

BỘ GIÁO DỤC & ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH



CƠ SỞ VÀ ỨNG DỤNG IOTS

BÀI TẬP CUỐI HỌC KỲ

**ĐỀ TÀI: Hệ thống thu thập dữ liệu thông qua LoRa
và tải dữ liệu lên NodeJS, Firebase, MQTT**

Giảng viên hướng dẫn: Trương Quang Phúc

Sinh viên thực hiện:

Nguyễn Tấn An	19151069
Bùi Lê Anh	19151099
Nguyễn Quốc Khánh	19151140

Tp. Hồ Chí Minh, tháng 11 năm 2022

Thành viên	Phân công nhiệm vụ	Email
Nguyễn Tấn An	Thiết lập phần cứng và giao tiếp LoRa với ESP Master	19151069@student.hcmute.edu.vn
Bùi Lê Anh	Giao tiếp ESP Master với web server thông qua MQTT	19151099@student.hcmute.edu.vn
Nguyễn Quốc Khánh	Xây dựng web server NodeJS và giao diện người dùng. Giao tiếp ESP Master với web server thông qua Firebase	19151140@student.hcmute.edu.vn

MỤC LỤC

1. Giới thiệu	1
2. Thiết kế	2
2.1. Sơ đồ khối.....	2
2.2. Phương pháp thực hiện.....	3
2.2.1. Web server NodeJS	3
2.2.2. Firebase Realtime Database	4
2.2.3. MQTT và Adfruit IO	5
2.2.4. LoRa	6
3. Kết quả.....	8
4. Kết luận và hướng phát triển	12
4.1. Kết luận	12
4.2. Hướng phát triển	12

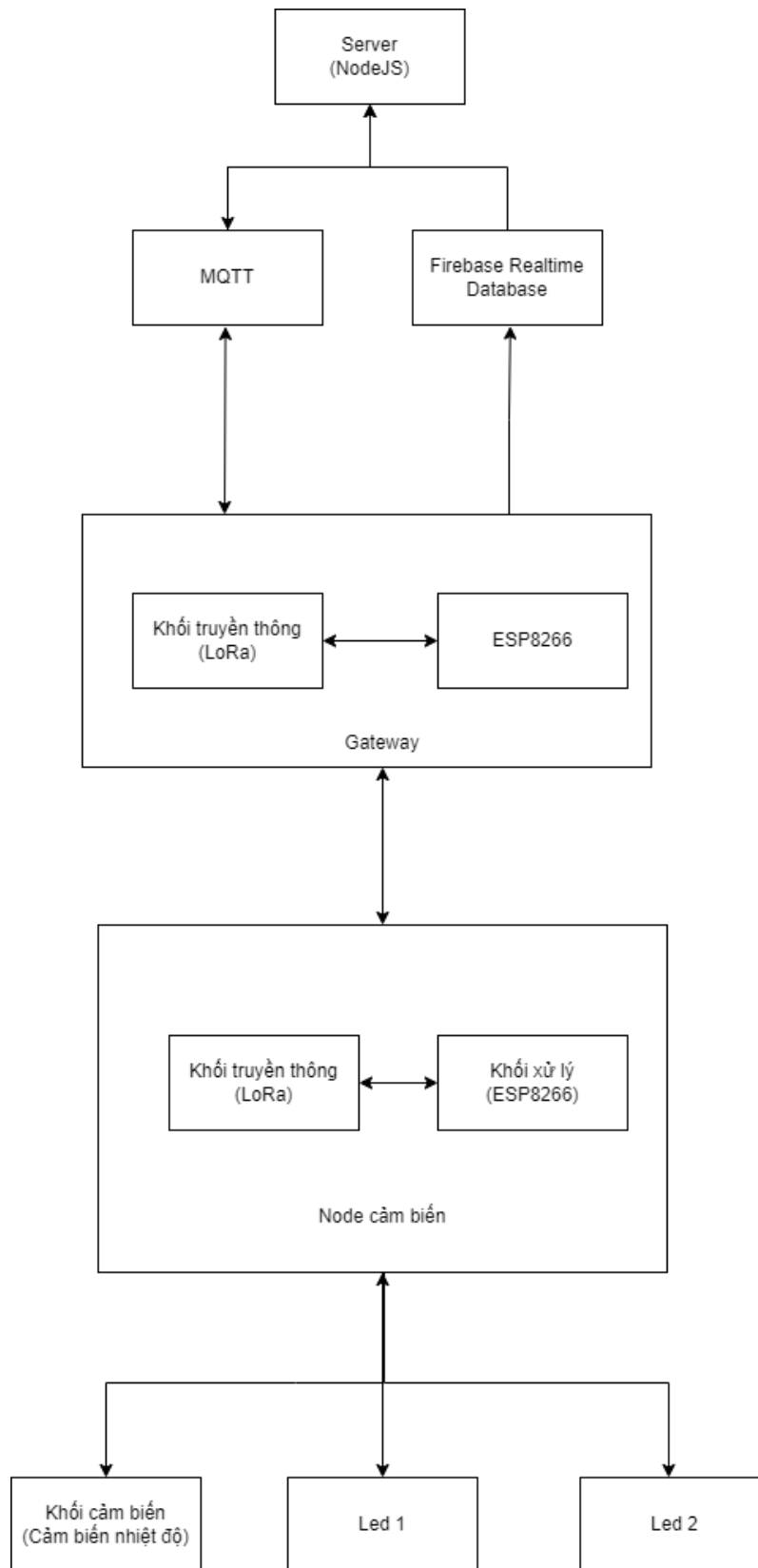
1. Giới thiệu

Nhờ vào Internet vạn vật (IoT - Internet of Things) mà chúng ta có thể tích hợp các hệ thống truyền thông khác nhau sử dụng để kết nối, thu thập dữ liệu và điều khiển nhiều thiết bị khác nhau chỉ với một chiếc điện thoại hoặc laptop được kết nối internet.

Dựa vào sự tiện lợi đó, hệ thống thu thập dữ liệu, điều khiển thông qua LoRa và tải dữ liệu lên NodeJS, Firebase Realtime Database, MQTT được thực hiện để thu thập dữ liệu, giám sát, và điều khiển hệ thống với các cảm biến, vì điều khiển ESP8266 được kết nối với nhau thông qua mạng truyền thông LoRa và hệ thống lưu trữ dữ liệu Firebase Realtime Database. Hệ thống có khả năng thu thập và điều khiển linh hoạt, nhanh chóng ở khoảng cách xa.

2. Thiết kế

2.1. Sơ đồ khối



Hình 1: Sơ đồ khối

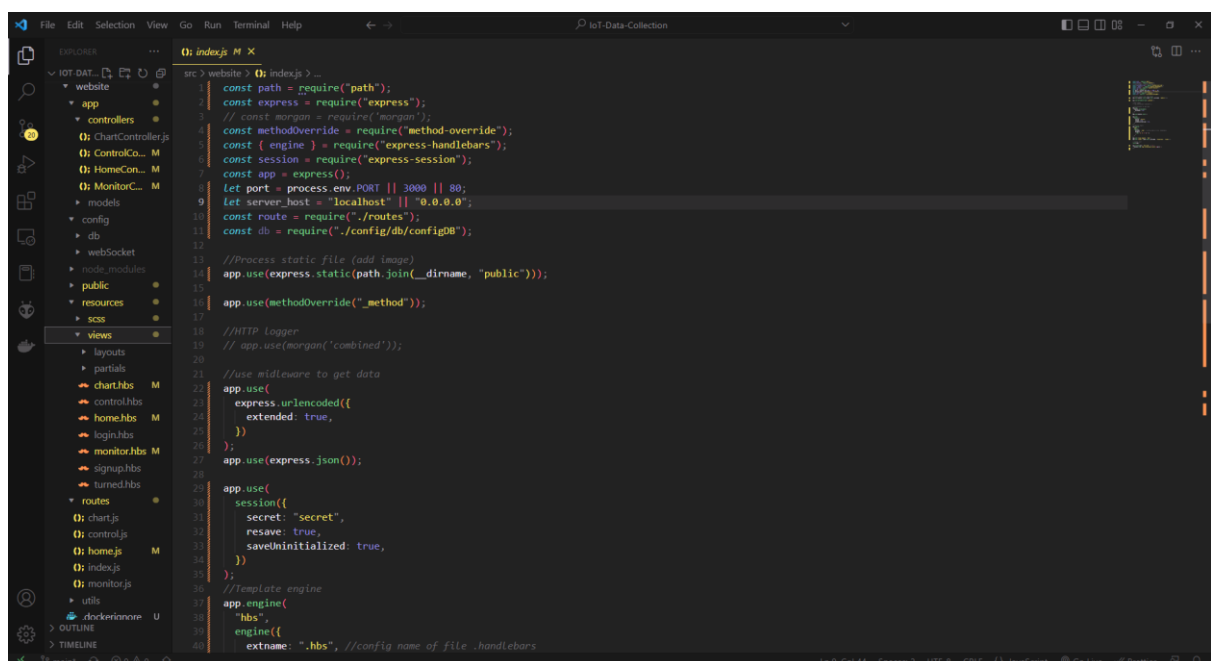
2.2. Phương pháp thực hiện

Hệ thống bao gồm 3 phần chính là Node cảm biến, Gateway và Server. Đối với luồng thu thập dữ liệu: các cảm biến (DHT-11) gửi thông tin nhiệt độ đến bộ xử lý của node cảm biến là ESP8266 để xử lý dữ liệu và gửi sang khối gateway thông qua hai module LoRa E32-TTL-100. Bộ xử lý master (ESP8266) của khối gateway sẽ gửi thông tin đến server NodeJS thông qua cơ sở dữ liệu Firebase Real Time Database.

Đối với luồng điều khiển: server NodeJS sẽ gửi tín hiệu điều khiển đến khối gateway thông qua server MQTT của adafruit.io. Khối gateway tiếp tục gửi tín hiệu điều khiển xuống node cảm biến thông qua hai module LoRa E32-TTL-100. Và bộ xử lý ESP8266 của node sẽ thực hiện bật/tắt đèn dựa vào tín hiệu nhận được.

2.2.1. Web server NodeJS

NodeJS là một nền tảng được xây dựng trên “V8 Javascript engine” được viết bằng c++ và Javascript. Nền tảng này được phát triển bởi Ryan Lienhart Dahl vào năm 2009. Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.



Hình 2: NodeJS Server

Web server sử dụng Expressjs - một framework được xây dựng trên nền tảng của Nodejs cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

2.2.2. Firebase Realtime Database

Realtime Database là một cơ sở dữ liệu NoSQL có chức năng lưu và đồng bộ dữ liệu trên mây dưới dạng JSON.

Kết nối Firebase với ESP:

- Bước 1: Khai báo biến thông tin cho Wifi và Firebase.
- Bước 2: Kết nối wifi cho ESP8266. (Hình 3)
- Bước 3: Khởi tạo thông tin xác thực của Firebase. (Hình 4)
- Bước 4: Gửi thông tin lên Firebase với hàm setFloat, setInt, setString,... và nhận thông tin từ firebase với hàm getFloat, getInt, getString,... (Hình 5)

```
WiFi.begin (WIFI_SSID, WIFI_PASSWORD);
Serial.print("Dang ket noi");
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
}

Serial.println("");
Serial.println ("Da ket noi WiFi!");
Serial.println(WiFi.localIP());
```

Hình 3: Kết nối wifi

```
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
delay(100);
```

Hình 4: Khởi tạo thông tin firebase

```
void FireBase()
{
  srand (time(NULL));
  char ref[100] = "Temperature/";
  Firebase.setFloat ( fbdo,"Led 4", StatusLed_4);
  Firebase.setFloat ( fbdo,"Led 3", StatusLed_3);
  Firebase.setFloat ( fbdo,"Led 2", StatusLed_2);
  Firebase.setFloat ( fbdo,"Led 1", StatusLed_1);
  strcat(ref, dateTime.c_str());
  Firebase.setFloat( fbdo,ref, random(0,230));
}
```

Hình 5: Gửi nhận thông tin firebase

2.2.3. MQTT và Adfruit IO

MQTT là một giao thức truyền thông theo mô hình Publish/Subscribe. Giao thức này được sử dụng cho các thiết bị IoT với băng thông thấp, khả năng sử dụng được trong mạng lưới không ổn định và độ tin cậy cao. MQTT dựa trên một Broker.

Adafruit.io là một dịch vụ đám mây kết nối thông qua internet cho phép các thiết bị IoT dễ dàng gửi và nhận dữ liệu.

Kết nối Adfruit IO với ESP8266:

- Bước 1: Khai báo biến thông tin cho Wifi và hai biến publishing, subscribing cho Adfruit IO.
- Bước 2: Kết nối wifi cho ESP8266.
- Bước 3: Thiết lập MQTT subscription
- Bước 4: Kết nối MQTT thông qua hàm MQTT_connect();
- Bước 5: Nhận dữ liệu từ MQTT thông qua hàm readSubscription, gửi dữ liệu đến MQTT thông qua hàm publish.

```
// Setup a feed called 'photocell' for publishing.  
// Notice MQTT paths for AIO follow the form: <username>/feeds/<feedname>  
Adafruit_MQTT_Publish pub = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/mqtt-server");  
  
// Setup a feed called 'onoff' for subscribing to changes.  
Adafruit_MQTT_Subscribe sub = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/mqtt-server");
```

Hình 6: Khai báo biến publishing, subscribing cho Adfruit IO

```
// Setup MQTT subscription for onoff feed.  
mqtt.subscribe(&sub);
```

Hình 7: Thiết lập MQTT subscription

```
Adafruit_MQTT_Subscribe *subscription;  
while ((subscription = mqtt.readSubscription(100))) {  
  if (subscription == &sub) {  
    Serial.print(F("Got: "));  
    Serial.println((char *)sub->lastread);  
    if (strcmp((char *)sub->lastread, "#turnOn") == 0) {  
      Serial.println("Turned ON");  
      pub->publish("#turnedOn");  
      hasTurnedOnLight1 = true;  
    }  
  }  
}
```

Hình 8: nhận dữ liệu MQTT

2.2.4. LoRa

LoRa là viết tắt của long-range là một công nghệ điều chế RF cho mạng diện rộng công suất thấp (LPWAN) có khả năng truyền dữ liệu lên đến 5km ở khu vực đô thị và 10-15km ở khu vực nông thôn. Đặc điểm của công nghệ Lora là yêu cầu điện năng cực thấp.

Hệ thống sử dụng Module RF SX1278 Lora E32 với tốc độ truyền: 0.3 – 19.2 Kbps (mặc định 2.4 Kbps), có tần số 410 – 441Mhz.

Module LoRa E32 có các chân sau:

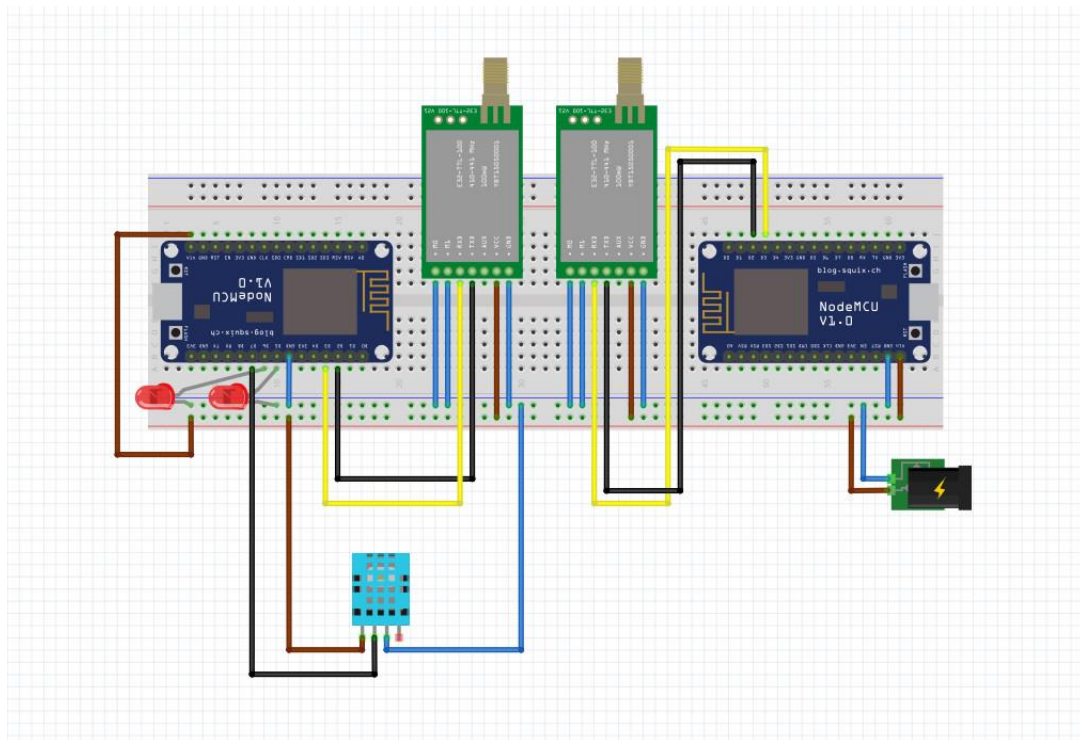
Pin No.	Pin item	Pin direction	Pin application
1	M0	Input (weak pull-up)	Làm việc với M1 & quyết định bốn chế độ hoạt động
2	M1	Input (weak pull-up)	Làm việc với M0 & quyết định bốn chế độ hoạt động
3	RXD	Input	Đầu vào TTL UART, kết nối với đầu ra TXD bên ngoài (MCU, PC).
4	TXD	Output	Đầu ra TTL UART, kết nối với RXD bên ngoài (MCU, PC) InputPin.
5	AUX	Output	Để chỉ ra trạng thái làm việc của mô - đun
6	VCC	Power supply 2.3V~5.5V DC	
7	GND	Ground	

Bảng 1: Các chân module LoRa E32

Mode	M1	M0	Explanation
Normal	0	0	UART và wireless đều mở
Wake-Up	0	1	Tương tự chế độ Normal nhưng có mã mở đầu được thêm vào dữ liệu truyền để đánh thức receiver.
Power-Saving	1	0	UART được vô hiệu hóa và không dây ở chế độ WOR (Wake On Radio) thiết bị sẽ bật khi có dữ liệu được nhận. Truyền bị vô hiệu hóa
Sleep	1	1	Truyền nhận bị vô hiệu hóa

Bảng 2: Các chế độ sử dụng module LoRa E32

Sơ đồ kết nối:

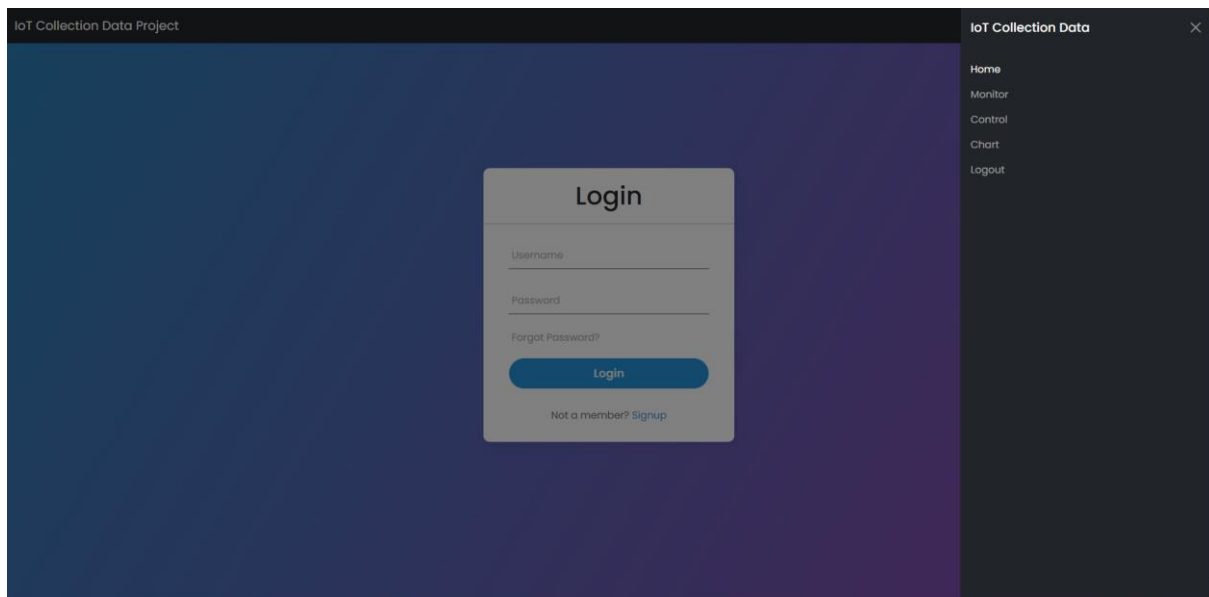


Hình 9: Sơ đồ kết nối

Kết nối LoRa E32 với ESP8266:

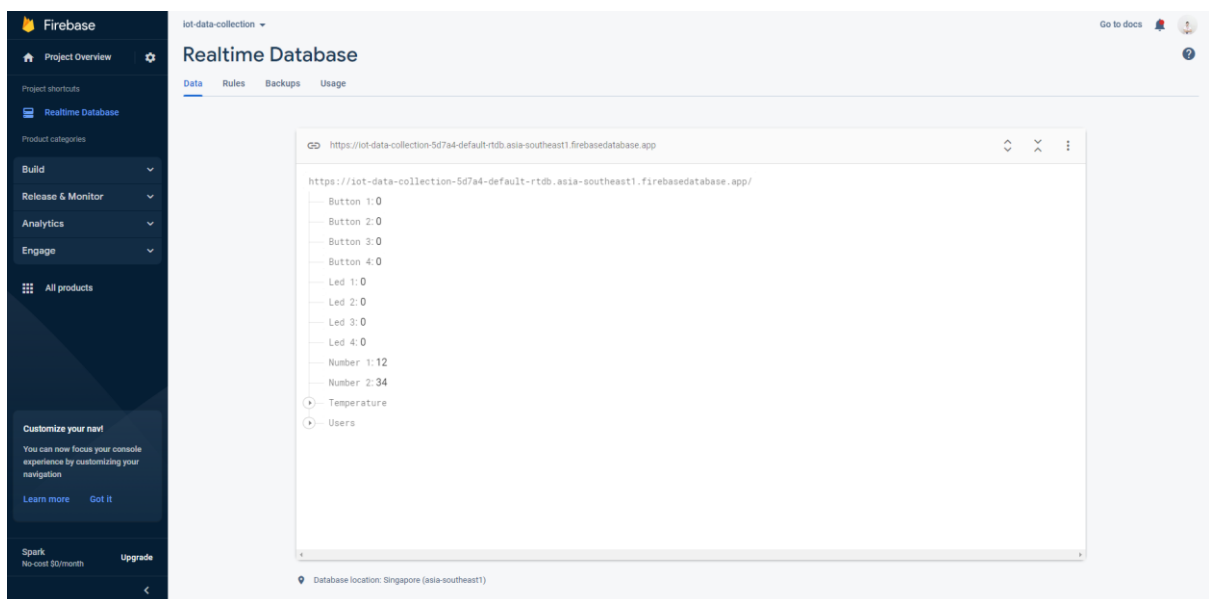
- Bước 1: Khai báo các chân rxPin, txPin, auxPin, m0Pin và m1Pin.
- Bước 2: Khởi động Serial, các chân đầu vào và đầu ra.
- Bước 3: Thiết lập module.
- Bước 4: Truyền/nhận dữ liệu.

3. Kết quả



Hình 10: Giao diện chính của website

Website có 4 tính năng chính: Đăng nhập/Đăng ký/Đăng xuất, Giám sát, Điều khiển, Biểu đồ.



Hình 11: Bảng dữ liệu Realtime Database

Firebase Realtime Database được sử dụng để lưu trữ trạng thái nút nhấn (Button), đèn (Led), nhiệt độ (Temperature) và thông tin người dùng (Users).

The screenshot shows the Adafruit IO MQTT server interface. At the top, there are buttons for '+ Add Data', 'Download All Data', and 'Filter'. Below these is a pagination bar showing 'page 1 of 43'. The main table has three columns: 'Created at', 'Value', and 'Location'. The table contains 13 rows of data. To the right of the table, there are several status indicators: 'You have 2131 data points from November 1st 2022, 4:27PM to November 22nd 2022, 8:33PM.', 'Notifications' (This feed is Online), 'Webhooks' (You have 1 receiving webhook active), 'Disable Feed', and 'License' (No Default License).

Created at	Value	Location
2022/11/22 8:33:56PM	false~false	
2022/11/22 8:33:54PM	#checkStatus	
2022/11/22 8:33:54PM	#turnedOffLight2	
2022/11/22 8:33:51PM	#turnOffLight2	
2022/11/22 8:33:51PM	#hasturnedOn	
2022/11/22 8:33:49PM	#checkStatusLight2	
2022/11/22 8:33:39PM	false~true	
2022/11/22 8:33:38PM	#checkStatus	
2022/11/22 8:33:38PM	#turnedOffLight2	
2022/11/22 8:33:36PM	#turnOffLight2	
2022/11/22 8:33:36PM	#hasturnedOn	
2022/11/22 8:33:34PM	#checkStatusLight2	
2022/11/22 8:33:32PM	true~true	
2022/11/22 8:33:30PM	#checkStatus	

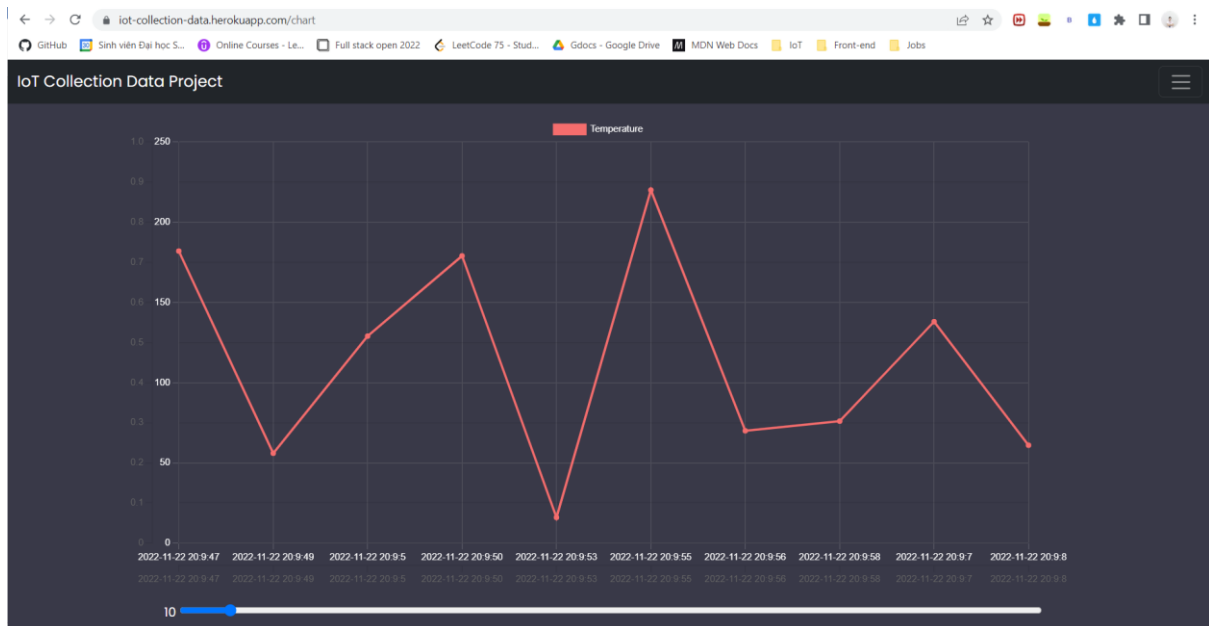
Hình 12: Bảng seed MQTT

Trang web adafruit io dùng để giao tiếp, truyền tín hiệu giữa web server với esp8266.

The screenshot shows the 'IoT Collection Data Project' website. It features a search bar with 'mm/dd/yyyy' and a 'Done' button. Below the search bar is a table with three columns: '#', 'Date', and 'Temperature'. The table contains 19 rows of data.

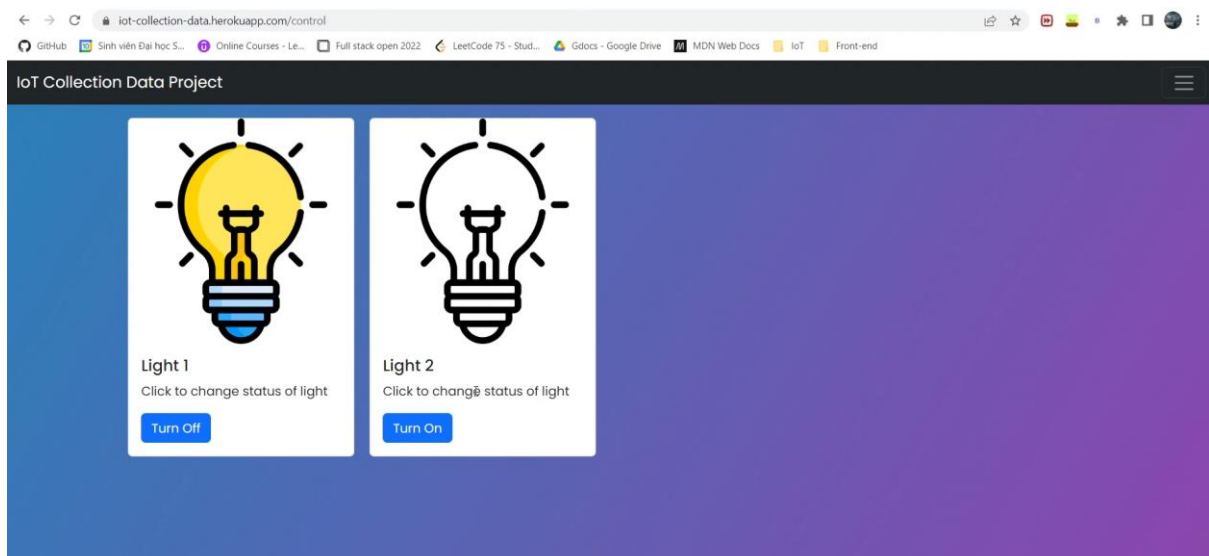
#	Date	Temperature
1	2022-11-22 18:16:42	120
2	2022-11-22 18:16:45	156
3	2022-11-22 18:16:46	65
4	2022-11-22 18:16:48	52
5	2022-11-22 18:16:49	223
6	2022-11-22 18:16:51	150
7	2022-11-22 18:16:53	163
8	2022-11-22 18:16:54	109
9	2022-11-22 18:16:56	41
10	2022-11-22 18:16:57	162
11	2022-11-22 18:16:59	53
12	2022-11-22 18:17:0	58
13	2022-11-22 18:17:11	82
14	2022-11-22 18:17:13	13
15	2022-11-22 18:17:14	1
16	2022-11-22 18:17:16	186
17	2022-11-22 18:17:18	224
18	2022-11-22 18:17:2	76
19	2022-11-22 18:17:20	131

Hình 13: Bảng dữ liệu nhiệt độ trên website



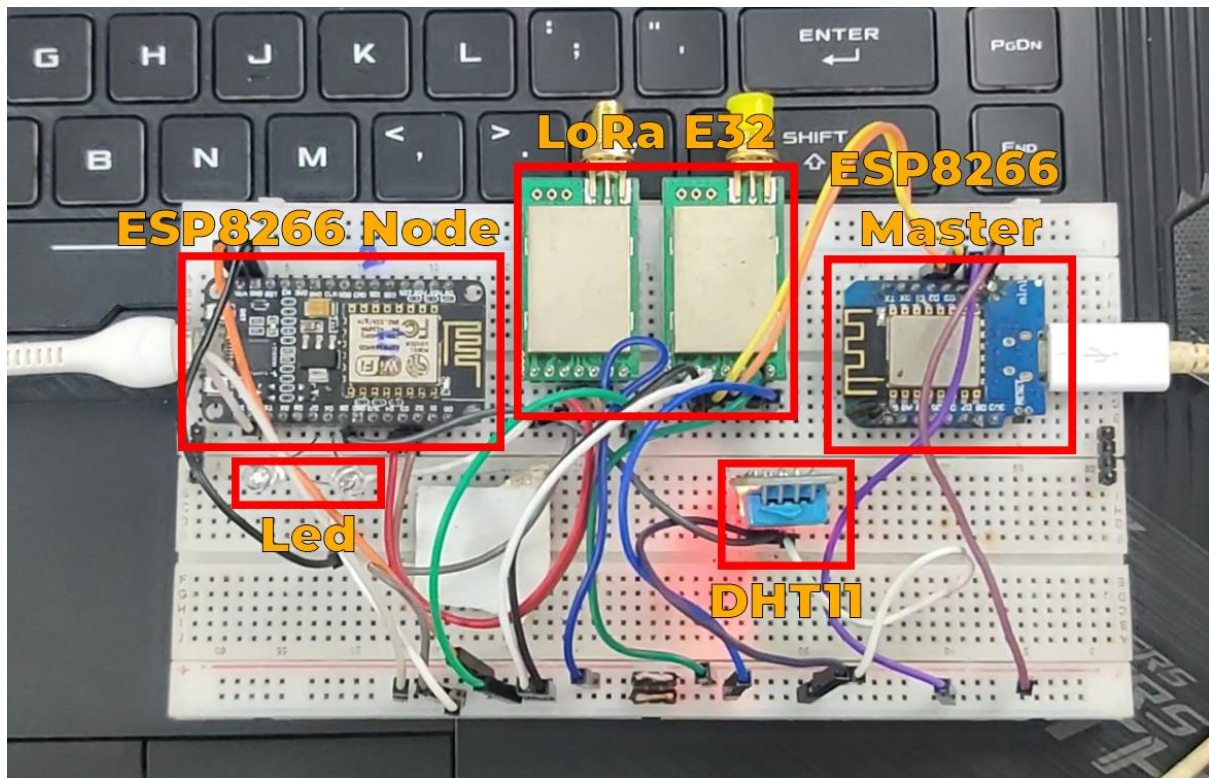
Hình 14: Đồ thị nhiệt độ trên website

Dữ liệu được thu thập từ các node sẽ được tải lên Firebase Realtime Database sau đó xử lý bởi NodeJS và hiển thị thông qua dạng bảng và biểu đồ.



Hình 15: Chức năng bật/tắt đèn trên website

Điều khiển bật tắt đèn từ web thông qua Adafruit IO MQTT đến hệ thống với message đã được thiết lập sẵn.



Hình 16: Hệ thống thực tế

Mạch thực tế hệ thống bao gồm 2 ESP8266, 2 Module Lora E32, 1 cảm biến DHT11, 2 bóng đèn led.

4. Kết luận và hướng phát triển

4.1. Kết luận

- Hệ thống truyền nhận tín hiệu nhanh chóng, ổn định với khoảng cách xa nhờ vào truyền thông LoRa.
- Hệ thống gửi dữ liệu lên server nhanh chóng và ổn định thông qua Firebase Realtime Database.
- Giao tiếp MQTT giữa hệ thống và server có độ trễ và đáp ứng không tốt.

4.2. Hướng phát triển

- Tối ưu web server, cơ sở dữ liệu để có thể mở rộng khi có lưu lượng truy cập cao.
- Tăng tốc độ truyền nhận và tính chính xác giữa các node thông qua LoRa.
- Tối ưu tính chính xác và nhanh chóng khi gửi nhận dữ liệu giữa web và các node thông qua MQTT.
- Tăng số lượng cảm biến, cơ cấu chấp hành và các node để mở rộng quy mô.