

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN

_____o0o_____



BÀI TẬP LỚN

**Đề tài: Hệ thống đo nhiệt độ, độ ẩm, ánh sáng & điều
khiển thiết bị**

Môn học: IoT và ứng dụng
Nhóm lớp: 05
Họ tên SV: Đoàn Thị Trà My
Mã SV: B21DCAT134

Giảng viên hướng dẫn: Nguyễn Quốc Uy

HÀ NỘI, 10/2024

TÓM TẮT NỘI DUNG

Bài tập lớn này trình bày một hệ thống IoT giúp người dùng theo dõi và điều khiển các thông số môi trường trong thời gian thực. Với nền tảng là ESP32/ESP8266, hệ thống có thể thu thập dữ liệu từ các cảm biến như nhiệt độ, độ ẩm, ánh sáng và điều khiển các thiết bị như đèn, quạt một cách linh hoạt. Điểm nổi bật của dự án là giao diện người dùng trực quan, dễ sử dụng, giúp người dùng không chuyên cũng có thể dễ dàng lắp đặt và vận hành hệ thống.

MỤC LỤC

| | |
|--|-----------|
| CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI..... | 1 |
| 1.1 Đặt vấn đề..... | 1 |
| 1.2 Mục tiêu và định hướng giải pháp | 1 |
| 1.3 Đóng góp của bài tập lớn..... | 1 |
| 1.4 Bố cục bài tập lớn..... | 1 |
| CHƯƠNG 2. THIẾT KẾ HỆ THỐNG..... | 3 |
| 2.1 Tính năng và giao diện | 3 |
| 2.1.1 Tính năng..... | 3 |
| 2.1.2 Giao diện | 3 |
| 2.2 Thiết kế tổng thể..... | 4 |
| 2.3 Thiết kế chi tiết..... | 5 |
| 2.3.1 Thiết bị | 5 |
| 2.3.2 Sơ đồ tuần tự: Luồng dữ liệu | 5 |
| 2.3.3 Cơ sở dữ liệu | 6 |
| 2.3.4 API doc..... | 9 |
| CHƯƠNG 3. CẤU HÌNH..... | 14 |
| 3.1 Giao thức MQTT..... | 14 |
| 3.1.1 Cách cài đặt..... | 14 |
| 3.1.2 Kết quả chạy pub sub..... | 14 |
| 3.2 Kết nối Hardware với MQTT | 14 |
| 3.3 Code Backend | 14 |
| CHƯƠNG 4. KẾT QUẢ..... | 15 |

DANH MỤC HÌNH VẼ

| | | |
|-----------|--|----|
| Hình 2.1 | Giao diện Dashboard | 3 |
| Hình 2.2 | Giao diện Data Sensor | 3 |
| Hình 2.3 | Giao diện Acstion History | 4 |
| Hình 2.4 | Giao diện Profile | 4 |
| Hình 2.5 | Thiết kế tổng thể của hệ thống | 4 |
| Hình 2.6 | Luồng đọc dữ liệu từ cảm biến | 5 |
| Hình 2.7 | Luồng điều khiển thiết bị | 6 |
| Hình 2.8 | Cấu trúc bảng cơ sở dữ liệu tbl_data_sensor | 6 |
| Hình 2.9 | Dữ liệu trong bảng tbl_data_sensor | 7 |
| Hình 2.10 | Cấu trúc bảng cơ sở dữ liệu tbl_action_history | 7 |
| Hình 2.11 | Dữ liệu trong bảng tbl_action_history | 8 |
| Hình 2.12 | Các API | 9 |
| Hình 2.13 | API Lấy 10 dữ liệu cảm biến cuối để vẽ biểu đồ | 9 |
| Hình 2.14 | API Lấy toàn bộ dữ liệu cảm biến trong CSDL - input | 10 |
| Hình 2.15 | API Lấy toàn bộ dữ liệu cảm biến trong CSDL - output | 10 |
| Hình 2.16 | API Lấy dữ liệu lịch sử hành động bật/tắt thiết bị - input | 11 |
| Hình 2.17 | API Lấy dữ liệu lịch sử hành động bật/tắt thiết bị - input | 11 |
| Hình 2.18 | API Lấy trạng thái bật/tắt cuối cùng của thiết bị trong CSDL | 12 |
| Hình 2.19 | API Gửi tín hiệu bật tắt đèn | 13 |
| Hình 4.1 | Github code | 15 |

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong thời đại công nghệ 4.0, việc tự động hóa các thiết bị trong nhà đang trở nên ngày càng phổ biến. Với sự phát triển của công nghệ IoT, chúng ta có thể điều khiển các thiết bị điện từ xa thông qua các ứng dụng trên điện thoại di động. Bài tập lớn này nhằm mục tiêu xây dựng một hệ thống IoT kết hợp phần cứng và phần mềm, để giám sát các thông số môi trường như nhiệt độ, độ ẩm, ánh sáng,... và điều khiển các thiết bị điện như đèn, điều hòa, quạt,... thông qua một ứng dụng web thân thiện với người dùng.

1.2 Mục tiêu và định hướng giải pháp

Bài tập lớn này hướng tới mục tiêu tạo ra một sản phẩm cuối cùng có thể ứng dụng thực tế trong cuộc sống hàng ngày, như tại nhà ở, văn phòng, nhà xưởng. Người dùng có thể tùy chỉnh hệ thống để phù hợp với nhu cầu riêng của mình, từ việc giám sát nhiệt độ, độ ẩm trong phòng ngủ của trẻ sơ sinh, đến việc điều khiển hệ thống tưới tiêu tự động trong vườn.

Bên cạnh đó, bài tập lớn này còn xây dựng một hướng dẫn chi tiết, từng bước, giúp cho những người mới bắt đầu, kể cả những người không có nhiều kiến thức về lập trình hoặc điện tử, cũng có thể tự tay lắp ráp và lập trình một hệ thống giám sát và điều khiển môi trường tương tự. Bằng cách này, bài tập không chỉ mang tính học thuật mà còn có giá trị ứng dụng thực tế cao, góp phần phổ biến công nghệ IoT và khuyến khích mọi người tham gia vào việc xây dựng một cuộc sống thông minh hơn.

Để xây dựng một hệ thống IoT hiệu quả và chi phí thấp, bài tập này sử dụng ESP32/ESP8266 làm nền tảng chính. Các module này sở hữu nhiều ưu điểm như kích thước nhỏ gọn, tích hợp Wi-Fi, nhiều chân GPIO, dễ lập trình, và giá thành phải chăng.

1.3 Đóng góp của bài tập lớn

Bài tập lớn này có 2 đóng góp chính như sau:

1. Tạo ra một sản phẩm có thể ứng dụng thực tiễn.
2. Tạo ra tài liệu đầy đủ và dễ tiếp cận cho người mới thực hiện.

1.4 Bố cục bài tập lớn

Phần còn lại của báo cáo bài tập lớn này được tổ chức như sau.

Chương 2 sẽ trình bày chi tiết về thiết kế hệ thống, bao gồm: Giao diện, thiết kế tổng thể và thiết kế chi tiết.

Chương 3 sẽ trình bày một số cấu hình code chính của sản phẩm.

Chương 4 sẽ đưa ra kết quả sản phẩm cuối cùng.

Đi kèm với tài liệu này là mã nguồn với link github.

CHƯƠNG 2. THIẾT KẾ HỆ THỐNG

2.1 Tính năng và giao diện

2.1.1 Tính năng

- Ứng dụng theo dõi thông số trong phòng cho biết: nhiệt độ, độ ẩm, ánh sáng, độ bụi, mưa, khí gas,... thời gian thực đang như thế nào.
- Bật tắt đèn, điều hòa, quạt,... thông qua ứng dụng web hoặc app.

2.1.2 Giao diện

Bao gồm 4 giao diện chính:

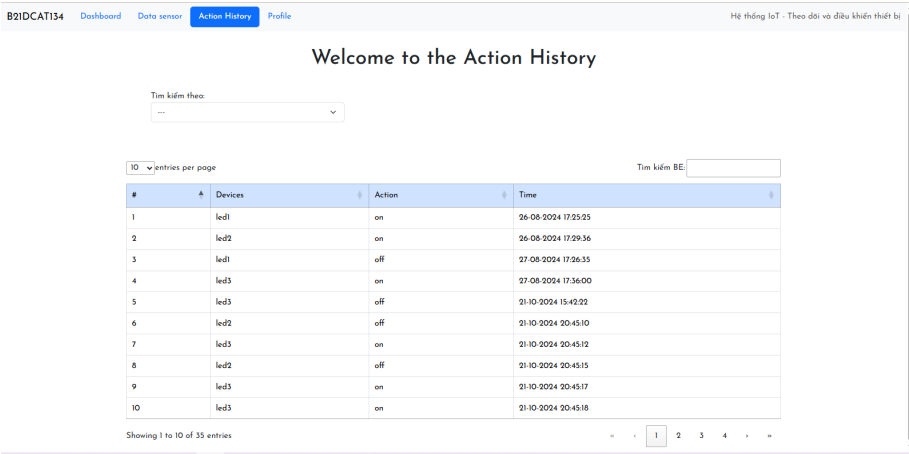


Hình 2.1: Giao diện Dashboard

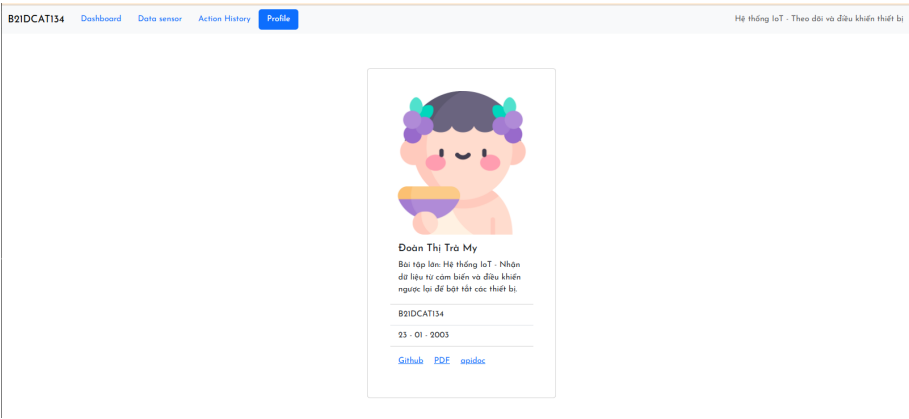
The Data sensor interface for B2IDCAT134 displays a table of sensor data entries. The table has columns for #, Humidity, Brightness, Temperature, and Time. The data shows a sequence of readings over time, with values for Humidity, Brightness, and Temperature fluctuating around 30-40, 400-600, and 30-40 respectively.

| # | Humidity | Brightness | Temperature | Time |
|----|----------|------------|-------------|---------------------|
| 1 | 30 | 1000 | 30 | 02-08-2024 17:30:44 |
| 2 | 40 | 400 | 32 | 03-08-2024 17:31:18 |
| 3 | 23 | 430 | 34 | 04-08-2024 17:31:18 |
| 5 | 34 | 200 | 36 | 26-08-2024 21:00:35 |
| 6 | 23 | 450 | 35 | 27-08-2024 21:00:35 |
| 7 | 45 | 659 | 43 | 26-08-2024 16:00:25 |
| 8 | 67 | 890 | 27 | 27-08-2024 21:45:35 |
| 9 | 90 | 600 | 23 | 27-08-2024 21:56:35 |
| 26 | 15 | 303 | 19 | 21-10-2024 00:13:07 |
| 27 | 12 | 304 | 49 | 21-10-2024 09:14:55 |

Hình 2.2: Giao diện Data Sensor

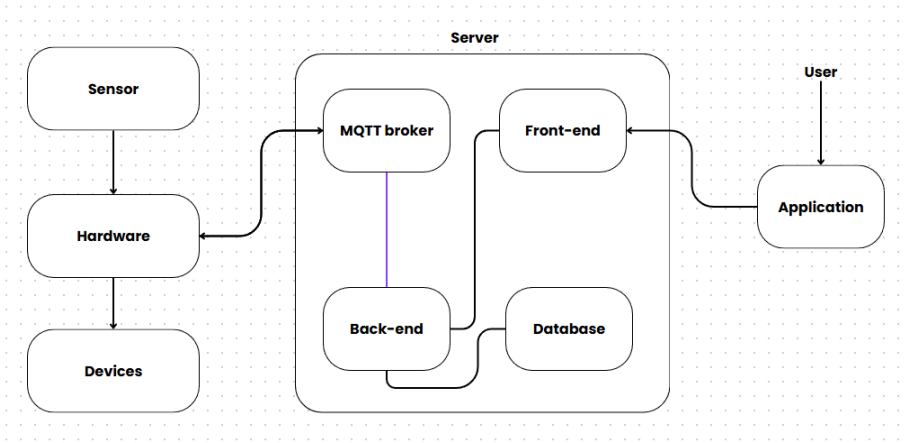


Hình 2.3: Giao diện Acstion History



Hình 2.4: Giao diện Profile

2.2 Thiết kế tổng thể



Hình 2.5: Thiết kế tổng thể của hệ thống

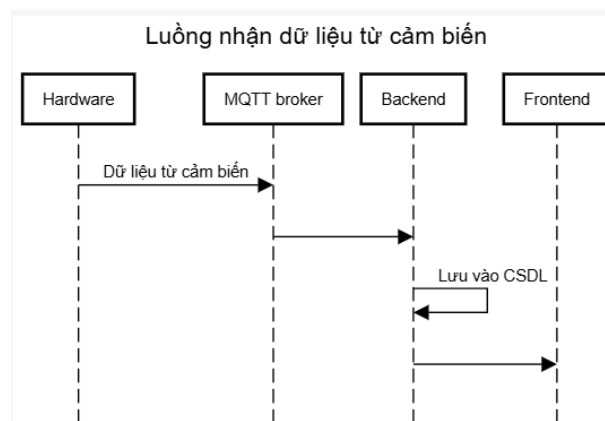
2.3 Thiết kế chi tiết

2.3.1 Thiết bị

- ESP32 hoặc ESP8266
- Cảm biến:
 - DHT11 hoặc DHT22
 - Cảm biến quang trở
 - Cảm biến nhiệt độ LM35
- breadboard: ổ cắm
- Dây nối
- đèn led

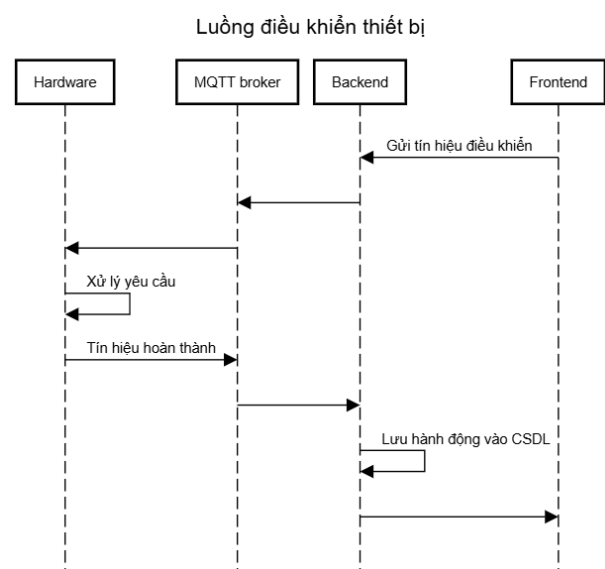
2.3.2 Sơ đồ tuần tự: Luồng dữ liệu

a, Luồng đọc dữ liệu



Hình 2.6: Luồng đọc dữ liệu từ cảm biến

b, Luồng điều khiển thiết bị



Hình 2.7: Luồng điều khiển thiết bị

2.3.3 Cơ sở dữ liệu

Máy chủ: 127.0.0.1 > Cơ sở dữ liệu: iot_database > Bảng: tbl_data_sensor

Duyệt | Cấu trúc | SQL | Tìm kiếm | Chèn | Xuất | Nhập | Đặc quyền | Thao tác | Theo

Cấu trúc bảng | Hiện thị quan hệ

| # | Tên | Kiểu | Bảng mã đối chiếu | Thuộc tính | Null | Mặc định | Ghi chú | Thêm | Hành động |
|----------------------------|-------------|----------|-------------------|------------|-------|----------|----------------|------|-------------------|
| <input type="checkbox"/> 1 | id | int(11) | | Không | Không | | AUTO_INCREMENT | | Thay đổi Xóa Thêm |
| <input type="checkbox"/> 2 | humid | int(11) | | Không | Không | | | | Thay đổi Xóa Thêm |
| <input type="checkbox"/> 3 | bright | int(11) | | Không | Không | | | | Thay đổi Xóa Thêm |
| <input type="checkbox"/> 4 | temperature | int(11) | | Không | Không | | | | Thay đổi Xóa Thêm |
| <input type="checkbox"/> 5 | time | datetime | | Không | Không | | | | Thay đổi Xóa Thêm |

Hình 2.8: Cấu trúc bảng cơ sở dữ liệu tbl_data_sensor

Cơ sở dữ liệu: iot_database » Bảng: tbl_data_sensor

SQL Tìm kiếm Chèn Xuất Nhập

| | id | humid | bright | temperature | time |
|--------|----|-------|--------|-------------|---------------------|
| Xóa bỏ | 1 | 30 | 1000 | 30 | 2024-08-02 17:30:44 |
| Xóa bỏ | 2 | 40 | 400 | 32 | 2024-08-03 17:31:18 |
| Xóa bỏ | 3 | 23 | 430 | 34 | 2024-08-04 17:31:18 |
| Xóa bỏ | 5 | 34 | 200 | 36 | 2024-08-26 21:00:35 |
| Xóa bỏ | 6 | 23 | 450 | 35 | 2024-08-27 21:00:35 |
| Xóa bỏ | 7 | 45 | 659 | 43 | 2024-08-26 16:00:25 |
| Xóa bỏ | 8 | 67 | 890 | 27 | 2024-08-27 21:45:35 |
| Xóa bỏ | 9 | 90 | 600 | 23 | 2024-08-27 21:56:35 |
| Xóa bỏ | 26 | 15 | 303 | 19 | 2024-10-21 00:13:07 |
| Xóa bỏ | 27 | 12 | 304 | 49 | 2024-10-21 09:14:55 |
| Xóa bỏ | 28 | 12 | 305 | 49 | 2024-10-21 09:28:17 |
| Xóa bỏ | 29 | 12 | 300 | 49 | 2024-10-21 09:30:20 |
| Xóa bỏ | 30 | 12 | 300 | 49 | 2024-10-21 09:38:11 |
| Xóa bỏ | 31 | 12 | 320 | 49 | 2024-10-21 09:40:36 |
| Xóa bỏ | 32 | 12 | 323 | 49 | 2024-10-21 10:02:32 |
| Xóa bỏ | 33 | 12 | 301 | 49 | 2024-10-21 10:03:10 |
| Xóa bỏ | 34 | 12 | 391 | 49 | 2024-10-21 10:05:52 |
| Xóa bỏ | 35 | 12 | 392 | 49 | 2024-10-21 10:08:47 |
| Xóa bỏ | 36 | 12 | 393 | 49 | 2024-10-21 10:10:23 |

Hình 2.9: Dữ liệu trong bảng tbl_data_sensor

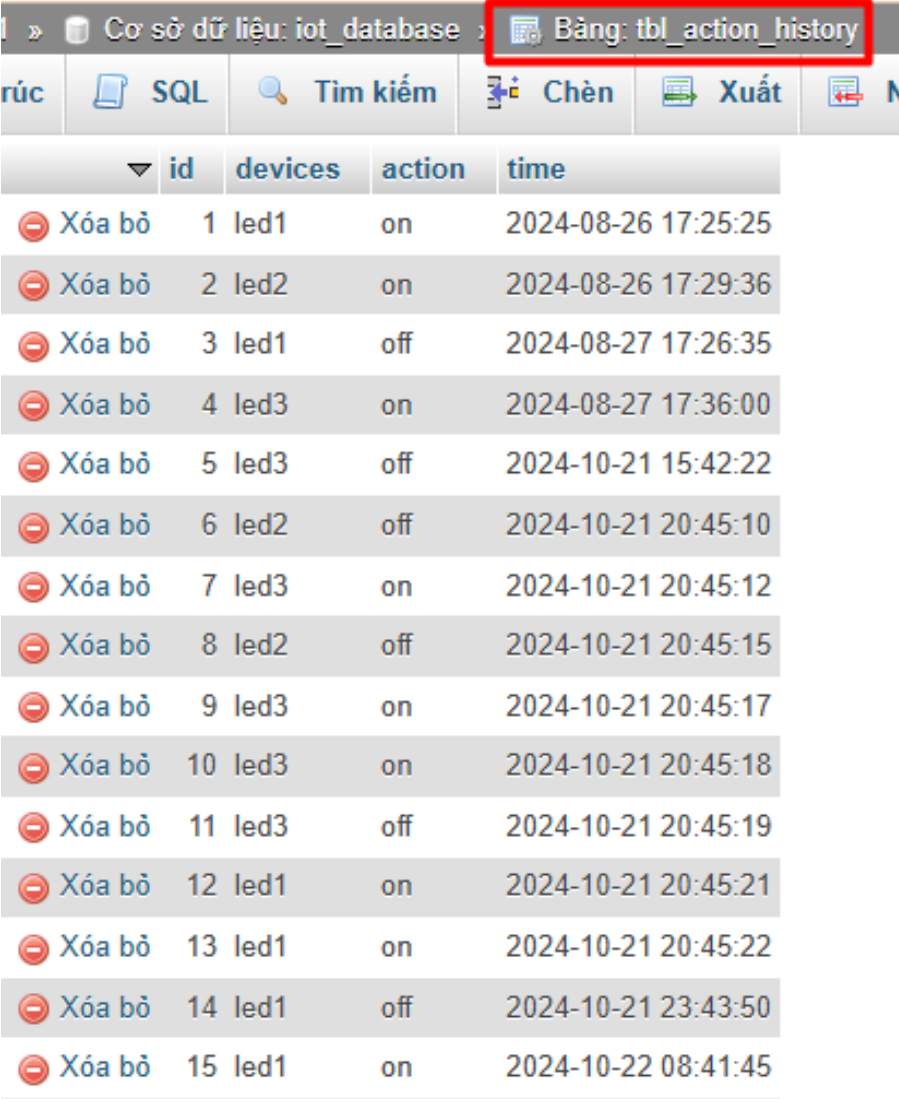
Máy chủ: 127.0.0.1 » Cơ sở dữ liệu: iot_database » Bảng: tbl_action_history

Duyệt Cấu trúc SQL Tìm kiếm Chèn Xuất Nhập Đặc quyền Thao tác Theo dõi 26 B

Cấu trúc bảng Hiển thị quan hệ

| # | Tên | Kiểu | Bảng mã đối chiếu | Thuộc tính | Null | Mặc định | Ghi chú | Thêm | Hành động |
|---|---------|------------------------------|--------------------|------------|-------|----------|----------------|------|-------------------|
| 1 | id | int(11) | | Không | Không | | AUTO_INCREMENT | | Thay đổi Xóa Thêm |
| 2 | devices | enum('led1', 'led2', 'led3') | utf8mb4_general_ci | Không | Không | | | | Thay đổi Xóa Thêm |
| 3 | action | enum('on', 'off') | utf8mb4_general_ci | Không | Không | | | | Thay đổi Xóa Thêm |
| 4 | time | datetime | | Không | Không | | | | Thay đổi Xóa Thêm |

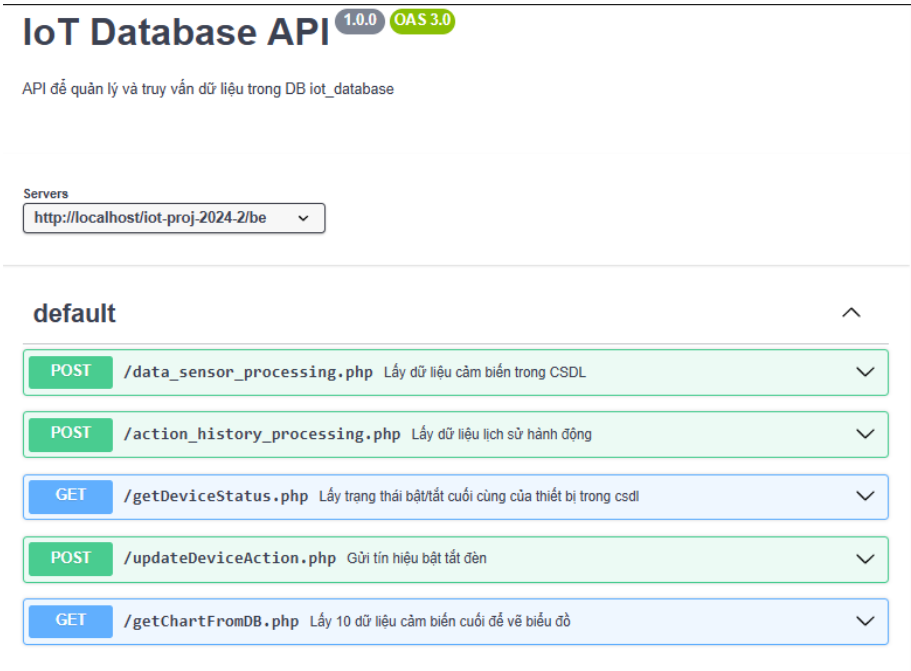
Hình 2.10: Cấu trúc bảng cơ sở dữ liệu tbl_action_history



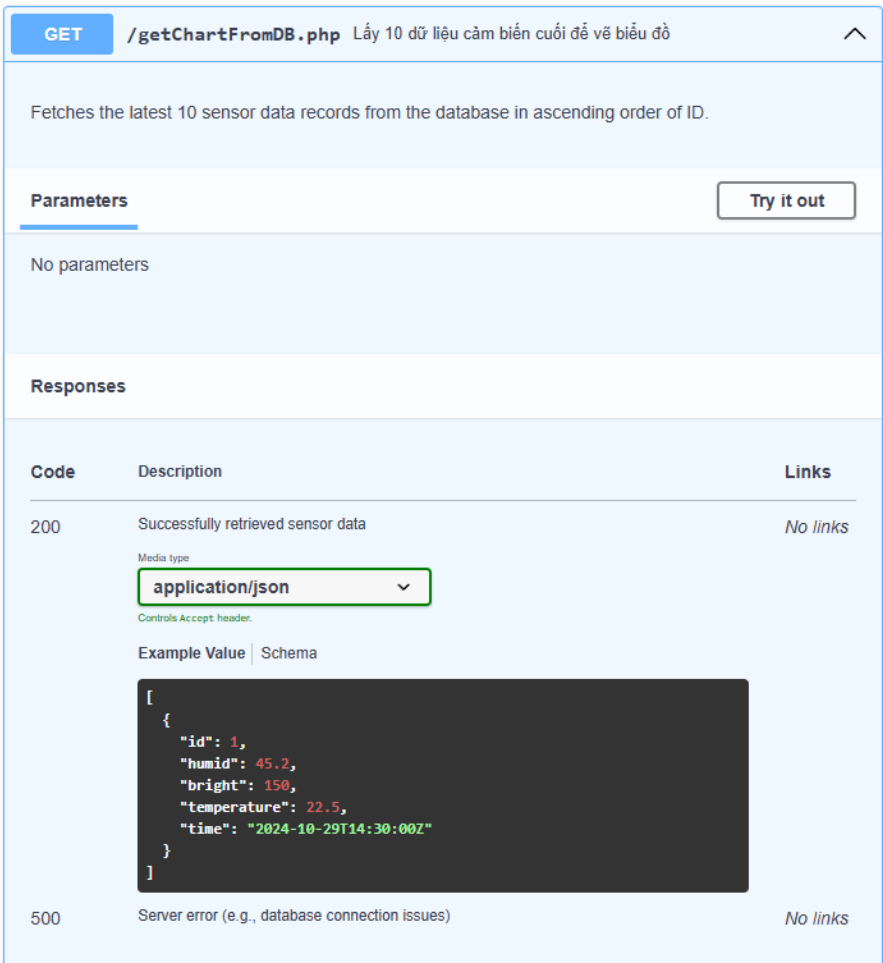
| | id | devices | action | time |
|--------|----|---------|--------|---------------------|
| Xóa bỏ | 1 | led1 | on | 2024-08-26 17:25:25 |
| Xóa bỏ | 2 | led2 | on | 2024-08-26 17:29:36 |
| Xóa bỏ | 3 | led1 | off | 2024-08-27 17:26:35 |
| Xóa bỏ | 4 | led3 | on | 2024-08-27 17:36:00 |
| Xóa bỏ | 5 | led3 | off | 2024-10-21 15:42:22 |
| Xóa bỏ | 6 | led2 | off | 2024-10-21 20:45:10 |
| Xóa bỏ | 7 | led3 | on | 2024-10-21 20:45:12 |
| Xóa bỏ | 8 | led2 | off | 2024-10-21 20:45:15 |
| Xóa bỏ | 9 | led3 | on | 2024-10-21 20:45:17 |
| Xóa bỏ | 10 | led3 | on | 2024-10-21 20:45:18 |
| Xóa bỏ | 11 | led3 | off | 2024-10-21 20:45:19 |
| Xóa bỏ | 12 | led1 | on | 2024-10-21 20:45:21 |
| Xóa bỏ | 13 | led1 | on | 2024-10-21 20:45:22 |
| Xóa bỏ | 14 | led1 | off | 2024-10-21 23:43:50 |
| Xóa bỏ | 15 | led1 | on | 2024-10-22 08:41:45 |

Hình 2.11: Dữ liệu trong bảng tbl_action_history

2.3.4 API doc



Hình 2.12: Các API



Hình 2.13: API Lấy 10 dữ liệu cảm biến cuối để vẽ biểu đồ

POST

/data_sensor_processing.php

Lấy dữ liệu cảm biến trong CSDL

^

Parameters

Try it out

No parameters

Request body

required

application/x-www-form-urlencoded

▼

draw

integer

Số lần yêu cầu (tăng dần theo mỗi lần yêu cầu)

start

integer

Chỉ số bắt đầu của dữ liệu

length

integer

Số lượng bản ghi cần lấy

search

object

Giá trị tìm kiếm

order

array

Chỉ số cột để sắp xếp và hướng sắp xếp (asc hoặc desc)

searchColumn

string

Cột cần tìm kiếm

Hình 2.14: API Lấy toàn bộ dữ liệu cảm biến trong CSDL - input

Responses

Code

Description

Links

200

A list of action history records

No links

Media type

application/json

▼

Controls

Accept header

Example Value

Schema

```
{
  "draw": 0,
  "recordsTotal": 0,
  "recordsFiltered": 0,
  "data": [
    {
      "id": 0,
      "device": "string",
      "action": "string",
      "time": "2024-10-29T14:35:21.474Z"
    }
  ],
  "extraInfo": "string"
}
```

Hình 2.15: API Lấy toàn bộ dữ liệu cảm biến trong CSDL - output

POST

/action_history_processing.php

Lấy dữ liệu lịch sử hành động

⌵

Parameters

Try it out

No parameters

Request body

required

application/x-www-form-urlencoded

⌵

draw

integer

Số lần yêu cầu (tăng dần theo mỗi lần yêu cầu)

start

integer

Chỉ số bắt đầu của dữ liệu

length

integer

Số lượng bản ghi cần lấy

search

object

order

array

searchColumn

string

Cột cần tìm kiếm

Hình 2.16: API Lấy dữ liệu lịch sử hành động bật/tắt thiết bị - input

Responses

Code

Description

Links

200

Danh sách lịch sử hành động

No links

Media type

application/json

⌵

Controls

Accept header

Example Value

Schema

```
{
  "draw": 0,
  "recordsTotal": 0,
  "recordsFiltered": 0,
  "data": [
    {
      "id": 0,
      "devices": "string",
      "action": "string",
      "time": "2024-10-29T14:38:10.446Z"
    }
  ],
  "extraInfo": "string"
}
```

Hình 2.17: API Lấy dữ liệu lịch sử hành động bật/tắt thiết bị - input

GET

/getDeviceStatus.php

Lấy trạng thái bật/tắt cuối cùng của thiết bị trong csdl

^

This endpoint retrieves the latest status of each device (led1, led2, led3) as recorded in the database.

Parameters

Try it out

No parameters

Responses

| Code | Description | Links |
|------|--|----------|
| 200 | Lấy thành công trạng thái bật tắt của thiết bị | No links |
| | <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <pre>{ "led1": "on", "led2": "on", "led3": "on" }</pre> | |
| 500 | Lỗi server, kết nối với csdl không thành công. | No links |

Hình 2.18: API Lấy trạng thái bật/tắt cuối cùng của thiết bị trong CSDL

POST

/updateDeviceAction.php

Gửi tín hiệu bật tắt đèn

This endpoint sends a request to turn a device on or off by publishing a message to an MQTT topic. The action is only saved to the database upon receiving a confirmation message from the device.

Parameters

Try it out

No parameters

Request body required

application/json

Example Value

Schema

```
{
  "device": "led1",
  "action": "on"
}
```

Responses

| Code | Description | Links |
|------|--|----------|
| 200 | Successfully sent action to the device | No links |
| | <div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div><div>Example Value</div><div>Schema</div><div><pre>{ "success": true }</pre></div></div></div> | |
| 400 | Bad request (e.g., missing or invalid parameters) | No links |
| 500 | Server error (e.g., MQTT connection issues) | No links |

Hình 2.19: API Gửi tín hiệu bật tắt đèn

CHƯƠNG 3. CẤU HÌNH

3.1 Giao thức MQTT

Dùng giao thức MQTT để nhận và gửi dữ liệu lên topic.

3.1.1 Cách cài đặt

a, Đổi cổng kết nối

Mở tệp cấu hình Mosquitto: Tìm tệp cấu hình `mosquitto.conf`. Nếu đã cài đặt Mosquitto theo mặc định, tệp này thường nằm trong thư mục cài đặt của Mosquitto, ví dụ: `C:\Program Files\mosquitto`.

Chỉnh sửa tệp cấu hình: Mở tệp `mosquitto.conf` bằng trình soạn thảo văn bản như Notepad. Tìm dòng chứa `listener` và thay đổi cổng theo ý muốn. Ví dụ, để sử dụng cổng 2003 thay vì 1883:

```
1 listener 1884
```

Lưu và khởi động lại Mosquitto: Lưu tệp `mosquitto.conf` sau khi chỉnh sửa. Khởi động lại dịch vụ Mosquitto bằng cách mở Command Prompt với quyền quản trị và chạy lệnh:

```
1 net stop mosquitto
2 net start mosquitto
```

3.1.2 Kết quả chạy pub sub

3.2 Kết nối Hardware với MQTT

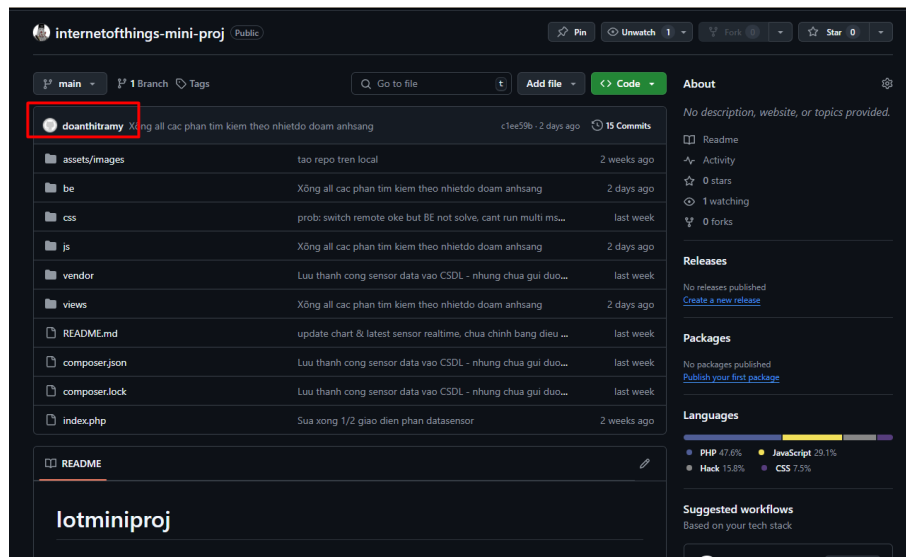
Mở Andruino, để nạp code cho ESP32.

3.3 Code Backend

Cấu hình BE được thiết lập trong folder `\be` tại link [Github](#)

CHƯƠNG 4. KẾT QUẢ

Phần code tại: Github - người thực hiện: doanthitramy



Hình 4.1: Github code