

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ



BÁO CÁO

LẬP TRÌNH THIẾT BỊ THÔNG MINH

**Đề tài: Tạo một thing sub nội dung lên internet sử dụng HIVEMQ
hiển thị lên LED P10.**

Sinh viên thực hiện:	Trương Ngôn Nghĩa	20002147
	Trần Thanh Phong	20002152
	Nguyễn Hoàng Anh	20002103
	Nguyễn Bá Kiên	20002140
Lớp:	K65 Kỹ thuật Điện tử và tin học	
Giảng viên hướng dẫn:	Nguyễn Cảnh Việt	

HÀ NỘI – 2024

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ



BÁO CÁO
LẬP TRÌNH THIẾT BỊ THÔNG MINH

**Đề tài: Tạo một thing sub nội dung lên internet sử dụng HIVEMQ
hiển thị lên LED P10.**

Sinh viên thực hiện:	Trương Ngôn Nghĩa	20002147
	Trần Thanh Phong	20002152
	Nguyễn Hoàng Anh	20002103
	Nguyễn Bá Kiên	20002140
Lớp:	K65 Kỹ thuật Điện tử và tin học	
Giảng viên hướng dẫn:	Nguyễn Cảnh Việt	

HÀ NỘI – 2024

MỤC LỤC

MỤC LỤC	3
PHÂN CHIA CÔNG VIỆC	4
I. Tổng quan về đề tài.....	5
1. Giới thiệu đề tài.	5
2. Mục tiêu của dự án:	5
3. Ứng dụng thực tế:	6
4. Lợi ích của dự án:	6
II. Các linh kiện và app sử dụng	7
1. Module Wifi ESP 8266.....	7
2. Module chuyển đổi USB-TTL UART CP2102.	8
3. IC LM1117T 3v3.	10
4. SR360	11
5. HiveMQ.....	12
6. Hivemq public broker.....	13
7. App myMQTT.	14
8. Led P10.....	15
III. Cấu hình Schematic, PCB.....	17
1. Cấu hình Schematic.....	17
2. PCB mô phỏng.	17
3. PCB thực tế.....	18
4. Code chương trình.	18
KẾT LUẬN	20
TÀI LIỆU THAM KHẢO	21

PHÂN CHIA CÔNG VIỆC

Tên thành viên	Công việc	Phần trăm công việc
Trương Ngôn Nghĩa	Code, Schematic, PCB, HiveMQ	35%
Trần Thanh Phong	HIVEMQ, MyMQTT, Schematic	25%
Nguyễn Bá Kiên	Tìm hiểu linh kiện, thông số	20%
Nguyễn Hoàng Anh	Slide, báo cáo, code	15%

I. Tổng quan về đề tài

1. Giới thiệu đề tài.

Giới thiệu Dự án Hiển thị Nội dung MQTT trên Màn hình LED P10 bằng ESP8266.

Trong kỷ nguyên của Internet of Things (IoT), việc kết nối và giao tiếp giữa các thiết bị trở nên ngày càng quan trọng và phổ biến. Một trong những giao thức truyền thông phổ biến và hiệu quả cho các ứng dụng IoT là MQTT (Message Queuing Telemetry Transport). MQTT là một giao thức nhẹ và dễ triển khai, giúp các thiết bị IoT truyền tải và nhận dữ liệu một cách nhanh chóng và hiệu quả.

Trong dự án này, chúng ta sẽ triển khai một ứng dụng thực tế sử dụng ESP8266 - một module WiFi mạnh mẽ và phổ biến, để kết nối và nhận dữ liệu từ một máy chủ MQTT (HiveMQ) và hiển thị nội dung nhận được lên một màn hình LED P10. Màn hình LED P10 là một loại màn hình ma trận LED phổ biến, thường được sử dụng để hiển thị thông tin trong các ứng dụng quảng cáo, thông báo công cộng và nhiều ứng dụng khác.

2. Mục tiêu của dự án:

Kết nối ESP8266 với mạng WiFi: Thiết lập kết nối WiFi cho ESP8266 để có thể giao tiếp với máy chủ MQTT.

Kết nối với máy chủ MQTT: Sử dụng thư viện PubSubClient để kết nối ESP8266 với HiveMQ và đăng ký một chủ đề cụ thể.

Nhận và xử lý dữ liệu từ MQTT: Khi có tin nhắn mới được gửi đến chủ đề đã đăng ký, ESP8266 sẽ nhận dữ liệu và xử lý.

Hiển thị nội dung lên màn hình LED P10: Sử dụng thư viện PAMatrix để điều khiển màn hình LED P10 và hiển thị nội dung nhận được từ MQTT.

3. Ứng dụng thực tế:

Bảng thông báo thông minh: Hiển thị thông tin từ xa như thời tiết, thông tin giao thông, thông báo công ty, hoặc bất kỳ thông tin nào từ máy chủ MQTT.

Quảng cáo số: Cập nhật và hiển thị nội dung quảng cáo từ xa một cách nhanh chóng và tiện lợi.

Giám sát từ xa: Hiển thị dữ liệu từ các cảm biến IoT như nhiệt độ, độ ẩm, chất lượng không khí, v.v.

4. Lợi ích của dự án:

Hiệu quả cao: Sử dụng MQTT giúp truyền tải dữ liệu nhanh chóng và đáng tin cậy.

Tiết kiệm chi phí: ESP8266 và màn hình LED P10 là các phần cứng có chi phí thấp, dễ dàng triển khai và mở rộng.

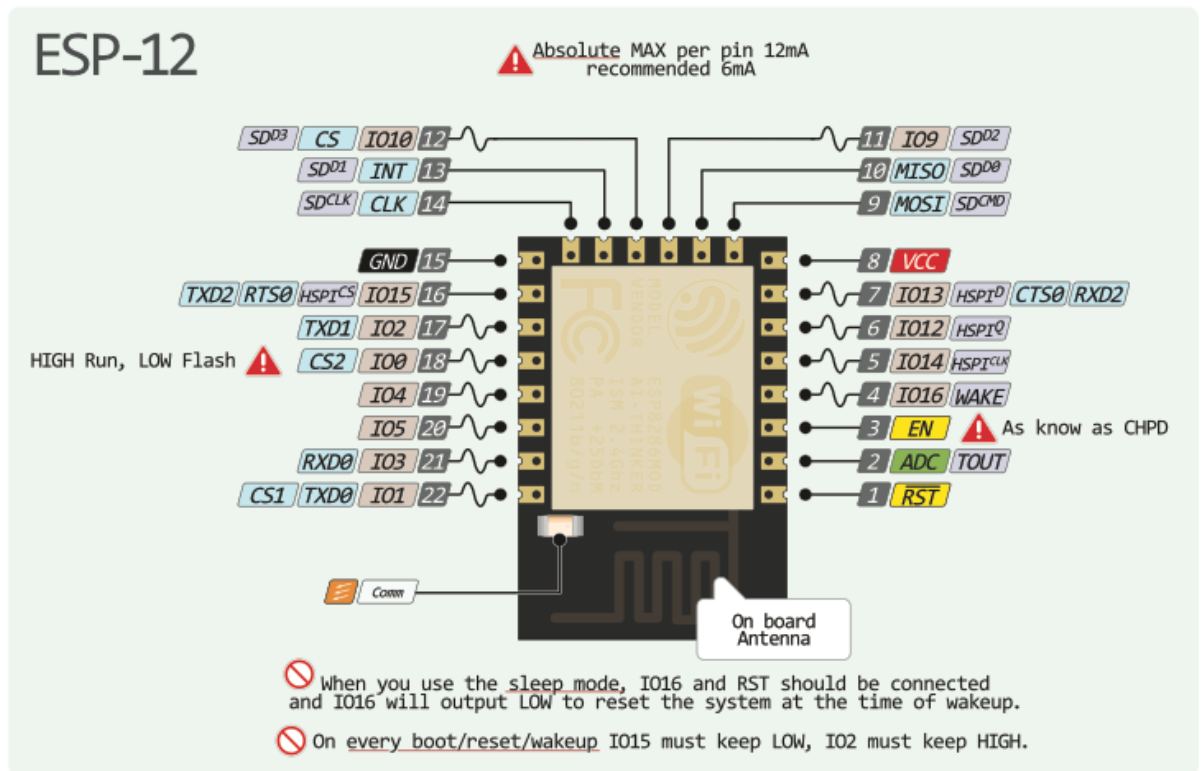
Dễ dàng tùy chỉnh: Dự án có thể dễ dàng tùy chỉnh để phù hợp với các nhu cầu hiển thị thông tin khác nhau.

Dự án này sẽ cung cấp một cái nhìn tổng quan và các bước chi tiết để thực hiện từ việc kết nối phần cứng đến viết mã và triển khai thực tế. Với sự kết hợp giữa ESP8266 và màn hình LED P10, bạn sẽ có một hệ thống hiển thị thông tin từ xa mạnh mẽ và linh hoạt, mở ra nhiều cơ hội cho các ứng dụng IoT sáng tạo và hữu ích.

II. Các linh kiện và app sử dụng

1. Module Wifi ESP 8266

ESP8266 là một chip vi điều khiển có khả năng kết nối Wi-Fi và thực hiện các ứng dụng IOT. ESP8266 có thể lập trình bằng nhiều ngôn ngữ khác nhau, nhưng phổ biến nhất là dùng Arduino IDE.



Thông số kỹ thuật:

- SoC: esp8288.
- Điện áp sử dụng: 3.0V ~ 3.6V
- Dòng điện làm việc: ~70mA (lớn nhất là 170mA), ở chế độ chờ < 200mA.
- Cường độ dòng điện giữ ở chế độ ngủ: 10uA.
- Số ngõ ra giao tiếp: 30 pins (10 GPIO đều có chức năng PWM, I2C, onewire).
- Tần số MCU: 80 - 160 MHz, 32-bit MCU.

- SRAM: 36 KB.
- ROM: 4 MB (SPI External Flash).
- Ăng ten tích hợp onboard.
- Tốc độ truyền nhận dữ liệu: 110 - 460800 bps.
- ADC độ phân giải 10bit (0 ~ 1V).
- Hỗ trợ WiFi @2.4GHz, bảo mật WPA/WPA2, hỗ trợ chuẩn 802.11 b/g/n.
- Hỗ trợ chế độ làm việc STA/AP/STA + AP3.
- Xây dựng trên giao thức TCP/IP, hỗ trợ tối đa 5 trạm.
- Nhiệt độ làm việc: -40°C ~ 125°C.
- Kích thước: 24 x 16 x 3 mm.

2. Module chuyển đổi USB-TTL UART CP2102.

Là module chuyển đổi tín hiệu USB sang tín hiệu Serial tuân tự theo chuẩn TTL. Mạch sử dụng chip CP2102.

CP2102 là chip USB to UART của Silabs. CP2102 có kích thước nhỏ gọn và yêu cầu rất ít thành phần bên ngoài để hoạt động được ngay. CP2102 không sử dụng thạch anh ngoài như các chip PL2303.

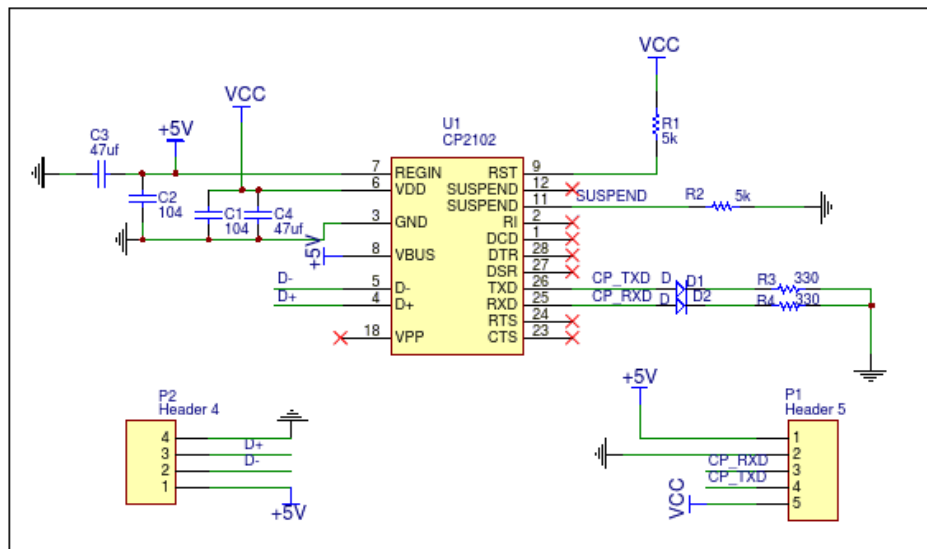
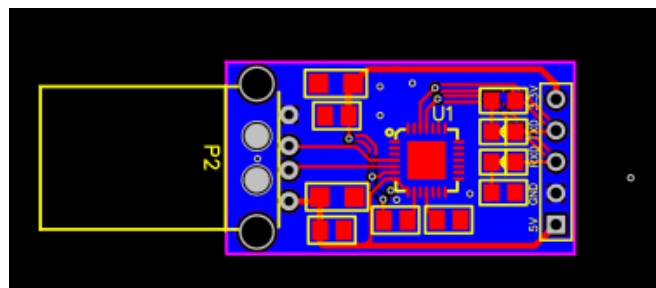
Module có sẵn ngõ ra điện áp 3.3V với dòng tải tối đa là 100mA.

Trên mạch có 5 cổng đầu ra: 3.3V 5V Tx Rx Gnd.



- Thông số kỹ thuật mạch chuyển USB to TTL UART:
- Phạm vi nhiệt độ: -40Cto + 85C
- Hỗ trợ windows vista / xp / server 2003/200, Mac OS-X / OS-9, Linux
- Màu chính: đỏ
- Kích thước: chiều dài (Không bao gồm USB): 30 mm
- USB để lấy nguồn, dẫn đến giao diện bao gồm 3,3V (<40mA), 5V, GND, TX, RX, mức pin tín hiệu là 3,3V, logic dương.
- Hỗ trợ tốc độ truyền trong khoảng 300bps ~ 1Mbps.

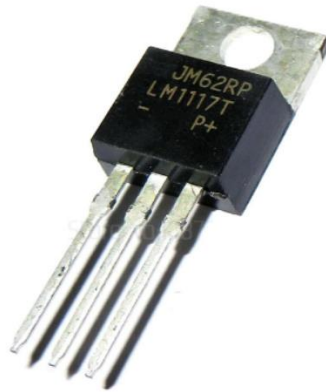
❖ Sơ đồ mạch USB to TTL UART:



3. IC LM1117T 3v3.

IC LM1117I 3v3 là ic chuyên dụng để ổn định điện áp tuyến tính với sụt áp thấp được dùng trong mạch điện tử.

IC ổn áp LM1117T 3.3V To220 chân cắm là một bộ điều chỉnh điện áp thả thấp với bỏ 1,2 V ở dòng tải 800 mA và có sẵn trong một phiên bản điều chỉnh điện áp ra 3.3V, cung cấp giới hạn dòng điện và nhiệt tắt.

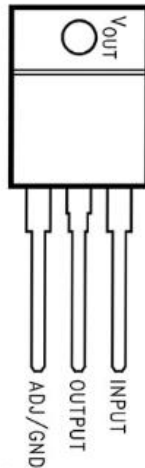


Thông số kỹ thuật:

- Tên sản phẩm: LM1117T-3.3V
- Điện áp đầu ra: 3.3V
- Điện áp đầu vào: 4.75V đến 20V
- Dòng điện đầu ra tối đa: 800mA
- Dropout Voltage (Điện áp giảm): Khoảng 1.2V tại dòng điện 800mA
- Quiescent Current (Dòng tĩnh): Khoảng 5mA
- Nhiệt độ hoạt động: -40°C đến +125°C
- Chức năng và ứng dụng:
- Chức năng chính: Ổn định điện áp đầu ra 3.3V từ điện áp đầu vào cao hơn.
- Ứng dụng: Cung cấp điện cho vi điều khiển, các mạch số, hệ thống nhúng, các mạch nguồn.

❖ Sơ đồ chân:

LM1117 3.3V Pinout



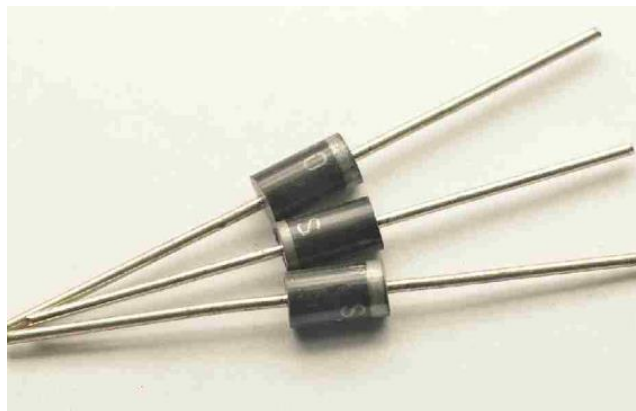
4. SR360

Là một loại diode thuộc dòng diode chỉnh lưu có độ phân giải cao.

Nó được sử dụng để bảo vệ mạch điện và các thiết bị khác khỏi những đỉnh điện áp ngược và làm giảm nhiễu.

Diode có điện áp đánh thủng tối đa là 600V, dòng điện định mức là 3A và có thể hoạt động ở nhiệt độ cao.

Nó được sử dụng trong các ứng dụng như nguồn cung cấp điện, bảo vệ quá dòng, bảo vệ quá áp và điều chỉnh mức độ điện áp.



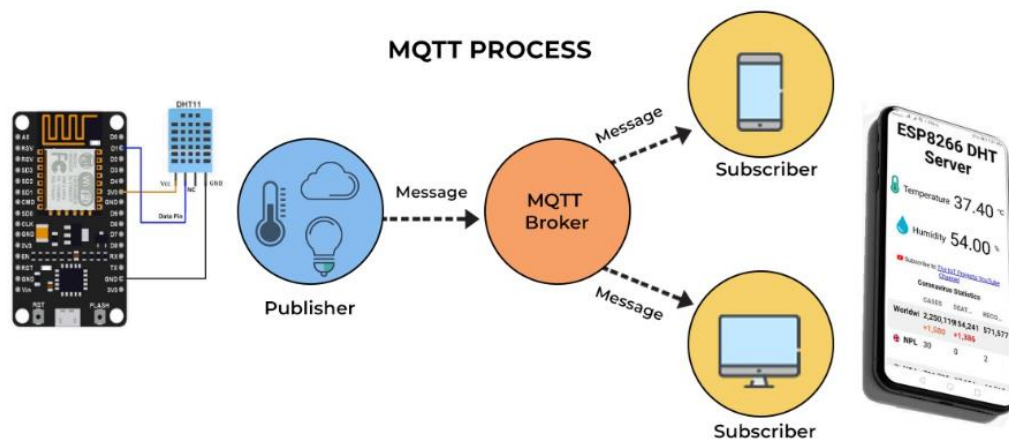
❖ Thông số kỹ thuật:

- Điện áp đánh thủng tối đa: 600V
- Dòng điện định mức: 3A
- Điện áp phân cực ngược tối đa: 600V
- Điện áp phân cực thuận tối đa: 1.2V
- Tốc độ đáp ứng nhanh
- Điện áp chuyển đổi thấp
- Có thể hoạt động ở nhiệt độ cao

5. HiveMQ

MQTT là viết tắt của Message Queuing Telemetry Transport, là một giao thức giao tiếp nhẹ dựa trên mô hình publish-subscribe. Giao thức này cho phép các thiết bị IOT gửi và nhận các tin nhắn nhỏ và đơn giản qua mạng Internet.

HiveMQ là một nền tảng MQTT, cung cấp giải pháp cho việc triển khai, quản lý và mở rộng hệ thống MQTT, giúp các tổ chức xây dựng ứng dụng IoT có khả năng mở rộng, tin cậy và an toàn.



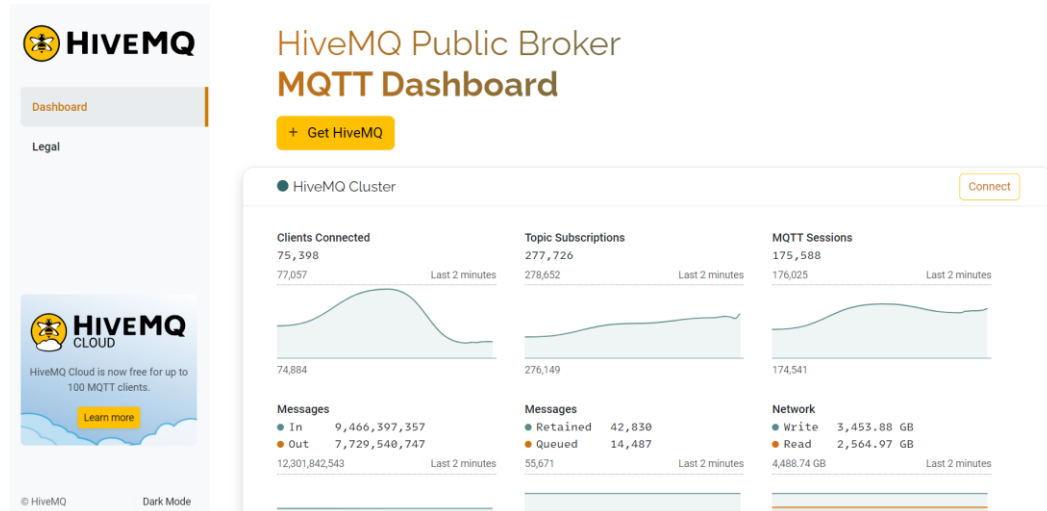
MQTT có ba thành phần chính: publisher, broker và subscriber. Publisher là thiết bị gửi tin nhắn đến broker, broker là máy chủ trung gian quản lý các tin nhắn và chuyển tiếp chúng đến subscriber, subscriber là thiết bị nhận tin nhắn từ broker.

MQTT sử dụng khái niệm topic để phân loại các tin nhắn. Mỗi publisher và subscriber có thể đăng ký hoặc hủy đăng ký các topic tùy ý. Broker sẽ chuyển tiếp các tin nhắn từ publisher đến subscriber dựa trên topic của tin nhắn.

6. Hivemq public broker

HiveMQ Broker chịu trách nhiệm cho việc quản lý kết nối, xử lý thông điệp, quản lý chủ đề, bảo mật dữ liệu và cung cấp các công cụ giám sát và quản lý trong hệ thống IoT sử dụng giao thức MQTT.

HiveMQ Public Broker là một dịch vụ cung cấp bởi HiveMQ GmbH cho phép các nhà phát triển và người dùng thử nghiệm và phát triển ứng dụng MQTT mà không cần triển khai một broker MQTT riêng.



7. App myMQTT.

myMQTT là một ứng dụng dành cho việc quản lý và giám sát các thiết bị IoT thông qua giao thức MQTT.

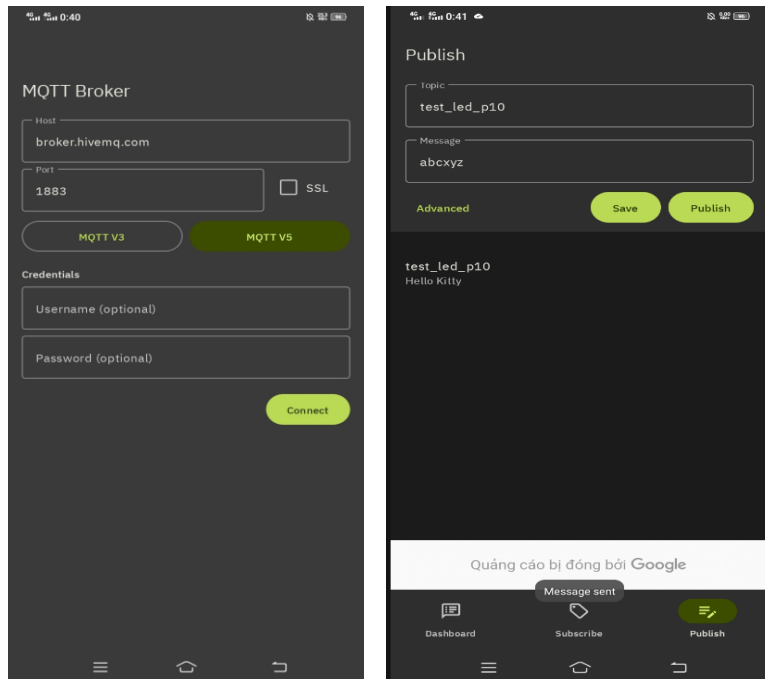
Các chức năng của app myMQTT:

Tạo và Quản lý Kết nối MQTT: Cho phép người dùng tạo và quản lý các kết nối MQTT đến các máy chủ MQTT khác nhau. Bạn có thể cấu hình thông tin như địa chỉ máy chủ, cổng, tên đăng nhập, mật khẩu và giao thức bảo mật (như SSL/TLS).

Thêm và Xóa Thiết bị IoT: myMQTT cho phép người dùng thêm và quản lý các thiết bị IoT thông qua giao thức MQTT. Bạn có thể nhập thông tin về tên thiết bị, chủ đề MQTT, cài đặt mức độ chất lượng dữ liệu (QoS), và nhiều thông số khác.

Giám Sát Trạng Thái Thiết bị: Hiển thị thông tin trạng thái của các thiết bị IoT, bao gồm dữ liệu cảm biến, trạng thái kết nối (online/offline), thông báo và tình trạng hoạt động.

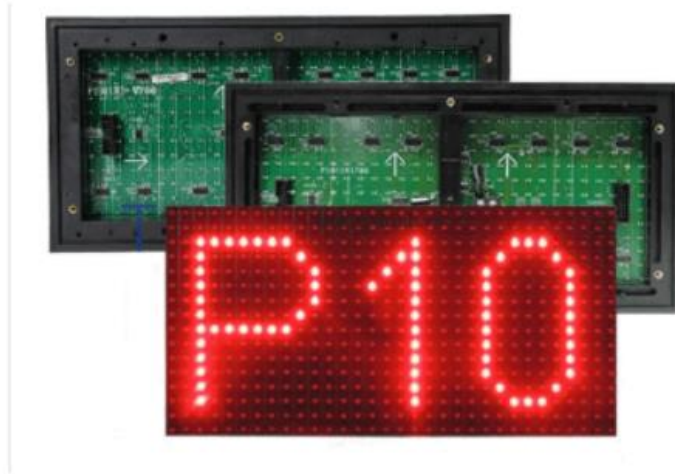
Điều Khiển Thiết bị: Cho phép người dùng thực hiện các thao tác điều khiển trên các thiết bị từ xa thông qua giao thức MQTT. Các thao tác này có thể bao gồm bật/tắt thiết bị, điều chỉnh các thiết lập, và thực hiện các hành động khác.



8. Led P10

LED ma trận P10 là một loại màn hình LED được cấu tạo từ nhiều module LED nhỏ ghép lại với nhau, tạo thành một màn hình lớn có khả năng hiển thị nội dung đa dạng.

Mỗi module LED P10 thường có kích thước 16cm x 32cm và có khoảng cách điểm ảnh là 10mm, giúp hiển thị rõ nét từ khoảng cách xa.



❖ Thông số kỹ thuật:

- Kích thước module: Thông thường là 320mm x 160mm
- Độ phân giải module: 32 x 16 pixel
- Màu sắc: Đỏ đơn sắc
- Khoảng cách điểm ảnh (pixel pitch): 10mm
- Độ sáng: Thường từ 800-1200 cd/m²
- Điện áp hoạt động: 5V DC.
- Điện áp đầu vào (DC): $4,5 \pm 0,1V$
- Tối đa hiện tại: $4,5 \pm 0,1V$
- Tiêu thụ điện tối đa: 6A/tấm

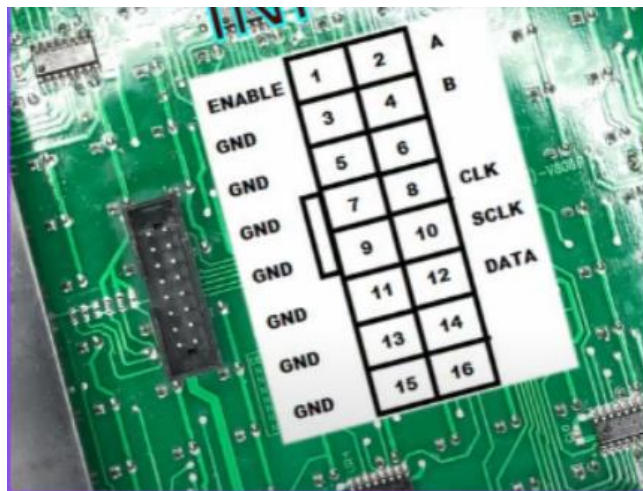
❖ Ưu điểm:

- Hiệu suất cao: Độ sáng cao giúp hiển thị tốt ngay cả dưới ánh sáng ban ngày.
- Tiết kiệm điện: Tiêu thụ năng lượng thấp
- Độ bền: Thường có tuổi thọ từ 50,000 đến 100,000 giờ
- Góc nhìn rộng: Giúp nội dung hiển thị rõ ràng từ nhiều góc độ khác nhau.

❖ Ứng dụng

- Biển quảng cáo ngoài trời: Thường được sử dụng để làm các biển quảng cáo LED cỡ lớn.
- Màn hình hiển thị thông tin: Dùng trong các bến xe, sân bay, nhà ga.
- Sự kiện và triển lãm: Dùng để hiển thị thông tin sự kiện, bảng thông báo
- Dễ dàng lắp đặt.
- Bảo trì thuận tiện.

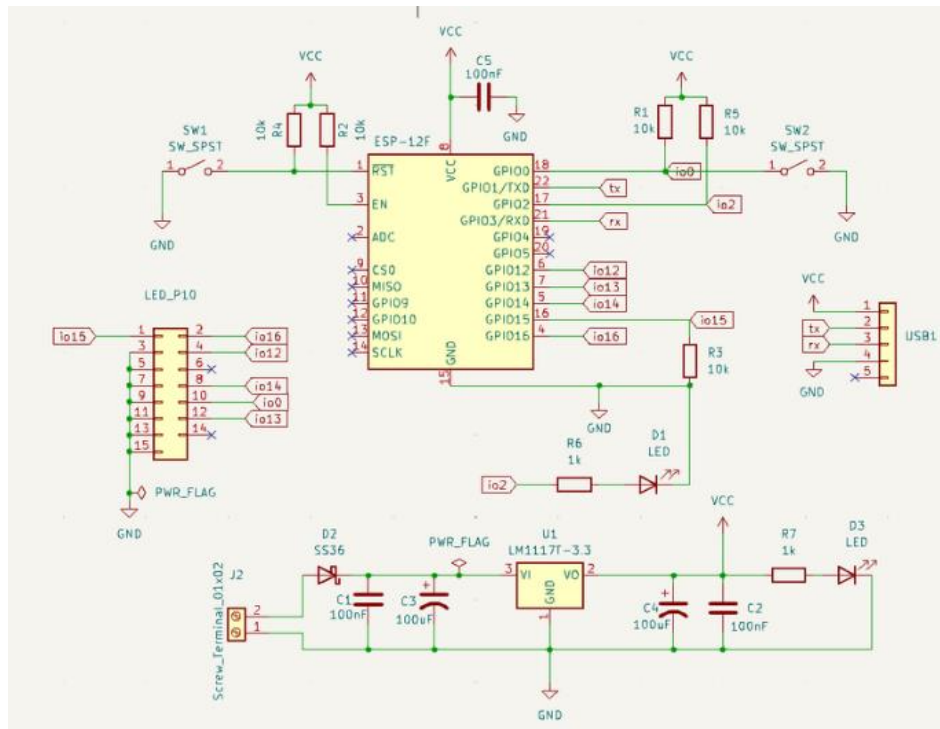
❖ Sơ đồ chân:



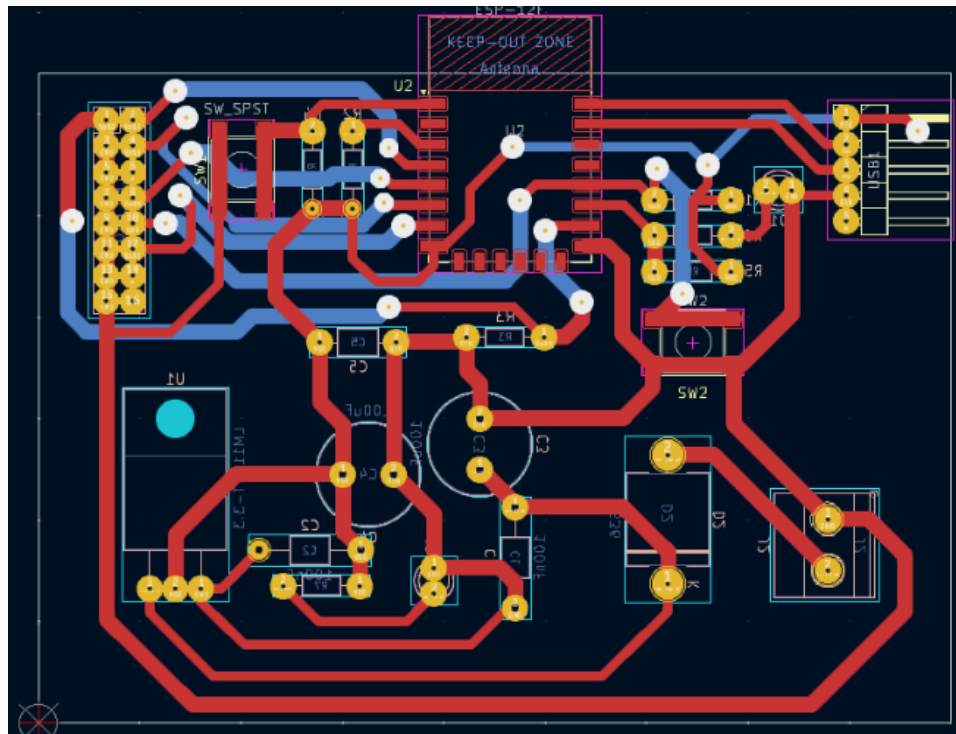
- Enable: Chân này được sử dụng để điều khiển độ sáng của bảng LED bằng cách cung cấp xung PWM cho nó.
- A, B: Đây được gọi là các chân chọn ghép kênh. Họ sử dụng đầu vào kỹ thuật số để chọn bất kỳ dòng ghép kênh A, B: Đây được gọi là các chân chọn ghép kênh. Họ sử dụng đầu vào kỹ thuật số để chọn bất kỳ dòng ghép kênh nào.
- Shift Clock (CLK), Serial Clock (SCLK) và Data: Đây là các chân điều khiển thanh ghi dịch chuyển thông thường.
- Ground: chân nối đất.

III. Cấu hình Schematic, PCB

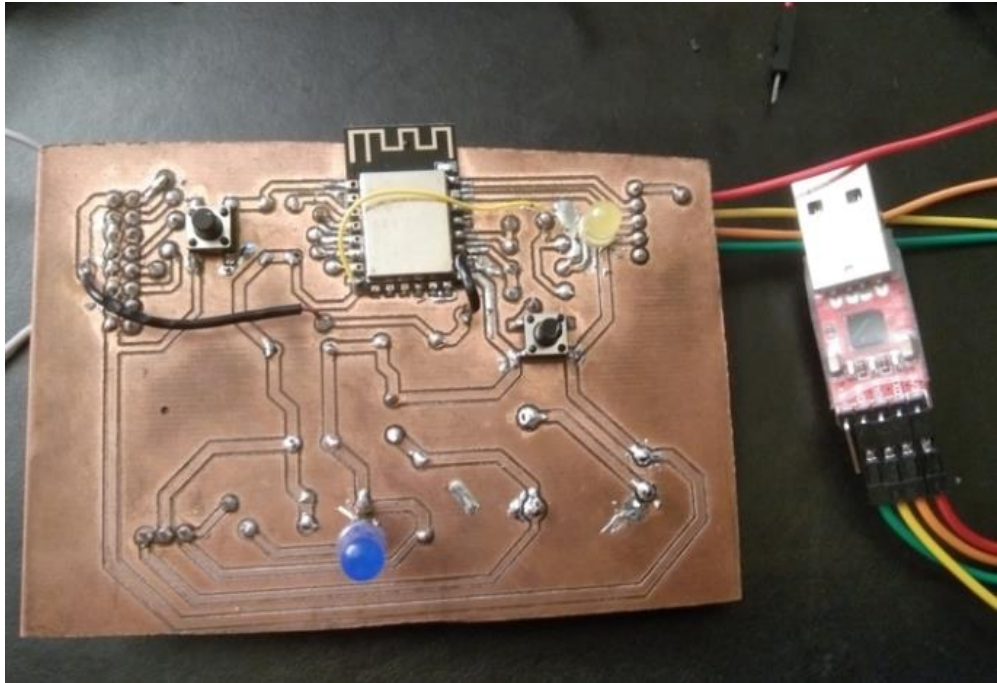
1. Cấu hình Schematic.



2. PCB mô phỏng.



3. PCB thực tế.



4. Code chương trình.

```
1  #include <ESP8266WiFi.h>
2  #include <PubSubClient.h>
3
4  #include <RoninDMD.h>
5  #include <fonts/Arial_Black_16.h>
6  #define FONT Arial_Black_16
7
8  #define WIDTH 1
9  #define HEIGHT 1
10 RoninDMD P10(WIDTH, HEIGHT);
11
12 const char *ssid      = "HUS"; // Wi-Fi
13 const char *password  = "12345678";
14 bool cInt_state = 0 ;
15
16 const char *mqtt_broker = "broker.hivemq.com"; // mqtt broker
17 const char *topic = "esp12f/buttonStatus"; // TOPIC !!!
18 const char *mqtt_username = "";
19 const char *mqtt_password = "";
20 const int mqtt_port = 1883;
21
22 String Message = "Hi";
23
24
25 WiFiClient espClient;
26 PubSubClient client(espClient);
```

```

28 void setup() {
29     Serial.begin(115200);
30
31     WiFi.begin(ssid, password);
32     while (WiFi.status() != WL_CONNECTED) {
33
34         delay(500);
35         Serial.println("Connecting to WiFi..");
36
37     }
38     Serial.println("Connected to the WiFi network");
39     //ket noi den 1 mqtt broker
40     client.setServer(mqtt_broker, mqtt_port);
41     client.setCallback(callback);
42     while (!client.connected()) {
43         String client_id = "TEST-2152";
44         client_id += String(WiFi.macAddress());
45         Serial.printf("The client %s connects to the public mqtt broker\n", client_id.c_str());
46         if (client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
47             Serial.println("mqtt broker connected");
48             clnt_state = 1;
49         } else {
50             clnt_state = 0;
51             Serial.print("Failed with state ");
52
53             Serial.print(client.state());
54             delay(2000);
55         }
56     }
57
58     P10.begin();          // font
59     P10.setFont(FONT);
60
61     P10.setBrightness(20); // do sang led
62
63     P10.drawText(0, 0, " : "); // P10.drawText(position x , position y, String type text);
64     client.subscribe(topic);
65     client.publish("dsp_ronin", "dsp++");
66 }

```

```

70 void callback(char *topic, byte *payload, unsigned int length) {
71     payload[length] = 0;
72     Message = String((char *) payload);
73     delay(100);
74     Serial.println(Message);
75 }
76
77 void Scrolling_text(int text_height , int scroll_speed , String scroll_text ) {
78     static uint32_t pM ;
79     pM = millis();
80     static uint32_t x = 0;
81     scroll_text = scroll_text + " ";
82
83     bool scr1_while = 1 ;
84     int dsp_width = P10.width();
85     int txt_width = P10.textWidth(scroll_text);
86
87     while (scr1_while == 1) {
88         P10.loop();
89         delay(1);
90         if (millis() - pM > scroll_speed) {
91             P10.setFont(FONT);
92             P10.drawText(dsp_width - x, text_height, scroll_text);
93             client.loop();
94             x++;
95             if (x > txt_width + dsp_width) {
96                 x = 0 ;
97                 scr1_while = 0 ;
98             }
99             pM = millis();
100         }
101     }
102 }

```

```

103 void loop() {
104     P10.loop();          // Run DWD loop
105     client.loop();        // tin nhan duoc gui nguoc lai
106     Scrolling_text(0, 50, Message ); // Call the function to write scrolling text on screen.
107                                     // like -> Scrolling_text( position y , scroll speed, String type text);
108                                     // or for not scroll -> P10.drawText(position x , position y, String type text);
109     delay(100);
110     if (!client.connected()) {
111         String client_id = "TEST-2152";
112         client_id += String(WiFi.macAddress());
113         Serial.printf("The client %s connects to the public mqtt broker\n", client_id.c_str());
114         if (client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
115             Serial.println("mqtt broker connected");
116             clnt_state = 1;
117             client.subscribe(topic);
118         } else {
119             clnt_state = 0 ;
120             Serial.print("Failed with state ");
121             Serial.print(client.state());
122             delay(2000);
123         }
124     }
125 }
126

```

KẾT LUẬN

Qua các bước triển khai chi tiết từ việc thiết lập kết nối WiFi cho ESP8266, kết nối với máy chủ MQTT, xử lý dữ liệu nhận được, cho đến việc hiển thị nội dung lên màn hình LED P10, chúng ta đã thấy rõ tiềm năng và tính linh hoạt của các công nghệ này trong việc tạo ra các hệ thống thông tin thời gian thực và tự động hóa.

Dự án này không chỉ minh họa cách sử dụng các công nghệ IoT một cách hiệu quả mà còn mở ra nhiều cơ hội phát triển các ứng dụng sáng tạo khác. Bằng cách tận dụng sức mạnh của MQTT và khả năng linh hoạt của ESP8266 cùng với màn hình LED P10, chúng ta có thể xây dựng các hệ thống thông minh, tự động hóa và dễ dàng tùy chỉnh để phục vụ nhiều mục đích khác nhau.

Với nền tảng và kiến thức đạt được từ dự án này, người dùng có thể tiếp tục mở rộng và phát triển thêm các tính năng, ứng dụng khác nhau dựa trên nhu cầu thực tế, góp phần vào sự phát triển của IoT và các giải pháp thông minh trong cuộc sống hàng ngày.

TÀI LIỆU THAM KHẢO

1. [HiveMQ Dashboard \(mqtt-dashboard.com\)](http://mqtt-dashboard.com)
2. [MQTT Publish/Subscribe Architecture \(Pub/Sub\) – MQTT Essentials: Part 2 \(hivemq.com\)](http://hivemq.com)
3. [MQTT Topics, Wildcards, & Best Practices – MQTT Essentials: Part 5 \(hivemq.com\)](http://hivemq.com)
4. [MQTT Publish, MQTT Subscribe & Unsubscribe – MQTT Essentials: Part 4 \(hivemq.com\)](http://hivemq.com)
5. [Hướng dẫn nạp Code cho ESP8266 sử dụng USB-TTL và Arduino IDE - TAPIT](#)
6. [Hướng Dẫn Lập Trình LED Ma Trận P10: Từng Bước Chi Tiết và Dễ Hiểu \(rdsic.edu.vn\)](http://rdsic.edu.vn)
7. [Tìm hiểu IC LM1117 \(dientutuonglai.com\)](http://dientutuonglai.com)