

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN ĐHQG – HCM

KHOA CÔNG NGHỆ THÔNG TIN

-----∞O∞-----



**BÁO CÁO CÁ NHÂN**  
**DỰ ÁN KHÓA CỬA THÔNG MINH**

**Học phần:** Vật lý cho Công nghệ Thông tin

**Giảng viên hướng dẫn:** ThS Cao Xuân Nam

ThS Đặng Hoài Thương

**Sinh viên:** 22127344 – Lý Anh Quân

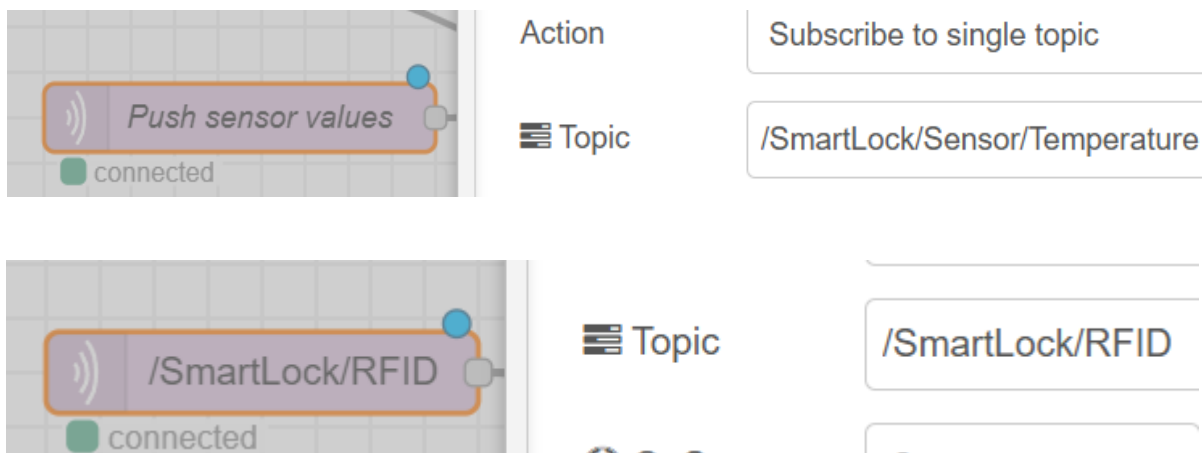
TP. Hồ Chí Minh, ngày 17 tháng 08 năm 2024

## I. Phần yêu cầu căn bản:

### Phần Website:

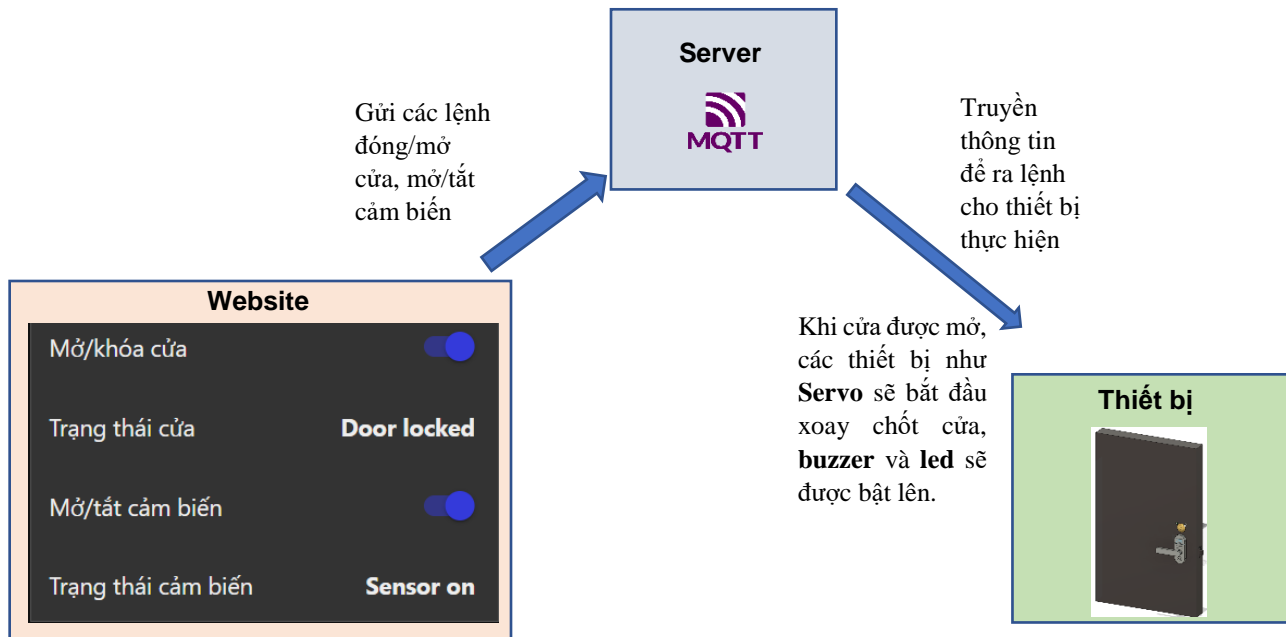
#### 1. Có thể hiển thị được thông tin cơ bản đọc được từ thiết bị INPUT thông qua internet

- Ở phần Website, chúng em đã thực hiện việc gửi từ thiết bị lên website các dữ liệu về **cảm biến nhiệt độ, độ ẩm, mã thẻ từ** được quét. Với các dữ liệu được gửi lên thì em sẽ thực hiện việc lấy dữ liệu thông qua việc **subcibe topic chung** giữa thiết bị và website. Thông qua **server mqtt** và **node mqtt in** thì ta sẽ có được dữ liệu từ thiết bị gửi lên.

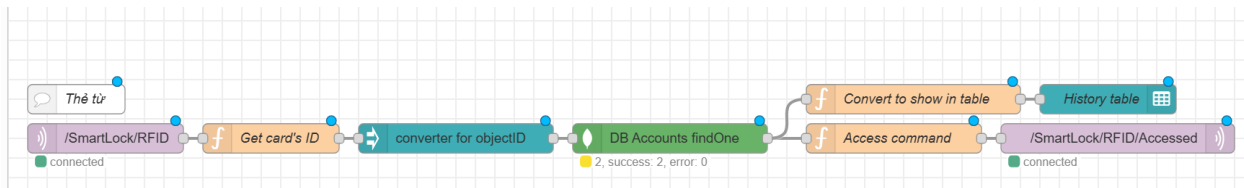


- Với mỗi dữ liệu cảm biến và độ ẩm được gửi lên, chúng sẽ được hiển thị theo thời gian thực thông qua **Biểu đồ Gauge** và sẽ được ghi lại vào trong **cloud ThingSpeak**. Từ đó ta sẽ có qua hai biểu đồ đường biểu diễn các giá trị dữ liệu gần nhất.
  - Với mỗi thẻ từ được quét vào, **giá trị thẻ từ** (ở đây chúng em xét dựa vào **id riêng biệt và duy nhất** của mỗi accounts nên sẽ không trùng nhau) sẽ được kiểm tra trên cơ sở dữ liệu MongoDB để xem thẻ từ này đã có trên cơ sở dữ liệu hay chưa và cấp phép cho người dùng dựa vào đó.
- #### 2. Có ít nhất một chức năng trên web có thể gửi dữ liệu xuống thiết bị thông qua internet
- Chức năng mở/khóa cửa dựa vào công tắc bật/tắt switch sẽ truyền dữ liệu xuống thiết bị và yêu cầu các thiết bị output thực hiện. Dữ liệu về những lần đóng/mở cửa này cũng sẽ được lưu lại vào cloud ThingSpeak.
  - Chức năng mở/tắt cảm biến sẽ thực hiện việc gửi lệnh điều khiển tắt/mở cảm biến nhiệt độ, độ ẩm.

## Sơ đồ luồng xử lý từ Website → Thiết bị



- Chức năng kiểm tra thẻ khóa từ có hợp lệ hay không qua **\_id** của từng tài khoản đã đăng ký: Khi người dùng sử dụng sản phẩm, mọi account được tạo sẽ có một mã **định danh \_id duy nhất** và khớp với dữ liệu của thẻ từ, khi đó mỗi khi người dùng quét thẻ từ thì chúng ta sẽ nhận **dữ liệu thẻ từ** từ thiết bị và đem chúng so sánh với **\_id** đã có trong MongoDB. Nếu khớp, ta sẽ gửi lệnh xác nhận về cho thiết bị (Access allowed) còn nếu không thì ta sẽ gửi về lệnh không chấp nhận thẻ từ này (Access denied).



## II. Phân yêu cầu nâng cao:

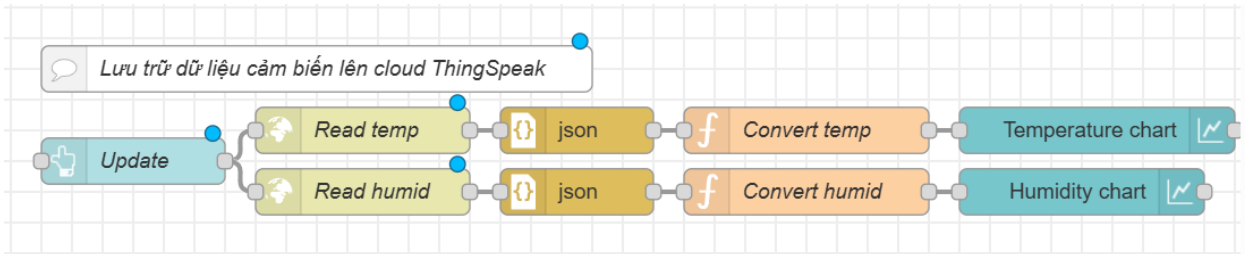
### 1. Có sử dụng thêm thiết bị OUTPUT trong nội dung giảng dạy

- Các thiết bị output đã sử dụng cho thiết bị khóa cửa thông minh của nhóm là:
- **LCD 20x4 (kèm mạch I2C):** Dùng để hiển thị các thông báo cho người dùng như trạng thái ánh sáng, nhiệt độ và độ ẩm.
- **Buzzer 5VDC:** Được dùng để phát ra âm thanh nhỏ khi cửa vừa được mở và khi có cảnh báo về mail, điện thoại người dùng.
- **Servo:** Thu đẩy chốt cửa mỗi lúc cửa đóng hoặc mở, đóng vai trò như hỗ trợ một chốt khóa
- **Led:** Đèn để thông báo khi cửa đã mở

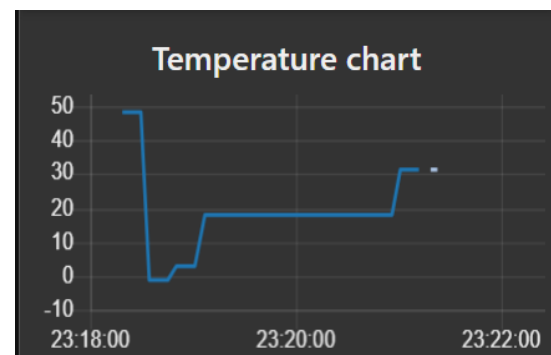
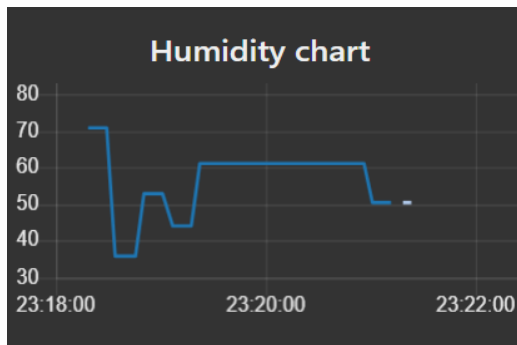
## 2. Có sử dụng thêm thiết bị OUTPUT ngoài nội dung giảng dạy

- **Đèn rgb:** Khi trời sáng, hoặc có người chuyển động ở gần khu vực cửa thì đèn RGB sẽ bật lên nhằm soi sáng cho người dùng

## 3. Có thể xem lịch sử dữ liệu đã lưu trên cloud trên giao diện web



- **Cloud ThingSpeak** là nơi mà chúng em lưu trữ các nhiệt độ cảm biến đo được nên vì thế để đọc được thì ta sẽ sử dụng **api** đã có khi tạo tài khoản và thực hiện đọc các điểm dữ liệu gần nhất.
- Sau khi đọc thành công, để có thể lấy được dữ liệu thì ta phải đổi sang dạng **JSON object** và sau đó convert sang các số nguyên để thêm vào biểu đồ
- Đây là giao diện trên Website



## 4. Có giao diện web cho phép đăng ký tài khoản (ghi thành công thông tin tài khoản vào database/cloud/file)

- Giao diện chính của cửa sổ đăng ký

Sign up

Email \*

Username \*

Password \*

SUBMIT

TAP TO LOGIN

- Khi người dùng điền xong thông tin đăng ký, thông tin đăng ký sẽ được gửi đi và đối chiếu với những bản ghi dữ liệu đã có trong cơ sở dữ liệu **MongoDB** qua **phương thức findOne**. Nếu như tìm được dữ liệu trùng với dữ liệu mới đăng ký – tức là đã có người dùng này rồi thì sẽ gửi về đăng ký thất bại. Còn nếu chưa có dữ liệu thì sẽ dùng **phương thức Insert** của MongoDB để thêm tài khoản mới vào cơ sở dữ liệu.

```
{  
  "_id": ObjectId('66c8b74329307d7c32f8f125'),  
  "email": "anhquan@gmail.com",  
  "username": "anhquan",  
  "password": "****"  
}
```

```
{  
  "_id": ObjectId('66c8b79d29307d7c32f8f126'),  
  "email": "oung@gmail.com",  
  "username": "thanhtan",  
  "password": "*****"  
}
```

- **Khi thành công:** Một thông báo hiện ra và chuyển hướng sang trang đăng nhập
- **Khi thất bại:** Là khi người dùng này điền thông tin đã có trong cơ sở dữ liệu, chúng ta sẽ hiện thông báo cho người dùng.

Đăng ký thành công

Đăng ký thất bại

Đăng ký không thành công vì đã có tài khoản này. Hãy thử đăng nhập!

OK

## 5. Có giao diện web cho phép đăng nhập hệ thống:

- Giao diện chính của phần đăng nhập

Login

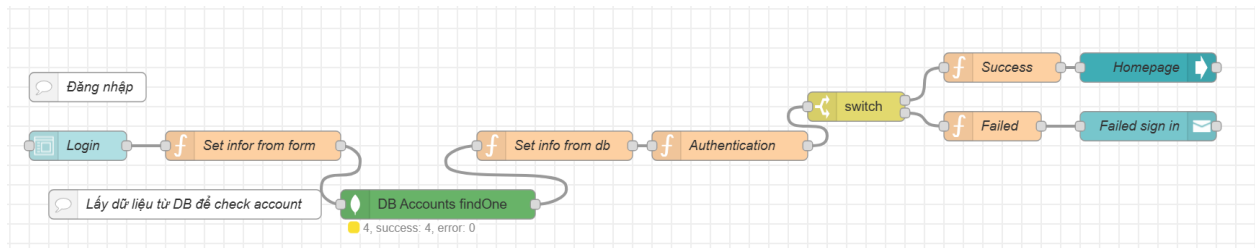
username \*

password \*

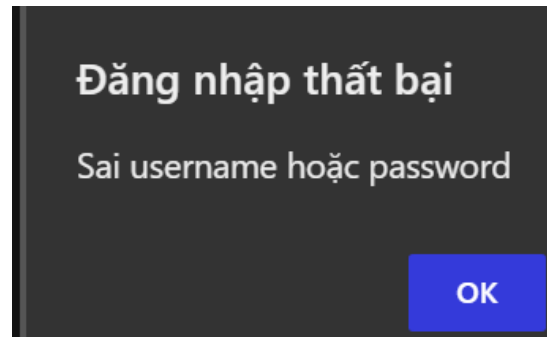
SUBMIT

TAP TO SIGN UP

- Với mỗi lần người dùng điền thông tin đăng nhập vào trong form đăng nhập thì ta sẽ lấy dữ liệu đăng nhập và dùng phương thức **findOne** của thư viện **MongoDB** trên node-red để tìm xem liệu tên đăng nhập và mật khẩu này có trùng khớp hay không.



- Dựa vào bước kiểm tra đó ta sẽ có 2 kết quả là:
- Nếu có, ta sẽ được chuyển đến màn hình chính. Nếu không thì sẽ có thông báo đăng nhập thất bại



- Giao diện web chính sau khi đăng nhập thành công

