

TRƯỜNG ĐẠI HỌC BÁCH KHOA

.....



ĐỒ ÁN 1

Đề tài:

**THIẾT KẾ PHẦN MỀM ĐIỀU KHIỂN TỰ ĐỘNG
VÀ GIÁM SÁT NHIỆT ĐỘ, ĐỘ ẨM**

Giảng viên hướng dẫn: TS.Trần Ngọc Tuấn

Sinh viên thực hiện: Bùi Đình Phong 20234030

Lớp: ET-E9

mục lục

chương 1. tổng quan đề tài

- 1.1. lý do chọn đề tài
- 1.2. mục tiêu của đề tài
- 1.3. phạm vi và đối tượng nghiên cứu

chương 2. cơ sở lý thuyết

- 2.1. tổng quan về hệ thống điều khiển tự động
- 2.2. lấy mẫu dữ liệu trong hệ thống nhúng
- 2.3. phương pháp lọc nhiễu số
- 2.4. nguyên lý điều khiển on/off
- 2.5. nền tảng iot và blynk

chương 3. phân tích và thiết kế hệ thống phần mềm

- 3.1. phân tích yêu cầu hệ thống
- 3.2. xác định đầu vào – đầu ra của hệ thống
- 3.3. thiết kế kiến trúc phần mềm
- 3.4. thiết kế thuật toán điều khiển
- 3.5. thiết kế thuật toán lọc nhiễu dữ liệu
- 3.6. sơ đồ khối hệ thống

chương 4. lưu đồ thuật toán và giải thuật

- 4.1. Xác định đầu vào – đầu ra của hệ thống
- 4.2. Mô tả thuật toán điều khiển

thực nghiệm và đánh giá

- 6.1. mô hình kiểm thử hệ thống
- 6.2. kết quả thực nghiệm
- 6.3. đánh giá hoạt động của phần mềm
- 6.4. nhận xét và thảo luận

kết luận

CHƯƠNG 1. GIỚI THIỆU

1.1.Lý do chọn đề tài

Trong những năm gần đây, sự phát triển mạnh mẽ của lĩnh vực Internet of Things (IoT) đã mở ra nhiều hướng ứng dụng mới trong giám sát và điều khiển tự động. Các hệ thống đo và điều khiển nhiệt độ – độ ẩm được sử dụng rộng rãi trong nhà ở, nhà kính, kho bảo quản và công nghiệp nhẹ. Việc nghiên cứu và xây dựng một hệ thống điều khiển tự động đơn giản giúp sinh viên nắm vững kiến thức cơ bản của môn Điều khiển tự động.

1.2.Nội dung thực hiện

Đồ án tập trung vào việc thiết kế phần mềm nhúng và triển khai hệ thống đo nhiệt độ – độ ẩm sử dụng ESP32 và cảm biến DHT11, áp dụng lấy mẫu rời rạc, lọc nhiễu dữ liệu và điều khiển tự động ON/OFF. Trạng thái hệ thống được hiển thị thông qua nền tảng Blynk.

1.3.Phạm vi đề tài

- Đo và giám sát nhiệt độ – độ ẩm môi trường.
- Điều khiển LED thay thế cho quạt.
- Ứng dụng điều khiển tự động cơ bản, chưa xét PID.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Hệ thống điều khiển tự động

Hệ thống điều khiển tự động bao gồm các khối cơ bản: đối tượng điều khiển, cảm biến, bộ điều khiển và cơ cấu chấp hành. Trong đồ án này, môi trường là đối tượng điều khiển, DHT11 là cảm biến, ESP32 là bộ điều khiển và LED là cơ cấu chấp hành.

2.2. Lấy mẫu rời rạc

Lấy mẫu là quá trình chuyển tín hiệu liên tục thành tín hiệu rời rạc theo thời gian. Chu kỳ lấy mẫu T_s ảnh hưởng trực tiếp đến độ chính xác và độ ổn định của hệ thống. Với cảm biến DHT11, T_s được chọn là 2 giây để đảm bảo cảm biến hoạt động ổn định.

2.3. Nhiễu và lọc nhiễu

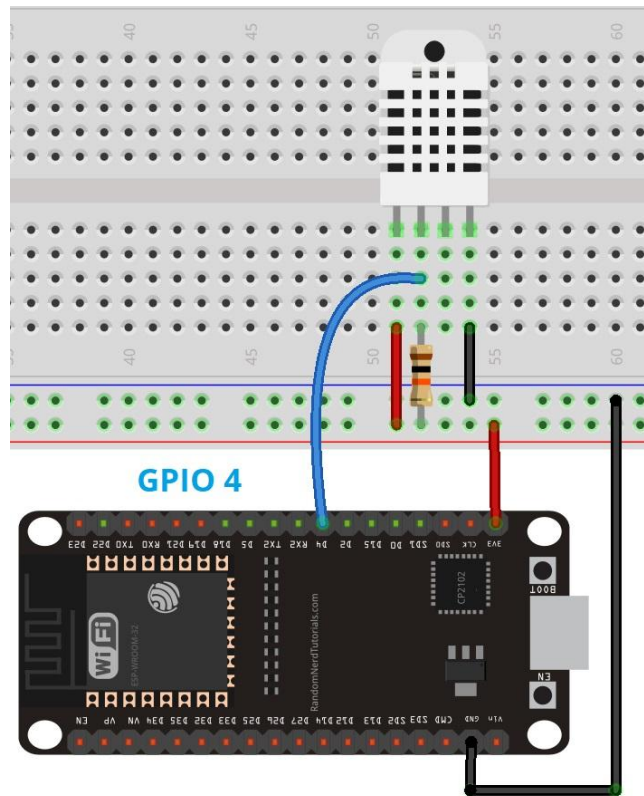
Tín hiệu đo từ cảm biến thực tế thường chứa nhiễu do môi trường và sai số linh kiện. Để cải thiện chất lượng tín hiệu, hệ thống sử dụng bộ lọc thông thấp số bậc một nhằm làm mượt dữ liệu đo.

2.4. Điều khiển ON/OFF

Điều khiển ON/OFF là phương pháp điều khiển đơn giản nhất, trong đó tín hiệu điều khiển chỉ có hai trạng thái. Khi giá trị đo vượt ngưỡng đặt, cơ cấu chấp hành được bật, ngược lại sẽ tắt.

CHƯƠNG 3. THIẾT KẾ HỆ THỐNG

3.1. Sơ đồ kết nối



- DHT11 VCC → 3.3V ESP32
- DHT11 GND → GND ESP32
- DHT11 DATA → GPIO4
- LED (+) → GPIO27 qua điện trở 220Ω
- LED (-) → GND

CHƯƠNG 4. THIẾT KẾ PHẦN MỀM

4.1. Xác định đầu vào – đầu ra của hệ thống

Đầu vào (Input)

- Giá trị nhiệt độ $T(k)$ từ cảm biến DHT11
- Giá trị độ ẩm $H(k)$ từ cảm biến DHT11
- Thời gian lấy mẫu $T_s = 2$ giây
- Ngưỡng nhiệt độ cài đặt $T_{set} = 30^{\circ}\text{C}$

Đầu ra (Output)

- Trạng thái LED (ON/OFF) – mô phỏng quạt
- Giá trị nhiệt độ sau lọc $T_f(k)$ hiển thị trên Blynk
- Giá trị độ ẩm $H(k)$ hiển thị trên Blynk

4.2. Mô tả thuật toán điều khiển

Thuật toán điều khiển được thực hiện theo chu trình lặp định kỳ, bao gồm các bước chính sau:

1. Khởi tạo hệ thống
2. Kết nối WiFi và Blynk
3. Lấy mẫu dữ liệu nhiệt độ – độ ẩm
4. Lọc nhiễu tín hiệu nhiệt độ
5. So sánh nhiệt độ với ngưỡng
6. Điều khiển LED theo luật ON/OFF
7. Gửi dữ liệu lên Blynk
8. Chờ chu kỳ lấy mẫu tiếp theo

4.3. Phân tích chương trình điều khiển

1. chỉ thị tiền xử lý (preprocessor directive)

```
1  #define BLYNK_PRINT Serial]
2  #define BLYNK_TEMPLATE_ID "TMPL6I6L4nwQj"
3  #define BLYNK_TEMPLATE_NAME "dht11"
4  #define BLYNK_AUTH_TOKEN "MJ2xp0-cCS0Ma9v-ELYkP1tpH1VnMh88"
```

dòng lệnh này cho phép in các thông tin trạng thái của blynk ra serial monitor nhằm phục vụ việc theo dõi và gỡ lỗi chương trình trong quá trình vận hành.

các tham số trên dùng để định danh thiết bị trên nền tảng blynk, giúp esp32 xác thực và gửi dữ liệu lên đúng dashboard đã tạo.

2. khai báo thư viện sử dụng

```
6  #include <WiFi.h>
7  #include <BlynkSimpleEsp32.h>
8  #include <DHT.h>
```

- thư viện WiFi.h dùng để kết nối mạng internet
- thư viện BlynkSimpleEsp32.h dùng để giao tiếp với nền tảng iot blynk
- thư viện DHT.h dùng để đọc dữ liệu từ cảm biến dht11

3. khai báo thông số mạng

```
10 char ssid[] = "MT7628_AP";
11 char pass[] = "12345678";
```

khai báo tên và mật khẩu wifi để thiết bị esp32 có thể kết nối internet và truyền dữ liệu.

4. khai báo chân kết nối và thiết bị

```
13 #define DHTPIN 4
14 #define DHTTYPE DHT11
15 #define LED_PIN 27
16
17 DHT dht(DHTPIN, DHTTYPE);
18 BlynkTimer timer;
```

- DHTPIN là chân tín hiệu của cảm biến dht11
- DHTTYPE xác định loại cảm biến được sử dụng
- LED_PIN là chân điều khiển đèn led

5. khai báo đối tượng và biến toàn cục

```
21 float alpha = 0.2;    // hệ số lọc
22 float t_f = 0;        // nhiệt độ sau lọc
23 float setTemp = 30.0; // ngưỡng bật LED
```

- alpha là hệ số lọc nhiễu
- t_f là nhiệt độ sau khi lọc
- setTemp là giá trị nhiệt độ ngưỡng để điều khiển

6. hàm đọc dữ liệu và xử lý chính

```
25 void sendDHT() {
26     float h = dht.readHumidity();
27     float t = dht.readTemperature();
28
29     if (isnan(h) || isnan(t)) {
30         Serial.println("Loi doc DHT11");
31         return;
32     }
```

hàm này thực hiện việc đọc dữ liệu đầu vào từ cảm biến dht11 gồm nhiệt độ và độ ẩm, xử lý và điều khiển, được gọi theo chu kỳ định trước.

kiểm tra dữ liệu có hợp lệ hay không, nếu có lỗi thì bỏ qua chu kỳ lấy mẫu hiện tại nhằm đảm bảo an toàn cho hệ thống.

7. thuật toán lọc nhiễu dữ liệu

```
35 | t_f = alpha * t + (1 - alpha) * t_f;
```

8. thuật toán điều khiển tự động

```
37 // ---- ĐIỀU KHIỂN TỰ ĐỘNG ----
38 if (t_f > setTemp) {
39     digitalWrite(LED_PIN, HIGH);
40     Blynk.virtualWrite(V3, 1);
41 } else {
42     digitalWrite(LED_PIN, LOW);
43     Blynk.virtualWrite(V3, 0);
44 }
```

nếu nhiệt độ sau lọc vượt quá ngưỡng cài đặt thì bật led, ngược lại thì tắt led.

9. gửi dữ liệu lên nền tảng blynk


```

46 // ---- GỬI LÊN BLYNK ----
47 Blynk.virtualWrite(V1, t_f);
48 Blynk.virtualWrite(V2, h);

```

10. hàm khởi tạo hệ thống

```

59 void setup() {
60     Serial.begin(115200);
61
62     pinMode(LED_PIN, OUTPUT);
63     digitalWrite(LED_PIN, LOW);
64
65     dht.begin();
66     Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
67
68     // LẤY MẪU 2 GIÂY
69     timer.setInterval(2000L, sendDHT);
70 }

```

khởi tạo cổng serial để giám sát hoạt động của hệ thống.

cấu hình chân điều khiển led là ngõ ra.

khởi tạo cảm biến dht11.

kết nối thiết bị với wifi và nền tảng blynk.

thiết lập chu kỳ lấy mẫu dữ liệu là 2 giây.

11. vòng lặp chính của chương trình

```

72 void loop() {
73     Blynk.run();
74     timer.run();

```

chương trình hoạt động theo cơ chế không chặn, đảm bảo duy trì kết nối iot và thực hiện các tác vụ đúng thời gian.

KẾT QUẢ THỰC NGHIỆM

Hệ thống hoạt động ổn định sau khi hiệu chỉnh phần cứng. LED phản hồi đúng với sự thay đổi nhiệt độ.

KẾT LUẬN

Trong khuôn khổ đề án thiết kế phần mềm, tôi đã thực hiện xây dựng và triển khai thành công một hệ thống phần mềm điều khiển và giám sát nhiệt độ, độ ẩm dựa trên nền tảng vi điều khiển esp32, sử dụng cảm biến dht11 và nền tảng iot blynk. Mục tiêu chính của đề án là thiết kế thuật toán và chương trình điều khiển, trong đó phần cứng chỉ đóng vai trò môi trường kiểm thử cho phần mềm.

Trong quá trình thực hiện, tôi đã thiết kế cấu trúc chương trình theo hướng mô-đun, bao gồm các chức năng: thu thập dữ liệu cảm biến, xử lý và lọc nhiễu dữ liệu, thực hiện thuật toán điều khiển on/off và truyền dữ liệu lên nền tảng giám sát từ xa. Việc áp dụng bộ lọc thông thấp số giúp dữ liệu nhiệt độ trở nên ổn định hơn, hạn chế ảnh hưởng của nhiễu và sai số cảm biến.

Phần mềm được xây dựng theo cơ chế lấy mẫu định kỳ kết hợp với bộ định thời phần mềm, đảm bảo chương trình hoạt động liên tục, không bị chặn và phù hợp với yêu cầu của hệ thống điều khiển nhúng đơn giản. Bên cạnh đó, tôi cũng đã tích hợp cơ chế kiểm tra dữ liệu không hợp lệ từ cảm biến nhằm nâng cao độ an toàn và độ tin cậy của hệ thống.

Kết quả thực nghiệm cho thấy chương trình hoạt động đúng theo yêu cầu đề ra: dữ liệu nhiệt độ và độ ẩm được thu thập và hiển thị ổn định; cơ cấu chấp hành (led mô phỏng quạt) được điều khiển tự động theo ngưỡng cài đặt. Điều này chứng tỏ giải pháp phần mềm mà tôi xây dựng là khả thi và có thể áp dụng trong thực tế.

Tuy nhiên, do giới hạn về thời gian và phạm vi nghiên cứu, hệ thống vẫn còn một số hạn chế như thuật toán điều khiển còn đơn giản và chưa có khả năng tự thích nghi với sự thay đổi của môi trường. Trong tương lai, tôi định hướng phát triển hệ thống theo hướng mở rộng thuật toán điều khiển thông minh hơn, nâng cao khả năng xử lý dữ liệu và cải thiện giao diện giám sát.

Nhìn chung, thông qua đề án này, tôi đã củng cố và nâng cao được kiến thức về lập trình nhúng, thiết kế phần mềm điều khiển và ứng dụng iot trong lĩnh vực điều khiển tự động.