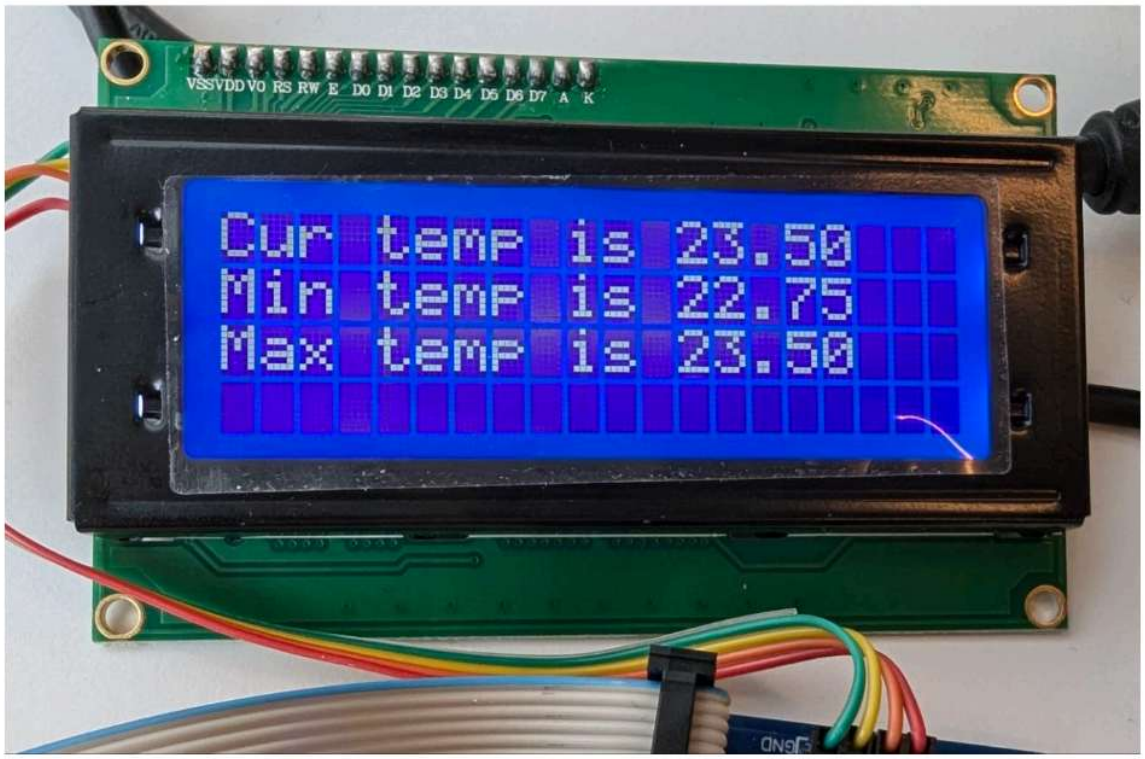
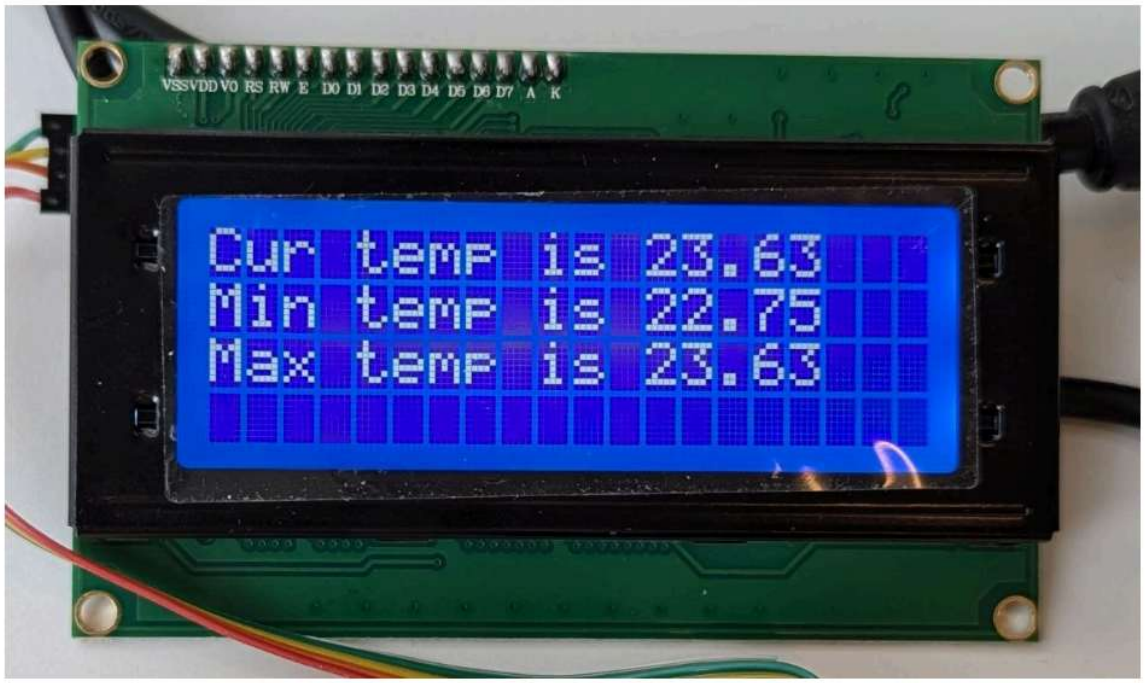
src\main.c

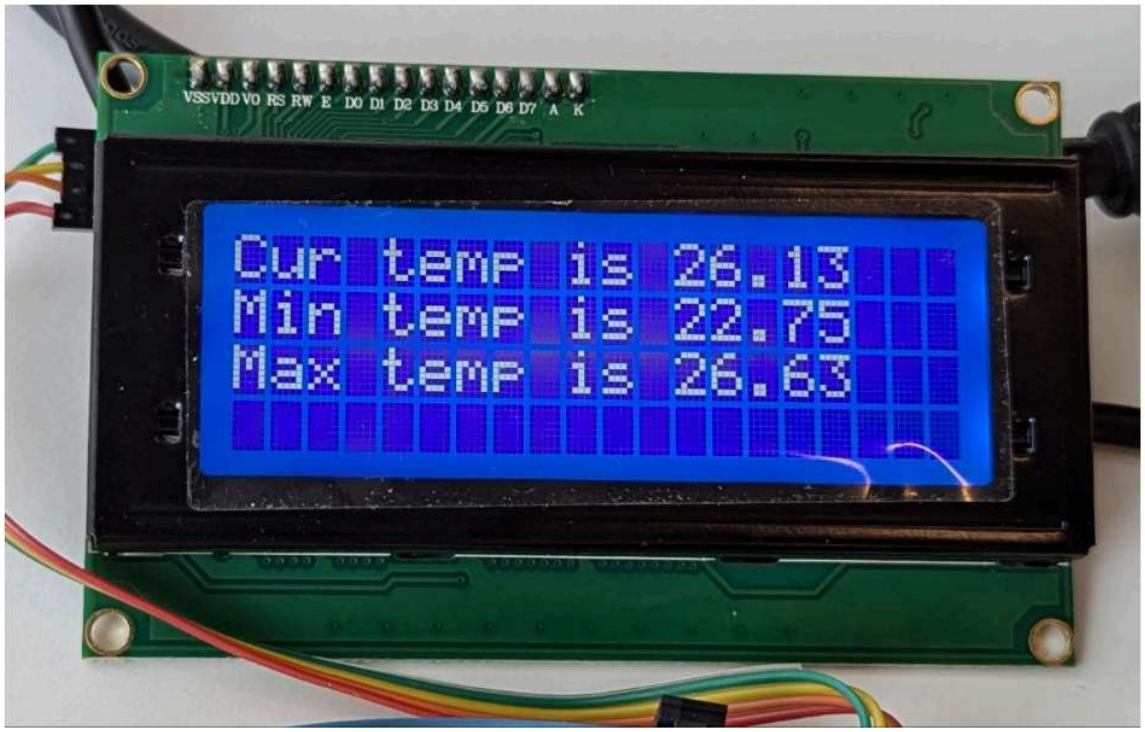
```
/*
 * Practice Assignment 3 (PART C).c
 *
 * Created: 20/3/2025 10:43:27 AM
 * Author : Sofiia
 */


#include <stdio.h>
#include <avr/io.h>
#include <util/delay.h>

#include "usart.h"
#include "lcd.h"
#include "i2cmaster.h"
#include "lm75.h"
#include <avr/eeprom.h>


unsigned int address1 = 0; // Define an address to store minimum temperature at
unsigned int address2 = 4; // Define an address t store maximum temperature at

int main()
 {
  i2c_init(); // Intitialise I2C communication
  LCD_init(); // Initialise the LCD
  lm75_init(); // Initiaise the temperature sensor

  float current_temp; // A variable that stores values of current temperature
  float min_temp; // A variable that stores values of minimum temperature
  float max_temp; // A variable that stores values of maximum temperature

  /*Configuration for the Buttons*/
  DDRC = 0xF0; // I/O board : PC0-PC3 configured as inputs for buttons
  PORTC = 0x3F; // Enables internal pull at PC0-PC3 inputs

  min_temp = eeprom_read_float((uint8_t*)address1); // Read a value that is stored at the address
  max_temp = eeprom_read_float((uint8_t*)address2); // Read a value that is stored at the address

  while(1)
  { current_temp =  (float)get_temperature(); // Current temperature is read from the temperature sensor

    if(current_temp < min_temp) // If the current temperature from the sensor is less than the minimum temperature at the address
    {
      min_temp = current_temp; // Minimum temperature will be updated to be current = to temperature
      eeprom_write_float((uint8_t*)address1,(float)min_temp); // And its updated value is written to the EEPROM address
```

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





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



FOURTH BUTTON pressed resets all temperatures to the current one.