

ДОДАТОК А

Лістинг програми

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

LiquidCrystal_I2C LCD(0x27, 16, 2);

int SwB4 = 5;
boolean VB4 = true;

int temperature1 = 0;
int temperature2 = 0;
float averageTemperature;

// Пины для реле и мотора
const int relayHeater = 3;
const int relayFan = 6;

unsigned long previousMillis = 0;
const long interval = 500; // Интервал для обновления дисплея в миллисекундах
const long errorDisplayInterval = 3000; // Интервал для отображения сообщений об
ошибках (в миллисекундах)

unsigned long lastErrorDisplayTime = 0;

void setup() {
  Serial.begin(9600);
  pinMode(SwB4, INPUT);
  pinMode(relayHeater, OUTPUT);
  pinMode(relayFan, OUTPUT);

  LCD.init();
  LCD.backlight();
}

void loop() {
  // Считывание температуры
  int temp1Raw = analogRead(A0);
  int temp2Raw = analogRead(A1);
  temperature1 = map(((temp1Raw - 20) * 3.04), 0, 1023, -40, 125);
  temperature2 = map(((temp2Raw - 20) * 3.04), 0, 1023, -40, 125);
  averageTemperature = (temperature1 + temperature2) / 2.0;

  VB4 = digitalRead(SwB4);
```

```

if (VB4 == 0) {
    // Проверка ошибок компонентов
    checkErrors(temp1Raw, temp2Raw);

    // Управление нагревателем
    if (averageTemperature < 25) {
        digitalWrite(relayHeater, HIGH);
        LCD.clear();
        LCD.setCursor(0, 0);
        LCD.print("Heater ON");
    } else {
        digitalWrite(relayHeater, LOW);
        LCD.setCursor(0, 2);
        LCD.print("Heater OFF");
    }

    // Регулировка скорости мотора
    int motorSpeed;
    if (averageTemperature > 25) {
        // Увеличение скорости мотора, если температура выше 25
        motorSpeed = map(averageTemperature, 25, 125, 128, 255); // Map temperature
to PWM value (128-255)
    } else {
        // Установка исходной скорости мотора, если температура ниже или равна
25
        motorSpeed = 128; // Средняя скорость
    }

    motorSpeed = constrain(motorSpeed, 0, 255); // Ensure PWM value is within 0-
255
    analogWrite(relayFan, motorSpeed);

    // Добавим отладочные сообщения для проверки значений
    Serial.print("averageTemperature: ");
    Serial.print(averageTemperature);
    Serial.print(" Motor Speed: ");
    Serial.println(motorSpeed);

    // Обновляем дисплей с определенным интервалом
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;
        WindowLCD();
    }
}

```

```

}

void checkErrors(int temp1Raw, int temp2Raw) {

    unsigned long currentMillis = millis();

    // Проверка датчика температуры 1
    if (temp1Raw <= 0 || temp1Raw >= 1023) {
        if (currentMillis - lastErrorDisplayTime >= errorDisplayInterval) {
            lastErrorDisplayTime = currentMillis;
            LCD.clear();
            LCD.setCursor(0, 0);
            LCD.print("Temp1 Sens Error");
            Serial.println("Temp1 Sensor Error");
        }
    }

    // Проверка датчика температуры 2
    if (temp2Raw <= 0 || temp2Raw >= 1023) {
        if (currentMillis - lastErrorDisplayTime >= errorDisplayInterval) {
            lastErrorDisplayTime = currentMillis;
            LCD.clear();
            LCD.setCursor(0, 1);
            LCD.print("Temp2 Sens Error");
            Serial.println("Temp2 Sensor Error");
        }
    }

    if (digitalRead(relayHeater) == HIGH && averageTemperature >= 25) {
        if (currentMillis - lastErrorDisplayTime >= errorDisplayInterval) {
            lastErrorDisplayTime = currentMillis;
            LCD.clear();
            LCD.setCursor(0, 2);
            LCD.print("Heater Relay Error");
            Serial.println("Heater Relay Error");
        }
    }
}

void WindowLCD() {
    LCD.clear();
    LCD.setCursor(0, 0);
    LCD.print("Temp1: ");
    LCD.setCursor(7, 0);
    LCD.print(temperature1);
}

```

```
LCD.setCursor(0, 1);  
LCD.print("Temp2: ");  
LCD.setCursor(7, 1);  
LCD.print(temperature2);  
  
}
```

ДОДАТОК Б

Демонстраційний матеріал

