Vermicompost Monitoring & Management System - Arduino Code

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

#define DHTPIN 2        // Pin connected to DHT11 data pin
#define MAX_TIMINGS 85
#define DHTTYPE DHT11   // DHT11 sensor type
#define moisturePin A0  // Analog pin A0 for the soil moisture sensor
#define PUMP_PIN P2_2     // Pin connected to the pump control
#define FAN_PIN P2_0       // Pin connected to the fan control
#define LED_GREEN  P1_4
#define LED_RED P1_5       // Pin connected to Red LED (for temperature indication)

int dht11_dat[5] = {0, 0, 0, 0, 0};  // Array to hold the data from DHT11
int moistureThreshold = 40;           // Threshold for moisture level (in percentage)
float temperatureThreshold = 30+.0;    // Threshold for fan activation (in Celsius)

LiquidCrystal_I2C lcd(0x27, 16, 2);   // Create LCD object for I2C Display

unsigned long lastReadingTime = 0;     // To track time for sensor readings
const unsigned long interval = 2000;   // 2-second interval
bool showTemperatureHumidity = true;    // Toggle between temp/humidity and moisture

int dryValue = 4095;   // Analog reading when the soil is completely dry
int wetValue = 1500;    // Analog reading when the soil is completely wet

void setup() {
   Serial.begin(9600);
   pinMode(DHTPIN, OUTPUT);
   digitalWrite(DHTPIN, HIGH);
   pinMode(PUMP_PIN, OUTPUT);
   digitalWrite(PUMP_PIN, HIGH);
   pinMode(FAN_PIN, OUTPUT);
   digitalWrite(FAN_PIN, HIGH);
   pinMode(LED_GREEN, OUTPUT);
   digitalWrite(LED_GREEN, HIGH);
   pinMode(LED_RED, OUTPUT);
   digitalWrite(LED_RED, HIGH);

   lcd.begin();
   lcd.backlight();
   lcd.setCursor(0, 0);
   lcd.print("Initializing...");
   delay(2000);
}

void loop() {
   unsigned long currentMillis = millis();
   if (currentMillis - lastReadingTime >= interval) {
      lastReadingTime = currentMillis;
      showTemperatureHumidity = !showTemperatureHumidity;
```

```
if (showTemperatureHumidity) {
  int chk = readDHT11();
  if (chk == 0) {
    float temperature = dht11_dat[2] + dht11_dat[3] / 10.0;
    if (temperature > temperatureThreshold) {
      digitalWrite(FAN_PIN, LOW);
      Serial.println("Fan: ON");
      digitalWrite(LED_GREEN, LOW);
      Serial.println("GREEN LED: ON");
    } else {
      digitalWrite(FAN_PIN, HIGH);
      Serial.println("Fan: OFF");
      digitalWrite(LED_GREEN, HIGH);
      Serial.println("GREEN LED: OFF");
    }
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temp: ");
    lcd.print(temperature);
    lcd.print(" C");
    lcd.setCursor(0, 1);
    lcd.print("Hum: ");
    lcd.print(dht11_dat[0]);
    lcd.print(" %");
  } else {
    displayError("DHT11 Error");
  }
} else {
  int moistureLevel = analogRead(moisturePin);
  int moisturePercent = map(moistureLevel, dryValue, wetValue, 0, 100);
  moisturePercent = constrain(moisturePercent, 0, 100);
  if (moisturePercent < moistureThreshold) {
    digitalWrite(PUMP_PIN, LOW);
    Serial.println("Pump: ON");
    digitalWrite(LED_RED, LOW);
    Serial.println("RED LED: ON");
  } else {
    digitalWrite(PUMP_PIN, HIGH);
    Serial.println("Pump: OFF");
    digitalWrite(LED_RED, HIGH);
    Serial.println("RED LED: OFF");
  }
  Serial.print("Soil Moisture Level: ");
  Serial.print(moisturePercent);
  Serial.println(" %");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Moisture Level:");
  lcd.setCursor(0, 1);
  lcd.print(moisturePercent);
  lcd.print(" %");
```

```
        }
    }
    delay(2000);
}

int readDHT11() {
    uint8_t laststate = HIGH;
    uint8_t counter = 0;
    uint8_t j = 0, i;
    dht11_dat[0] = dht11_dat[1] = dht11_dat[2] = dht11_dat[3] = dht11_dat[4] = 0;
    pinMode(DHTPIN, OUTPUT);
    digitalWrite(DHTPIN, LOW);
    delay(18);
    digitalWrite(DHTPIN, HIGH);
    delayMicroseconds(40);
    pinMode(DHTPIN, INPUT);

    for (i = 0; i < MAX_TIMINGS; i++) {
        counter = 0;
        while (digitalRead(DHTPIN) == laststate) {
            counter++;
            delayMicroseconds(1);
            if (counter == 255) break;
        }
        laststate = digitalRead(DHTPIN);
        if (counter == 255) break;
        if ((i >= 4) && (i % 2 == 0)) {
            dht11_dat[j / 8] <<= 1;
            if (counter > 16)
                dht11_dat[j / 8] |= 1;
            j++;
        }
    }
    if ((j >= 40) && (dht11_dat[4] == ((dht11_dat[0] + dht11_dat[1] + dht11_dat[2] + dht11_dat[3]) & 0xFF))) {
        return 0;
    }
    return 1;
}

void displayError(const char* message) {
    Serial.println(message);
    digitalWrite(LED_RED, LOW);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(message);
    delay(2000);
}
```