**THIẾT KẾ HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ VÀ ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG ESP32, CẢM BIẾN DHT22 VÀ MODULE RELAY TRONG ỨNG DỤNG TRỒNG NẤM BÀO NGƯ**

**1. Giới Thiệu và Mục Tiêu Đề Tài**

**1.1. Bối Cảnh và Tầm Quan Trọng**

Trong nhiều lĩnh vực, từ nông nghiệp thông minh (nhà kính), kho lạnh, nhà ở dân dụng cho đến các hệ thống công nghiệp, việc giám sát và điều khiển nhiệt độ/độ ẩm là một yêu cầu thiết yếu để duy trì điều kiện môi trường lý tưởng, bảo quản hàng hóa, đảm bảo sức khỏe con người, hoặc tối ưu hóa hiệu suất thiết bị. Hiện nay, các giải pháp điều khiển truyền thống thường bao gồm các thiết bị cơ điện cồng kềnh, thiếu khả năng kết nối mạng và đòi hỏi sự can thiệp thủ công cao.

Việc ứng dụng các hệ thống tự động dựa trên vi điều khiển nhúng hiện đại như **ESP32** giúp chuyển đổi hoàn toàn phương thức giám sát. Hệ thống nhúng không chỉ tăng độ **chính xác** trong đo lường (nhờ cảm biến kỹ thuật số DHT22) mà còn cho phép **phản ứng tức thời** với các biến động môi trường. Điều này đặc biệt quan trọng trong các môi trường nhạy cảm, nơi mà sự thay đổi nhỏ về nhiệt độ/độ ẩm có thể dẫn đến thiệt hại lớn (ví dụ: sản phẩm nông nghiệp bị hư hỏng, thiết bị điện tử bị ẩm). Đề tài này hướng tới việc tạo ra một **giải pháp chi phí thấp, hiệu quả cao** để giải quyết nhu cầu tự động hóa thiết yếu này.

**1.2. Mục Tiêu Cụ Thể của Đề Tài**

Đề tài này nhằm mục đích thiết kế, xây dựng và thử nghiệm một hệ thống nhúng có khả năng tự động hóa việc giám sát và điều khiển thiết bị làm mát dựa trên sự thay đổi của nhiệt độ môi trường.

Các mục tiêu chính bao gồm:

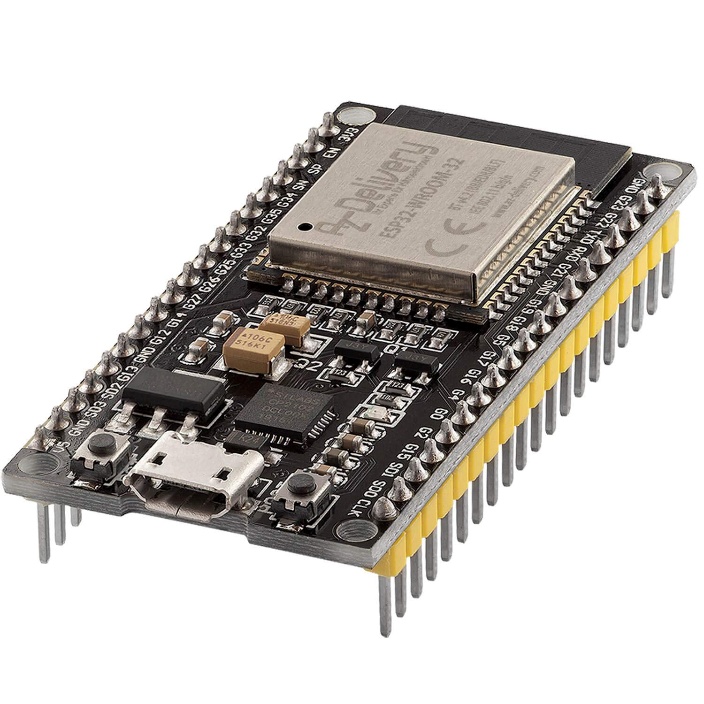
* **Thiết lập thành công kết nối** và giao tiếp giữa Vi điều khiển **ESP32** và Cảm biến nhiệt độ/độ ẩm **DHT22**, đảm bảo tính toàn vẹn và ổn định của dữ liệu truyền dẫn (giao thức 1-Wire).
* **Đo lường chính xác** và hiển thị liên tục các giá trị nhiệt độ (T) và độ ẩm (H) thu được từ cảm biến. Giá trị T được sử dụng làm biến đầu vào chính cho thuật toán điều khiển.
* **Thiết kế thuật toán điều khiển** **Ngưỡng On/Off** đơn giản nhưng hiệu quả để tự động kích hoạt/ngắt kết nối thiết bị tải (Relay) khi nhiệt độ vượt quá ngưỡng cho phép (mặc định là 30.0oC), mô phỏng hoạt động của một bộ điều nhiệt (thermostat) kỹ thuật số.
* Đảm bảo hệ thống hoạt động ổn định, có độ trễ thấp và **hiển thị trạng thái rõ ràng, trực quan** trên màn hình **LCD I2C** (các thông số T, H, và trạng thái Relay: ON/OFF).
* Cung cấp nền tảng để dễ dàng mở rộng tính năng hiển thị dữ liệu cục bộ (**LCD I2C**) hoặc giám sát, lưu trữ dữ liệu từ xa qua nền tảng IoT Cloud (ví dụ: ThingSpeak) bằng kết nối Wi-Fi tích hợp của ESP32.

**2. Cơ Sở Lý Thuyết và Linh Kiện Sử Dụng**

**2.1.** **Vi Điều Khiển ESP32 NodeMCU**

ESP32 là dòng vi điều khiển mạnh mẽ của Espressif Systems, nổi bật với việc tích hợp Wi-Fi và Bluetooth trên chip (System on a Chip - SoC). Đây là lựa chọn hàng đầu cho các dự án IoT (Internet of Things).

* **Ưu điểm:** Hiệu năng mạnh mẽ (lõi kép), tích hợp kết nối không dây, nhiều chân GPIO đa năng, giá thành hợp lý.
* **Vai trò trong đề tài:** Xử lý dữ liệu từ DHT22, thực hiện logic điều khiển, và gửi tín hiệu kích hoạt đến module relay.



Hình : ESP32 NodeMCU

**2.2. Cảm Biến DHT22 (AM2302)**

DHT22 là cảm biến nhiệt độ và độ ẩm kỹ thuật số phổ biến, cung cấp độ chính xác cao hơn so với DHT11.

* **Thông số kỹ thuật chính:**
  + Phạm vi đo nhiệt độ: **-40°C đến 80°C** (độ chính xác ±0.5°C).
  + Phạm vi đo độ ẩm: **0% đến 99.9%** (độ chính xác ±2%).
  + Giao tiếp: Giao thức 1-Wire (chỉ cần 1 chân data để truyền dữ liệu).
* **Vai trò trong đề tài:** Cung cấp dữ liệu môi trường đầu vào cho hệ thống

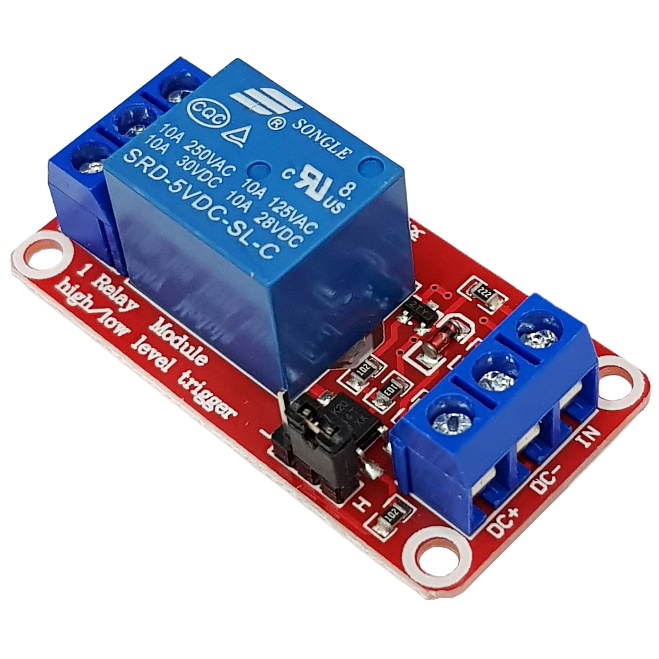


Hình : DHT 22

**2.3. Module Relay 5V**

Relay là một công tắc điện tử cho phép vi điều khiển (điện áp thấp) điều khiển các thiết bị điện áp cao, dòng điện lớn (thiết bị tải).

* **Nguyên tắc hoạt động:** Cuộn dây điện từ trong relay được kích hoạt bởi tín hiệu (HIGH hoặc LOW) từ ESP32, làm đóng/mở mạch điện của thiết bị tải.
* **Vai trò trong đề tài:** Là cầu nối để ESP32 bật/tắt thiết bị làm mát (tải).

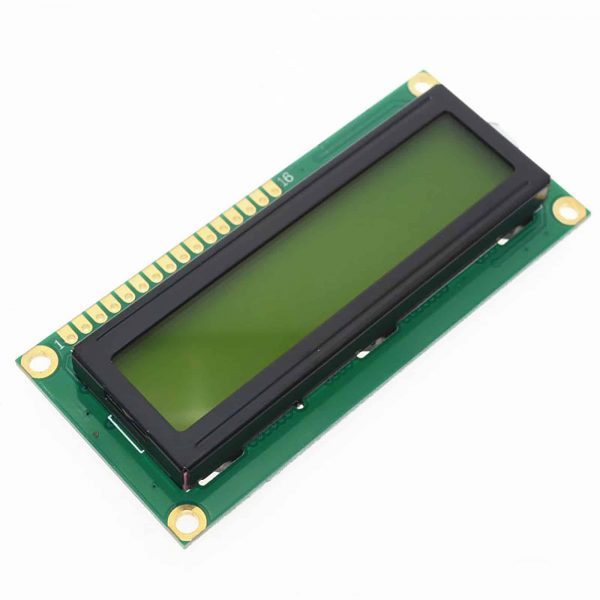


Hình 3: Module Replay 5V

**2.4. Màn Hình LCD 1602 I2C**

Màn hình LCD 1602 (16 cột, 2 dòng) được gắn kèm module I2C (PCF8574) giúp giao tiếp chỉ bằng 2 chân tín hiệu (SDA và SCL), tiết kiệm chân GPIO của ESP32.

* **Vai trò:** Hiển thị trực tiếp các giá trị đo và trạng thái hệ thống.

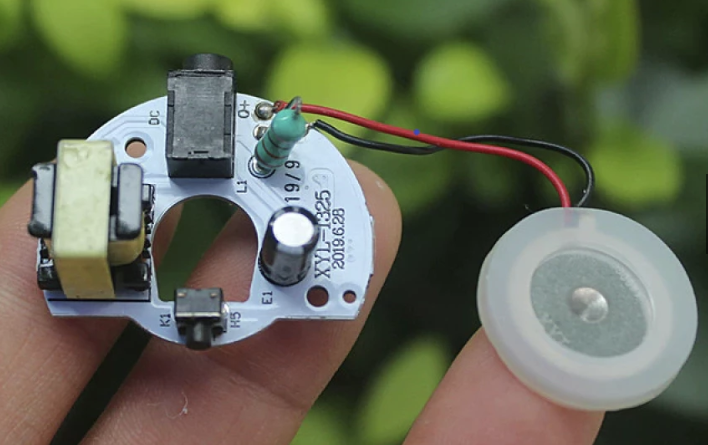


Hình 4: Màn Hình LCD

**2.5. Quạt Làm Mát (DC 5V)**

Quạt làm mát là một thiết bị sử dụng động cơ điện một chiều (DC) để tạo ra luồng không khí. Trong các hệ thống điện tử, chúng thường được dùng để tản nhiệt cho các linh kiện. Tuy nhiên, trong mô hình này, quạt đóng vai trò thiết yếu trong việc điều hòa môi trường.

* **Nguyên lý hoạt động:** Động cơ không chổi than (brushless) sử dụng năng lượng điện 5V để làm quay các cánh quạt, tạo ra luồng gió.
* **Vai trò trong đề tài:**
  + **Điều hòa nhiệt độ:** Khi nhiệt độ trong thùng trồng vượt ngưỡng cài đặt, quạt sẽ được kích hoạt để thổi khí nóng ra ngoài và hút khí mát hơn từ bên ngoài vào, giúp hạ nhiệt độ.
  + **Trao đổi không khí trong lành (Fresh Air Exchange - FAE):** Đây là chức năng cực kỳ quan trọng trong việc trồng nấm. Quạt giúp cung cấp oxy tươi và đẩy khí CO₂ (do nấm thải ra) ra ngoài, kích thích sự phát triển của quả thể nấm.
  + **Điều khiển:** Quạt được điều khiển BẬT/TẮT bởi ESP32 thông qua một kênh của module relay.



**2.6. Mạch Phun Sương Siêu Âm (DC 5V)**

Mạch phun sương siêu âm là thiết bị tạo ra hơi ẩm dạng sương mù mịn từ nước bằng cách sử dụng sóng siêu âm, thay vì dùng nhiệt để đun sôi.

* **Nguyên lý hoạt động:** Mạch điều khiển cấp một dòng điện xoay chiều có tần số siêu âm (khoảng 110 kHz) cho một đĩa gốm áp điện (piezoelectric transducer). Đĩa gốm này rung động với tốc độ cực nhanh, làm cho các phân tử nước bị "xé" ra khỏi bề mặt và biến thành các hạt sương siêu nhỏ.
* **Vai trò trong đề tài:**
  + **Điều hòa độ ẩm:** Đây là thiết bị chấp hành chính để tăng độ ẩm tương đối bên trong thùng trồng. Khi cảm biến DHT22 báo độ ẩm xuống dưới ngưỡng cài đặt, mạch phun sương sẽ được kích hoạt để nhanh chóng đưa độ ẩm về mức lý tưởng cho nấm phát triển.
  + **Điều khiển:** Tương tự như quạt, mạch phun sương được điều khiển BẬT/TẮT bởi ESP32 thông qua kênh còn lại của module relay.



**2.7. Danh Sách Linh Kiện**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Linh Kiện** | **Số Lượng** | **Ghi Chú** |
| 1 | ESP32 NodeMCU | 1 | Vi điều khiển chính, xử lý logic. |
| 2 | Cảm biến DHT22 | 1 | Thu thập dữ liệu nhiệt độ và độ ẩm. |
| 3 | Module Relay (2 kênh) | 1 | Điều khiển thiết bị tải (phun sương/quạt). |
| 4 | Màn hình LCD 1602 I2C | 1 | Hiển thị giá trị đo và trạng thái hệ thống. |
| 5 | Dây nối (Jumper Wires) | Nhiều | Kết nối các linh kiện. |
| 6 | Nguồn qua cổng USB | 1 | Cấp nguồn ổn định cho ESP32 và các module 5V. |
| 7 | Breadboard | 1 | Sử dụng Breadboard làm cầu nối. |
| 8 | Dây cáp | 3 | Một dây truyền dữ liệu và 2 dây cung cấp nguồn điện 5v cho quạt và mạch phun sương |
| 10 | Quạt làm mát 6015DC 5v | 1 |  |
| 11 | Mạch phun sương DC | 1 |  |

Bảng : Danh Sách Linh Kiện

**3. Thiết Kế và Sơ Đồ Kết Nối Phần Cứng**

**3.1. Sơ Đồ Khối Hệ Thống**

Hệ thống được thiết kế theo kiến trúc module hóa, tập trung vào việc thu thập dữ liệu môi trường, xử lý logic tại vi điều khiển trung tâm và điều khiển các thiết bị chấp hành để duy trì điều kiện lý tưởng. Dữ liệu và trạng thái hệ thống được hiển thị cục bộ qua màn hình LCD và giám sát từ xa qua nền tảng IoT ThingSpeak và giao diện web Streamlit.

* **Khối Nguồn (Power Supply):** Nguồn 5V từ adapter/USB là nguồn năng lượng chính, được phân phối cho toàn bộ hệ thống thông qua breadboard.
* **Khối Cảm Biến (Sensor Block):** Cảm biến DHT22 có nhiệm vụ đo lường liên tục hai thông số quan trọng của môi trường là nhiệt độ () và độ ẩm ().
* **Khối Xử Lý Trung Tâm (Central Processing Unit):** Vi điều khiển ESP32 NodeMCU đóng vai trò là "bộ não" của hệ thống.
* **Khối Chấp Hành (Actuator Block):** Module Relay 2 kênh nhận tín hiệu logic từ ESP32 để đóng/ngắt nguồn 5V cho quạt làm mát và mạch phun sương.
* **Khối Hiển Thị và Giao Tiếp (Display & Communication):**
  + **Cục bộ:** Màn hình LCD 1602 I2C hiển thị tức thời nhiệt độ, độ ẩm và trạng thái hoạt động của các thiết bị.
  + **Từ xa:** Module WiFi tích hợp trên ESP32 gửi dữ liệu lên kênh **ThingSpeak** để lưu trữ. Dữ liệu này sau đó được truy vấn bởi một **giao diện web (Dashboard)** xây dựng bằng **Streamlit** để theo dõi trực quan.

**3.2. Chi Tiết Kết Nối Phần Cứng**

**Bảng dưới đây mô tả chi tiết cách kết nối chân của từng linh kiện trong hệ thống. Nguồn điện trung tâm được cấp vào breadboard và từ đó phân phối đến các module.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thiết Bị** | **Chân Thiết Bị** | **Nối với Chân ESP32** | **Chức Năng** |
| **DHT22** | VCC | Thanh 3.3V của Breadboard | Cấp nguồn ổn định và an toàn cho cảm biến. |
|  | DATA | GPIO 4 | Truyền/nhận dữ liệu số về nhiệt độ và độ ẩm. |
|  | GND | Thanh GND của Breadboard | Nối đất. |
| **Module Relay (2 kênh)** | VCC (DC+) | Thanh 5V của Breadboard | Cấp nguồn 5V cho cả hai cuộn dây của relay. |
|  | GND (DC-) | Thanh GND của Breadboard | Nối đất. |
|  | IN1 | GPIO 26 | Tín hiệu logic điều khiển Mạch phun sương. |
|  | IN2 | GPIO 27 | Tín hiệu logic điều khiển Quạt làm mát. |
| **LCD 1602 I2C** | VCC | Thanh 5V của Breadboard | Cấp nguồn cho màn hình và đèn nền. |
|  | GND | Thanh GND của Breadboard | Nối đất. |
|  | SDA | GPIO 21 | Kênh truyền dữ liệu của giao thức I2C. |
|  | SCL | GPIO 22 | Kênh xung nhịp của giao thức I2C. |
| **ESP32 NodeMCU** | VIN | Thanh 5V của Breadboard | Nhận nguồn 5V để nuôi toàn bộ bo mạch. |
|  | GND | Thanh GND của Breadboard | Nối đất chung cho toàn bộ hệ thống. |

Bảng : Kết Nối Phần Cứng

**4. Nguyên Lý Hoạt Động và Thuật Toán Điều Khiển**

**4.1. Quy Trình Thu Thập và Xử Lý Dữ Liệu**

Quy trình hoạt động của hệ thống được thực hiện tuần tự theo các bước sau trong mỗi chu kỳ lặp:

1. **Khởi tạo: Khi cấp nguồn, hệ thống thực hiện các tác vụ cài đặt ban đầu trong hàm setup():**
   * Khởi tạo giao tiếp Serial Monitor để gỡ lỗi.
   * Khởi tạo thư viện cho cảm biến DHT22 và màn hình LCD 1602 I2C.
   * Cấu hình hai chân GPIO 26 (điều khiển Bơm Phun Sương) và GPIO 27 (điều khiển Quạt) là chân OUTPUT.
   * Gửi tín hiệu LOW tới cả hai chân relay để đảm bảo các thiết bị chấp hành ở trạng thái TẮT ban đầu.
   * Hiển thị thông báo chào mừng lên màn hình LCD.
   * Thực hiện kết nối vào mạng WiFi đã được cấu hình.
2. **Vòng lặp chính: Hệ thống liên tục thực hiện các công việc sau trong hàm loop():**
   * Đọc cảm biến: ESP32 đọc giá trị Nhiệt độ () và Độ ẩm () từ cảm biến DHT22.
   * Kiểm tra lỗi: Dữ liệu đọc về được kiểm tra tính hợp lệ (isnan). Nếu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi lên LCD và bỏ qua chu kỳ hiện tại.
   * Xử lý logic: ESP32 so sánh giá trị nhiệt độ và độ ẩm vừa đọc được với hai ngưỡng đã định sẵn:
     + So sánh nhiệt độ với ngưỡng nhiệt độ (30°C).
     + So sánh độ ẩm với ngưỡng độ ẩm (80%).
   * Ra quyết định và điều khiển: Dựa trên kết quả so sánh, ESP32 gửi tín hiệu HIGH (Bật) hoặc LOW (Tắt) đến các chân GPIO 26 và 27 để điều khiển độc lập Bơm Phun Sương và Quạt.
   * Hiển thị cục bộ: Cập nhật màn hình LCD với các thông số , và trạng thái "ON/OFF" hiện tại của cả hai thiết bị.
   * Gửi dữ liệu từ xa: Sử dụng bộ đếm thời gian (millis()), hệ thống kiểm tra nếu đã đủ 20 giây kể từ lần gửi cuối cùng. Nếu đủ, nó sẽ đóng gói dữ liệu (, , trạng thái Bơm, trạng thái Quạt) và gửi lên nền tảng ThingSpeak.

**4.2. Thuật Toán Điều Khiển Tự Động (Điều Khiển Ngưỡng - On/Off Control)**

Hệ thống sử dụng thuật toán điều khiển ngưỡng (On-Off Control) đơn giản nhưng hiệu quả để duy trì môi trường. Logic điều khiển cho hai thiết bị là hoàn toàn độc lập với nhau.

* Trạng thái Relay: Module Relay được sử dụng là loại Active HIGH, có nghĩa là:
  + Tín hiệu HIGH = Bật Relay (ON).
  + Tín hiệu LOW = Tắt Relay (OFF).

Bảng 3: Thuật toán điều khiển Bơm Phun Sương

|  |  |  |  |
| --- | --- | --- | --- |
| Điều Kiện Độ Ẩm () | Hành Động | Trạng Thái GPIO 26 | Trạng Thái Bơm |
|  | Bật bơm phun sương | HIGH | ON (Bật) |
|  | Tắt bơm phun sương | LOW | OFF (Tắt) |

Xuất sang Trang tính

Bảng 4: Thuật toán điều khiển Quạt Làm Mát

|  |  |  |  |
| --- | --- | --- | --- |
| Điều Kiện Nhiệt Độ () | Hành Động | Trạng Thái GPIO 27 | Trạng Thái Quạt |
|  | Bật quạt làm mát | HIGH | ON (Bật) |
|  | Tắt quạt làm mát | LOW | OFF (Tắt) |

**5. Mã Chương Trình (Phiên bản Hoàn thiện)**

Đây là phiên bản mã nguồn hoàn chỉnh, đã tích hợp LCD I2C, logic IoT (ThingSpeak) và xử lý Relay theo đúng cấu hình phần cứng:

#include <WiFi.h>

#include <HTTPClient.h>

#include "DHT.h"

#include <LiquidCrystal\_I2C.h>

// --- Cấu hình người dùng ---

const char\* ssid             = "Nha Pun Beo 1"; // Tên WiFi của bạn

const char\* password         = "@0849255588\*";   // Mật khẩu WiFi

const char\* thingspeakApiKey = "NVUWCM7W1AY4NQZY"; // Write API Key của bạn

const char\* thingspeakHost   = "https://api.thingspeak.com/update";

// --- Cấu hình cảm biến ---

#define DHTPIN 4

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

// --- Cấu hình Relay ---

#define PUMP\_RELAY\_PIN 26 // Kênh 1 - Điều khiển máy phun sương

#define FAN\_RELAY\_PIN  27 // Kênh 2 - Điều khiển quạt

// --- Ngưỡng điều khiển ---

const float HUMIDITY\_THRESHOLD = 80.0; // Bật phun sương nếu độ ẩm < 80%

const float TEMP\_THRESHOLD     = 30.0; // Bật quạt nếu nhiệt độ > 30°C

// --- Cấu hình LCD1602 I2C ---

LiquidCrystal\_I2C lcd(0x27, 16, 2);

// --- Cấu hình gửi dữ liệu ---

const unsigned long SEND\_INTERVAL = 20000UL; // Gửi mỗi 20 giây

unsigned long lastSend = 0;

void setup() {

  Serial.begin(115200);

  // Khởi tạo LCD

  lcd.init();

  lcd.backlight();

  lcd.setCursor(0, 0);

  lcd.print("He thong");

  lcd.setCursor(0, 1);

  lcd.print("Dang khoi dong...");

  dht.begin();

  pinMode(PUMP\_RELAY\_PIN, OUTPUT);

  pinMode(FAN\_RELAY\_PIN, OUTPUT);

  // Tắt cả hai relay khi khởi động (Active HIGH)

  digitalWrite(PUMP\_RELAY\_PIN, LOW);

  digitalWrite(FAN\_RELAY\_PIN, LOW);

  delay(2000);

  // Kết nối WiFi

  connectWiFi();

}

void connectWiFi() {

  Serial.printf("Dang ket noi WiFi: %s\n", ssid);

  WiFi.begin(ssid, password);

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("Connecting WiFi");

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

    delay(500);

    Serial.print(".");

    lcd.setCursor(0, 1);

    lcd.print("...Dang thu lai...");

  }

  Serial.printf("\nDa ket noi! IP: %s\n", WiFi.localIP().toString().c\_str());

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("WiFi Da Ket Noi!");

  lcd.setCursor(0, 1);

  lcd.print(WiFi.localIP().toString());

  delay(2000);

}

void loop() {

  // Đọc dữ liệu từ cảm biến

  float humidity = dht.readHumidity();

  float temperature = dht.readTemperature();

  if (isnan(humidity) || isnan(temperature)) {

    Serial.println("Loi doc DHT22!");

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("LOI CAM BIEN");

    delay(2000);

    return;

  }

  // Logic điều khiển

  bool pumpStatus = (humidity < HUMIDITY\_THRESHOLD);

  bool fanStatus  = (temperature > TEMP\_THRESHOLD);

  digitalWrite(PUMP\_RELAY\_PIN, pumpStatus ? HIGH : LOW);

  digitalWrite(FAN\_RELAY\_PIN, fanStatus ? HIGH : LOW);

  // Cập nhật màn hình LCD

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("T:");

  lcd.print(temperature, 1);

  lcd.print((char)223);

  lcd.print("C");

  lcd.setCursor(8, 0);

  lcd.print(" H:");

  lcd.print(humidity, 1);

  lcd.print("%");

  lcd.setCursor(0, 1);

  lcd.print("PUMP:");

  lcd.print(pumpStatus ? "ON " : "OFF");

  lcd.setCursor(8, 1);

  lcd.print(" FAN:");

  lcd.print(fanStatus ? "ON" : "OFF");

  // Kiểm tra và gửi dữ liệu lên ThingSpeak

  if (millis() - lastSend >= SEND\_INTERVAL) {

    lastSend = millis();

    Serial.println("Dang chuan bi gui du lieu...");

    // Kiểm tra lại WiFi trước khi gửi

    if(WiFi.status() == WL\_CONNECTED){

      sendToThingSpeak(humidity, temperature, pumpStatus, fanStatus);

    } else {

      Serial.println("Mat ket noi WiFi, dang thu ket noi lai...");

      connectWiFi();

    }

  }

  delay(2000);

}

void sendToThingSpeak(float temp, float hum, bool pump, bool fan) {

    HTTPClient http;

    // THAY ĐỔI: Thêm field4 cho trạng thái quạt

    String url = String(thingspeakHost) + "?api\_key=" + thingspeakApiKey +

                 "&field1=" + String(temp) +

                 "&field2=" + String(hum) +

                 "&field3=" + String(pump ? 1 : 0) +

                 "&field4=" + String(fan ? 1 : 0);

    http.begin(url.c\_str());

    int httpCode = http.GET();

    if (httpCode == 200) {

        Serial.println("=> Gui du lieu thanh cong!");

    } else {

        Serial.printf("=> Loi gui du lieu, HTTP Code: %d\n", httpCode);

    }

    http.end();

}

**6. Kết Quả Thực Nghiệm và Phân Tích**

**6.1. Quá Trình Thực Nghiệm**

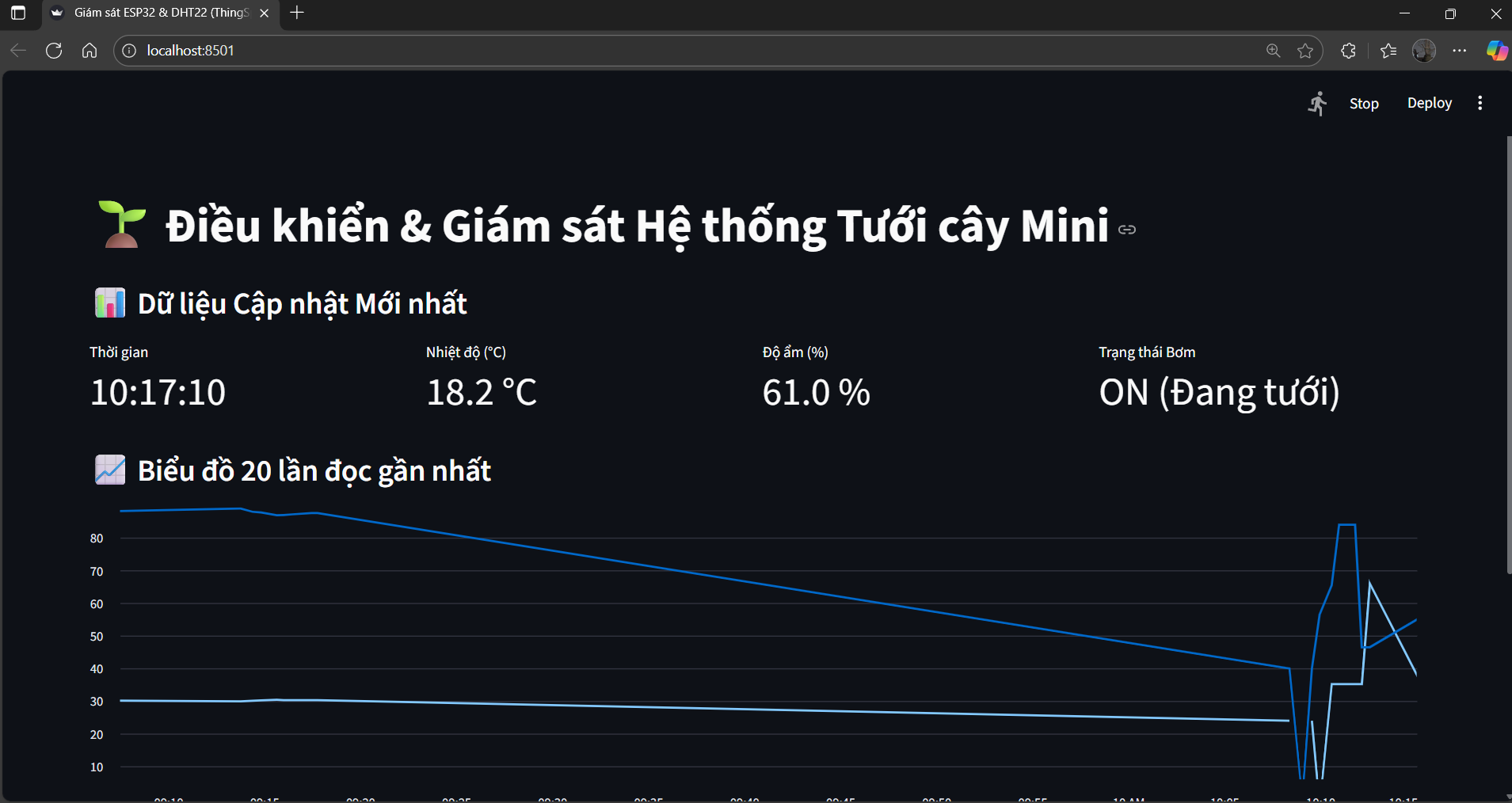
Hệ thống được lắp ráp theo sơ đồ kết nối và nạp chương trình. Quá trình kiểm tra được thực hiện bằng cách tạo ra sự thay đổi nhiệt độ đột ngột tại cảm biến (ví dụ: dùng hơi ấm) và kiểm tra khả năng phản ứng của hệ thống. Đồng thời, kiểm tra khả năng hiển thị và gửi dữ liệu.

**6.2. Kết Quả Thu Được** **Chức năng đo lường và hiển thị:** Hệ thống đã đọc chính xác và liên tục giá trị nhiệt độ và độ ẩm từ DHT22. Các giá trị này được hiển thị đồng thời trên Serial Monitor và màn hình LCD 1602 I2C, giúp người dùng dễ dàng theo dõi trạng thái cục bộ. **Chức năng điều khiển tự động:** Hệ thống phản ứng chính xác theo thuật toán đã lập trình:

* Khi **độ ẩm** đo được **dưới 80%**, chân **GPIO 26** chuyển sang mức **HIGH**, kích hoạt Relay kênh 1 và **bật (ON)** mạch phun sương. Khi độ ẩm từ 80% trở lên, relay được tắt (OFF).
* Khi **nhiệt độ** đo được **vượt quá 30°C**, chân **GPIO 27** chuyển sang mức **HIGH**, kích hoạt Relay kênh 2 và **bật (ON)** quạt làm mát. Khi nhiệt độ từ 30°C trở xuống, relay được tắt (OFF).

**Chức năng IoT và Giám sát từ xa:**

* Hệ thống kết nối WiFi thành công và gửi dữ liệu (nhiệt độ, độ ẩm, trạng thái bơm, trạng thái quạt) lên kênh ThingSpeak mỗi **20 giây**.
* Giao diện Streamlit Dashboard được triển khai thành công, truy vấn dữ liệu từ ThingSpeak và hiển thị các thông số một cách trực quan, đáp ứng yêu cầu giám sát từ xa qua trình duyệt web trên cả máy tính và điện thoại.



Hình : Giao diện trên máy tính



Hình : Giao diện trên điện thoại

**6.3. Đánh Giá Hiệu Suất**

Hệ thống hoạt động ổn định và đáp ứng đúng các mục tiêu đề ra. Việc tích hợp LCD I2C đã hoàn thành mục tiêu về giao diện người dùng cục bộ. Độ chính xác điều khiển phụ thuộc vào độ chính xác của cảm biến DHT22. Hệ thống được cấu hình gửi dữ liệu mỗi **20 giây**, phù hợp với giới hạn của nền tảng ThingSpeak miễn phí, đảm bảo khả năng giám sát gần thời gian thực.

**7. Kết Luận và Hướng Phát Triển**

**7.1. Các Kết Quả Đạt Được**

* Hoàn thành hệ thống nhúng: Thiết kế và triển khai thành công hệ thống giám sát và điều khiển tự động sử dụng ESP32.
* Tích hợp đa giao tiếp: Nắm vững cách thức giao tiếp 1-Wire (DHT22) và I2C (LCD 1602), cũng như giao tiếp mạng (WiFi/HTTP).
* Triển khai IoT: Thiết lập thành công kết nối và gửi dữ liệu lên nền tảng ThingSpeak, cho phép giám sát dữ liệu từ xa.
* Xây dựng Giao diện Web Giám sát (Streamlit):
  + Công nghệ: Python/Streamlit và thư viện requests.
  + Nguyên lý: Dashboard sử dụng Read API Key của ThingSpeak để truy vấn dữ liệu theo thời gian thực.
  + Tần suất cập nhật: Dashboard được cấu hình tự động load lại API sau mỗi 5 giây để đảm bảo dữ liệu hiển thị có độ trễ thấp nhất.
  + Chức năng hiển thị: Biểu đồ xu hướng (Nhiệt độ, Độ ẩm) và các Metric (thẻ) hiển thị giá trị mới nhất và trạng thái Bơm (ON/OFF) bằng giờ Việt Nam (UTC+7).

**7.2. Hướng Phát Triển và Mở Rộng**

* Nâng cao thuật toán điều khiển: Áp dụng Hysteresis (ví dụ: Bật ở , Tắt ở ) để tránh hiện tượng Relay chớp nháy liên tục (chattering), bảo vệ tuổi thọ thiết bị.
* Điều khiển từ xa: Sử dụng ThingSpeak hoặc Blynk để nhận lệnh từ Cloud, cho phép người dùng điều khiển Relay bằng tay (Override) từ xa.
* Cảnh báo: Thêm chức năng gửi cảnh báo qua Email/Telegram khi nhiệt độ vượt quá ngưỡng nghiêm trọng hoặc khi cảm biến báo lỗi.