**APPENDIX 1:**

#include <stdio.h>

#include <stdbool.h>

// Function to read temperature from sensor

float readTemperature() {

// Code to read temperature from sensor

// Replace this with your actual sensor reading code

// For example, you might use analog-to-digital conversion or a communication protocol

// Return the temperature value as a float

return 0;

}

// Function to control the fan based on the temperature

void controlFan(float temperature) {

static bool fanState = false; // Stores the current state of the fan

// Check if temperature is increasing by 35 degrees

if (temperature >= 35.0) {

fanState = true; // Turn on the fan

} // Check if temperature is decreasing below 30 degrees

if (temperature <= 30.0) {

fanState = true; // Turn on the fan to maintain temperature

}

// Code to control the fan

// Replace this with your actual fan control code

// For example, you might use digital outputs or a communication protocol

// Implement the logic to turn on or off the fan based on the fanState variable }

int main() {

// Variable to store the desired temperature

float desiredTemperature = 25.0;

// Main loop

while (true) {

// Read the temperature

float currentTemperature = readTemperature();

// Print the current temperature

printf("Current Temperature: %.2f\n", currentTemperature);

// Control the fan based on the temperature

controlFan(currentTemperature);

// Check if the current temperature is within an acceptable range

if (currentTemperature >= desiredTemperature - 1.0 && currentTemperature <= desiredTemperature + 1.0) {

// Temperature is within an acceptable range

// Implement any necessary actions or alerts here }

// Delay between temperature readings and fan control

// Adjust the delay based on your system requirements

// This is just an example delay for demonstration purposes

for (int i = 0; i < 10000000; i++) {

// Do nothing, just waste some time

} }

return 0;

}

**APPENDIX 2:**

#include <ThingSpeak.h>

#include <WiFi.h>

// WiFi credentials

const char\* ssid = "your\_wifi\_ssid";

const char\* password = "your\_wifi\_password";

// ThingSpeak settings

const char\* server = "api.thingspeak.com";

const char\* apiKey = "your\_thingspeak\_api\_key";

const unsigned long channelID = your\_channel\_id;

// Thermocouple connections

#define THERMOCOUPLE\_CLK 13

#define THERMOCOUPLE\_CS 10

#define THERMOCOUPLE\_DO 12

// Relay connection

#define RELAY\_PIN 9

// Temperature thresholds

#define LOWER\_THRESHOLD 50.0

#define UPPER\_THRESHOLD 100.0

WiFiClient client;

Adafruit\_MAX31855 thermocouple(THERMOCOUPLE\_CLK, THERMOCOUPLE\_CS, THERMOCOUPLE\_DO);

void setup() {

// Initialize serial communication

Serial.begin(115200);

// Connect to Wi-Fi

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.println("Connecting to WiFi...");

}

Serial.println("Connected to WiFi");

// Initialize ThingSpeak

ThingSpeak.begin(client);

}

void loop() {

// Read temperature from thermocouple

double temperature = thermocouple.readCelsius();

// Print temperature to serial monitor

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.println(" °C");

// Send temperature data to ThingSpeak

ThingSpeak.setField(1, temperature);

int httpCode = ThingSpeak.writeFields(channelID, apiKey);

if (httpCode == 200) {

Serial.println("Data sent to ThingSpeak successfully");

} else {

Serial.println("Error sending data to ThingSpeak"); }