



BÀI TẬP LỚN

IOT VÀ ỨNG DỤNG

Đề tài:

Xây dựng hệ thống giám sát nhiệt độ và độ ẩm dùng ESP32, cảm biến DHT11 và nền tảng ThingSpeak

Tên học phần: IoT và ứng dụng

Nhóm lớp học phần: INT14149

Nhóm làm đề tài: 11

Sinh viên tham gia thực hiện:

1. Trần Duy Mạnh Mã số: B24DTCN004 Lớp: D24TXCN01-B
2. Bùi Anh Tuấn Mã số: B17DCCN647 Lớp: D24TXCN01-B
3. Tạ Thu Hà Mã số: B24DTCN003 Lớp: D24TXCN01-B
4. Trần Anh Thái Mã số: K23DTCN200 Lớp: D23TXCN02-K

Giáo viên hướng dẫn: Nguyễn Đức Minh

BÁO CÁO BÀI TẬP LỚN IOT VÀ ỨNG DỤNG

1. Mục đích và yêu cầu hệ thống

1.1. Mục đích

Đề tài xây dựng một hệ thống IoT đơn giản dùng ESP32 và cảm biến DHT11 để giám sát nhiệt độ, độ ẩm môi trường. Dữ liệu sau khi thu thập được gửi lên nền tảng ThingSpeak và hiển thị trực quan trên web bằng nhiều dạng biểu đồ khác nhau như line chart, bar chart, pie chart và doughnut chart.

Hệ thống có thể ứng dụng trong phòng học, phòng làm việc, phòng máy tính, nhà thông minh hoặc khu vực cần theo dõi nhiệt độ và độ ẩm thường xuyên.

1.2. Yêu cầu chức năng

- Sử dụng ESP32 làm bộ xử lý trung tâm.
- Sử dụng cảm biến DHT11 để thu thập hai đại lượng môi trường: nhiệt độ và độ ẩm.
- Gửi dữ liệu lên ThingSpeak thông qua WiFi.
- Hiển thị dữ liệu cảm biến trên web bằng nhiều dạng biểu đồ.
- Cảnh báo khi nhiệt độ vượt ngưỡng cấu hình, ví dụ 35 độ C.
- Cho phép demo bằng dữ liệu mẫu khi chưa có phần cứng hoặc chưa có channel ThingSpeak thật.

1.3. Yêu cầu phi chức năng

- Hệ thống có chi phí thấp, dễ lắp đặt với bộ kit cơ bản có DHT11.
- Chu kỳ gửi dữ liệu lên ThingSpeak khoảng 20 giây/lần để phù hợp giới hạn nền tảng.
- Web dashboard dễ quan sát, chạy được trên trình duyệt.
- Cấu trúc chương trình rõ ràng, dễ mở rộng thêm cảm biến hoặc cảnh báo.

2. Đặc tả quy trình

Quy trình hoạt động của hệ thống được mô tả như sau:

1. ESP32 khởi động và kết nối vào mạng WiFi.
2. Cảm biến DHT11 đo nhiệt độ và độ ẩm môi trường.
3. ESP32 đọc dữ liệu từ chân DATA của DHT11.
4. ESP32 kiểm tra dữ liệu hợp lệ, loại bỏ trường hợp cảm biến đọc lỗi.
5. ESP32 so sánh nhiệt độ với ngưỡng cảnh báo, ví dụ 35 độ C.
6. Nếu nhiệt độ vượt ngưỡng, hệ thống bật LED cảnh báo và đặt `alert_state = 1`.
7. ESP32 gửi dữ liệu nhiệt độ, độ ẩm, trạng thái cảnh báo lên ThingSpeak.
8. Web dashboard gọi API của ThingSpeak để lấy dữ liệu.
9. Dashboard xử lý dữ liệu và hiển thị bằng nhiều biểu đồ.
10. Người dùng theo dõi trạng thái môi trường qua trình duyệt.

3. Đặc tả mô hình miền

Thực thể	Vai trò
Người dùng	Theo dõi nhiệt độ, độ ẩm và trạng thái cảnh báo trên web
Môi trường phòng	Không gian cần giám sát nhiệt độ và độ ẩm
Cảm biến DHT11	Đo nhiệt độ và độ ẩm
ESP32	Thu thập dữ liệu, xử lý cảnh báo và gửi dữ liệu lên cloud
ThingSpeak	Nền tảng IoT lưu trữ dữ liệu theo thời gian
Web dashboard	Giao diện trực quan hóa dữ liệu cảm biến
LED cảnh báo	Thiết bị báo trạng thái nhiệt độ cao tại chỗ

DHT11 được kết nối trực tiếp với ESP32. ESP32 đọc dữ liệu từ DHT11, xử lý trạng thái cảnh báo, gửi dữ liệu lên ThingSpeak. Người dùng truy cập dashboard web để xem dữ liệu đã được trực quan hóa.

4. Đặc tả mô hình thông tin

Channel ThingSpeak được cấu hình với 3 field:

Field	Tên dữ liệu	Kiểu dữ liệu	Ý nghĩa
Field 1	temperature_c	Float	Nhiệt độ đo được, đơn vị độ C
Field 2	humidity_percent	Float	Độ ẩm tương đối, đơn vị phần trăm
Field 3	alert_state	Integer	0: bình thường, 1: cảnh báo nhiệt độ cao

Ví dụ một bản ghi dữ liệu:

```
{
  "created_at": "2026-10-03T10:30:00Z",
  "field1": "31.5",
  "field2": "68.0",
  "field3": "0"
}
```

5. Đặc tả dịch vụ

Dịch vụ	Loại	Đầu vào	Đầu ra	Mô tả
Đọc DHT11	Thiết bị	Tín hiệu DATA từ DHT11	Nhiệt độ, độ ẩm	ESP32 đọc dữ liệu cảm biến
Xử lý cảnh báo	Cục bộ	Nhiệt độ hiện tại	alert_state	So sánh với ngưỡng nhiệt độ cao
Gửi dữ liệu	REST/HTTP	Write API Key, field1, field2, field3	Entry ID	ESP32 gửi dữ liệu lên ThingSpeak

Lấy dữ liệu	REST/HTTP	Channel ID, API Key	Read JSON feeds	Web dashboard đọc dữ liệu từ ThingSpeak
Trực quan hóa	Web	Dữ liệu JSON	Biểu đồ	Hiển thị line, bar, pie, doughnut chart

Endpoint gửi dữ liệu lên ThingSpeak:

```
https://api.thingspeak.com/update?
api_key=WRITE_API_KEY&field1=31.5&field2=68.0&field3=0
```

Endpoint đọc dữ liệu từ ThingSpeak:

```
https://api.thingspeak.com/channels/CHANNEL_ID/feeds.json?
api_key=READ_API_KEY&results=100
```

6. Đặc tả cấp độ IoT

Hệ thống thuộc mô hình IoT trong đó thiết bị cảm biến kết nối trực tiếp lên cloud:

- Thiết bị IoT: ESP32 và cảm biến DHT11.
- Kết nối mạng: WiFi.
- Nền tảng IoT: ThingSpeak.
- Ứng dụng: Web dashboard hiển thị dữ liệu.

ESP32 không cần máy chủ trung gian cục bộ. Thiết bị gửi dữ liệu trực tiếp lên ThingSpeak, sau đó web dashboard lấy dữ liệu từ cloud để hiển thị cho người dùng.

7. Đặc tả chế độ xem chức năng

Nhóm chức năng	Chức năng
Thu nhận dữ liệu	Đọc nhiệt độ và độ ẩm từ DHT11
Xử lý dữ liệu	Kiểm tra dữ liệu hợp lệ, xác định cảnh báo nhiệt độ cao

Truyền thông	Kết nối WiFi và gửi HTTP request lên ThingSpeak
Lưu trữ	ThingSpeak lưu dữ liệu theo thời gian
Trực quan hóa	Dashboard vẽ line chart, bar chart, pie chart, doughnut chart
Cảnh báo	LED bật khi nhiệt độ vượt ngưỡng

8. Đặc tả chế độ xem hoạt động

8.1. Thiết bị sử dụng

- ESP32 DevKit.
- Cảm biến nhiệt độ, độ ẩm DHT11.
- LED cảnh báo.
- Dây nối, breadboard, nguồn USB.

8.2. Sơ đồ nối chân

Linh kiện		Chân linh kiện	Chân ESP32
DHT11		VCC	3.3V hoặc 5V tùy module
DHT11		GND	GND
DHT11		DATA	GPIO 4
LED cảnh báo		Anode	GPIO 2 qua điện trở hạn dòng
LED cảnh báo		Cathode	GND

8.3. Phần mềm sử dụng

- Arduino IDE để lập trình ESP32.
- Thư viện DHT sensor library và Adafruit Unified Sensor.

- ThingSpeak để lưu dữ liệu IoT.
- HTML, CSS, JavaScript và Chart.js để xây dựng dashboard.

9. Tích hợp thiết bị và thành phần

1. Kết nối DHT11 với ESP32 theo sơ đồ chân.
2. Kết nối LED cảnh báo với GPIO 2.
3. Tạo channel ThingSpeak tên **DHT11 ESP32 Monitor**.
4. Bật 3 field: temperature_c, humidity_percent, alert_state.
5. Lấy Write API Key và điền vào chương trình ESP32.
6. Nạp code vào ESP32 bằng Arduino IDE.
7. Mở Serial Monitor tốc độ 115200 để kiểm tra dữ liệu.
8. Kiểm tra dữ liệu xuất hiện trên ThingSpeak.
9. Mở file dashboard, nhập Channel ID và Read API Key để xem biểu đồ.

10. Phát triển ứng dụng

10.1. Chương trình ESP32

Chương trình ESP32 thực hiện các nhiệm vụ: kết nối WiFi, khởi tạo DHT11, đọc nhiệt độ và độ ẩm, kiểm tra nhiệt độ vượt ngưỡng, bật LED cảnh báo, gửi dữ liệu lên ThingSpeak theo chu kỳ 20 giây.

```
#include <WiFi.h>
#include <HTTPClient.h>
#include "DHT.h"

#define DHT_PIN 4
#define DHT_TYPE DHT11

Field 1: temperature_c
Field 2: humidity_percent
Field 3: alert_state
```

10.2. Web dashboard

Web dashboard đọc dữ liệu từ API ThingSpeak và hiển thị các thông tin:

- Nhiệt độ hiện tại.
- Độ ẩm hiện tại.
- Trạng thái cảnh báo.
- Thời điểm cập nhật gần nhất.

Các biểu đồ được sử dụng gồm:

- Line chart: biểu diễn nhiệt độ và độ ẩm theo thời gian.
- Bar chart: biểu diễn 12 mẫu nhiệt độ gần nhất.
- Pie chart: tỷ lệ trạng thái bình thường và cảnh báo.
- Doughnut chart: phân bố nhiệt độ thấp, trung bình, cao.

Kết quả thực nghiệm dự kiến

Khi hệ thống hoạt động, ESP32 đọc được nhiệt độ và độ ẩm từ DHT11. Ví dụ, nếu nhiệt độ là 31.5 độ C và độ ẩm là 68%, dữ liệu sẽ được gửi lên ThingSpeak vào field1 và field2. Nếu nhiệt độ lớn hơn hoặc bằng 35 độ C, field3 được đặt bằng 1 và LED cảnh báo bật.

Trên dashboard, người dùng có thể quan sát xu hướng nhiệt độ, độ ẩm theo thời gian, số mẫu cảnh báo và tỷ lệ các mức nhiệt độ. Điều này giúp người dùng đánh giá điều kiện môi trường trong phòng một cách trực quan.

Kết luận

Bài thực nghiệm đã xây dựng được một hệ thống IoT đơn giản dùng ESP32 và DHT11. Hệ thống đáp ứng yêu cầu thu thập dữ liệu cảm biến, gửi dữ liệu lên nền tảng ThingSpeak và hiển thị dữ liệu trên web bằng nhiều loại biểu đồ. Đề tài có thể mở rộng thêm cảm biến ánh sáng, cảm biến khí gas, còi cảnh báo, gửi thông báo Telegram hoặc điều khiển quạt tự động khi nhiệt độ vượt ngưỡng.

Phụ lục: Cấu hình ThingSpeak

Cấu hình channel đề xuất:

Thông tin	Giá trị	
Channel name	DHT11	ESP32 Monitor
Field 1	temperature_c	
Field 2	humidity_percent	
Field 3	alert_state	