```
/*
 * SCD41_Multimodule_LED_CO2_Level_Lab11Task2.c
 *
 * Created: 4/26/2022 9:15:52 PM
 * Author : jason
 */

#include <avr/io.h>
#include <math.h>
#define F_CPU 4000000
#include "lcd_dog_AVR128_driver.h"
#include "SCD41_AVR128_driver.h"
#include "USART3_asynch_transmit.h"
#include "MCP23017_CO2_level_LED_simple_display.h"
#include <util/delay.h>

#define MAX_INPUT_DISPLAY 5


uint16_t CO2;
uint16_t Temp;
uint16_t Rh;

uint16_t baudRate = 9600;   //For the baud rate of USART3
uint8_t dataBits = USART_CHSIZE_8BIT_gc;    //For the (character size)  CHSIZE[2:0]
unsigned char parity = 0x00;    //PMODE[1:0]


int main(void)
{
    init_lcd_dog();      //Initialize the buffer of the LCD

    //Function to initialize the USART3 baud rate, data bit and parity
    USART3_init(baudRate, dataBits, parity);

    while (1)
    {
        I2C0_SCD41_init();   //Initializes  the AVR128DB48 I2C0 to communicate with
          SCD41 to change its BAUD RATE for SCD31
        SCD41_start_periodic_measurement(I2CSLAVE_ADDR_WRITE,
          ADDRESS_STARTPERIODIC_MSB, ADDRESS_STARTPERIODIC_LSB);

        //Keep polling until data is ready which can be measured
        while(!SCD41_get_data_ready_status(I2CSLAVE_ADDR_WRITE,
          ADDRESS_GETDATAREADY_MSB, ADDRESS_GETDATAREADY_LSB));

        SCD41_read_measurement(I2CSLAVE_ADDR_WRITE, ADDRESS_READMEASUR_MSB,
          ADDRESS_READMEASUR_LSB);

        CO2 = getParseCO2;
        Temp = -45 + ( (175 * getParseTemp) / (pow(2, 16)));
        Rh = 100 * (((float)getParseRh) / (pow(2, 16)));
```

```c
        //Print the CO2 value into LCD buffer
        sprintf(dsp_buff1, "CO2: %d", CO2);
        sprintf(dsp_buff2, "Temp: %d", Temp);
        //Print the humidity value into LCD buffer
        sprintf(dsp_buff3, "Relative Hum: %d", Rh);
        //Update the 3 line messages into the LCD buffer
        update_lcd_dog();

        I2C0_MCP23017_init(); //Initializes  the AVR128DB48 I2C0 to communicate with MCP23017 to change its BAUD RATE for MCP23017
        MCP23017_I2C_init();  //Initializes  GPIO (GPB) as outputs for outputting to the LEDS

        if(CO2 >= 400 && CO2 <= 499){
            MCP23017_I2C_write(WRITE_opcode, OLATBaddr_b1, 0x7F); //0111 1111 GPB7-GPB0
        }
        else if(CO2 >= 500 && CO2 <= 599){
            MCP23017_I2C_write(WRITE_opcode, OLATBaddr_b1, 0x3F); //0011 1111 GPB7-GPB0
        }
        else if(CO2 >= 600 && CO2 <= 699){
            MCP23017_I2C_write(WRITE_opcode, OLATBaddr_b1, 0x1F); //0001 1111 GPB7-GPB0
        }
        else if(CO2 >= 700 && CO2 <= 799){
            MCP23017_I2C_write(WRITE_opcode, OLATBaddr_b1, 0x0F); //0000 1111 GPB7-GPB0
        }
        else if(CO2 >= 800 && CO2 <= 899){
            MCP23017_I2C_write(WRITE_opcode, OLATBaddr_b1, 0x07); //0000 0111 GPB7-GPB0
        }
        else if(CO2 >= 900 && CO2 <= 999){
            MCP23017_I2C_write(WRITE_opcode, OLATBaddr_b1, 0x03); //0000 0011 GPB7-GPB0
        }
        else if(CO2 >= 1000 && CO2 <= 1099){
            MCP23017_I2C_write(WRITE_opcode, OLATBaddr_b1, 0x01); //0000 0001 GPB7-GPB0
        }
        else if(CO2 >= 1100 && CO2 <= 1199){
            MCP23017_I2C_write(WRITE_opcode, OLATBaddr_b1, 0x00); //0000 0000 GPB7-GPB0
        }

        //Where it will transmit the entire array of strings to display in TeraTerm or Termite
        char *inputUSART3DataDisplay[] = {dsp_buff1, "    ", dsp_buff2, "    ",
          dsp_buff3};

        //Loop through all the strings in the array
        for(int i = 0; i < MAX_INPUT_DISPLAY; i++){
            USART3_sendString(inputUSART3DataDisplay[i]);
        }
        USART3_sendString("\r\n");  //For new line of the Teraterm or Termite
        _delay_ms(1000);  //Transmit the readings once every 1 second
    }
}
```