



esp8266

[LCD và GUI] Bài 2: Tìm hiểu màn hình LCD TFT 1.8 inch ST7735 (phần 1)

Bit Band trên STM32

[IoT] Bài 10: Cập nhật chương trình cho esp8266 từ xa qua internet (FOTA)

Bài 0: Làm quen và cài đặt proteus và KeilC 4.0 để lập trình cho các dòng chip 8051

Giao tiếp USB HID trên stm32 với c# window form

Nhúng và phát âm thanh trên STM32 với DMA và PWM

Giao tiếp với module thẻ từ RFID RC522



CHUYÊN MỤC

8051

Chưa được phân loại

Ethernet

IoT tutorial

Lập trình nhúng

LCD và GUI

Ma trận LED

Sản phẩm

Vi điều khiển

WIFI-ESP8266

PHẢN HỒI GẦN ĐÂY

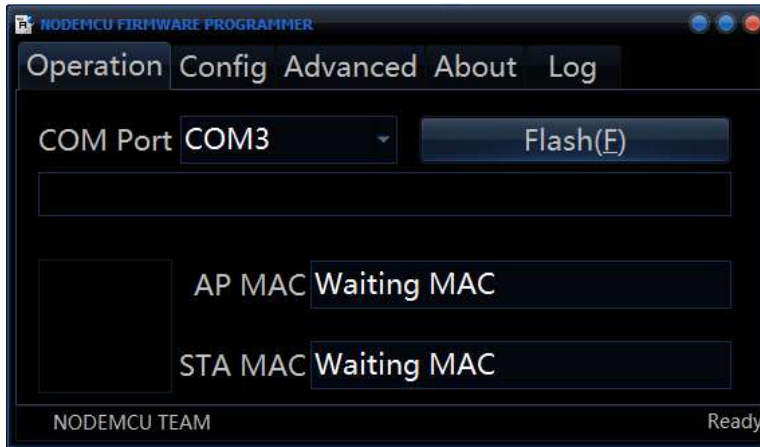
Khách trong [IoT] Bài 10: Cập nhật chương trình cho esp8266 từ xa qua internet (FOTA)

Khách trong Hiển thị đa ngôn ngữ trên màn hình lcd, matrix, thư viện Fontmaker.h

Khách trong [IoT] Bài 10: Cập nhật chương trình cho esp8266 từ xa qua internet (FOTA)

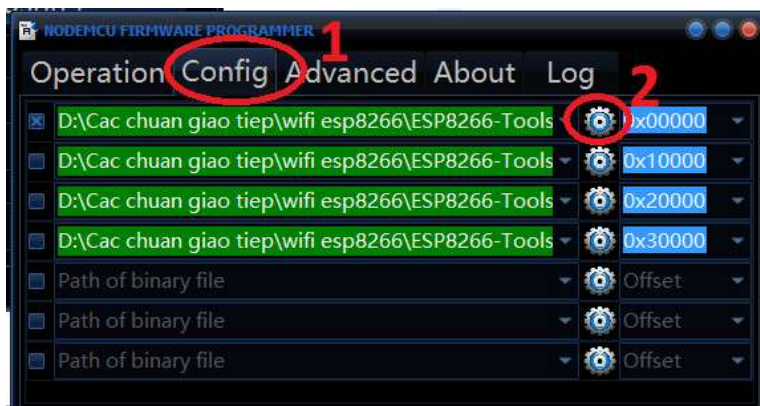
Khách trong [Matrix LED] Bài 4: Hiển thị chữ trên led ma trận 8x32

Tiến hành mở phần mềm nạp tên là **NODEMCU FIRMWARE PROGRAMMER**

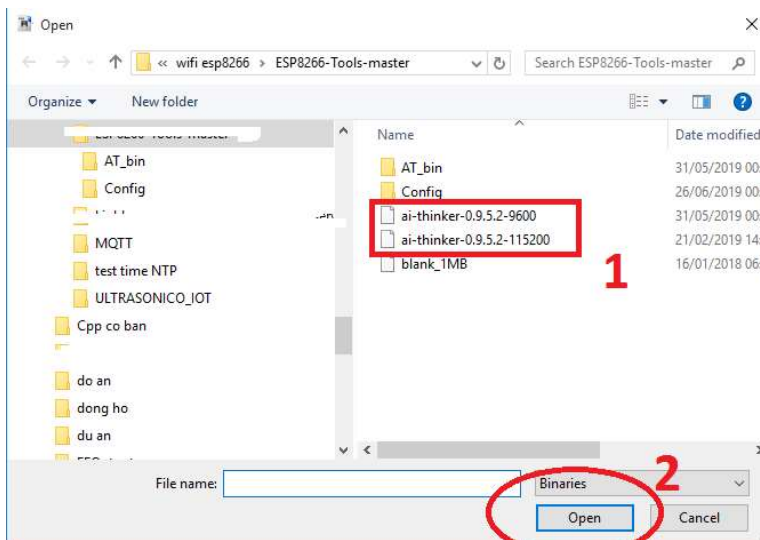


Đây là giao diện chính của phần mềm

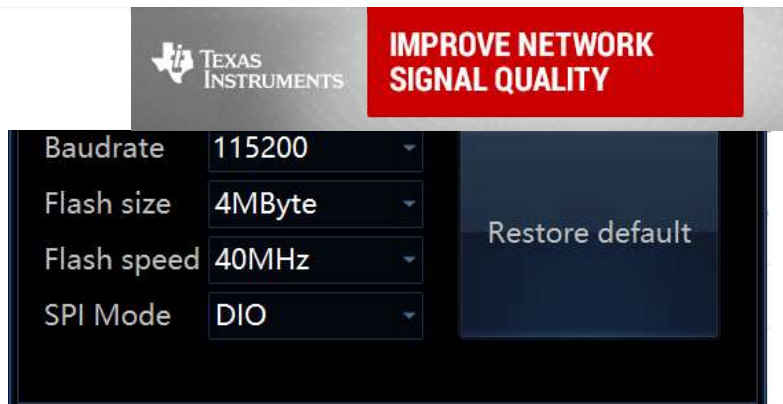
Chọn cổng COM tương ứng với module chuyển đổi USB-UART (thường thì phần mềm tự động lấy cổng, nhưng các bạn cũng nên kiểm tra lại trong My Coputer(chuột phải) -> Manage -> Device Manage -> Port (COM & LPT)



Chọn Config -> Ấn vào icon răng cưa để mở hộp thoại chọn file Firmwave



Chọn bất kì 1 trong 2 file firmware rồi ấn OK (chữ 9600 – 115200 chính là baudrate mà các bạn sẽ chọn để giao tiếp sau này). Nên dùng bản 115200 vì tốc độ sẽ nhanh hơn



LED ma trận P10 Full Color – lập trình led matrix full color

Khách trong [ENC28J60] Bài 7: Giao thức ARP (phần 2)

Khách trong [Matrix LED] Bài 6: Đồng hồ matrix 8x40 hiệu ứng lật trang đơn giản

Khách trong [Matrix LED] Bài 6: Đồng hồ matrix 8x40 hiệu ứng lật trang đơn giản

Khách trong [Matrix LED] Bài 6: Đồng hồ matrix 8x40 hiệu ứng lật trang đơn giản

Tiếp tục sang tab **Advanced** và chọn như trên



Quay về tab **Operation** và ấn Start để bắt đầu nạp firmware. Trong quá trình nạp các bạn có thể chuyển sang tab **Log** để xem thông báo nhé

Giao tiếp với module ESP8266 bằng các lệnh AT

Nhóm lệnh cấu hình hệ thống

Chúng ta sẽ chỉ cần quan tâm các lệnh cơ bản sau: (Chú ý **<CR><LF>** chính là 2 byte 0x0D 0x0A mà các bạn cần phải nhét vào sau mỗi lệnh AT để ESP8266 hiểu là kết thúc lệnh (sau đó nó mới phản hồi))

AT<CR><LF>

Phản hồi: OK (lệnh này để test)

ATE0<CR><LF>

Phản hồi: OK (lệnh này để tắt chế độ phản hồi, tức là bạn gửi gì thì nó trả về y hệt sau đó mới phản hồi, các bạn nên dùng lệnh này nhé)

AT+RST<CR><LF>

Phản hồi: OK (reset lại module)

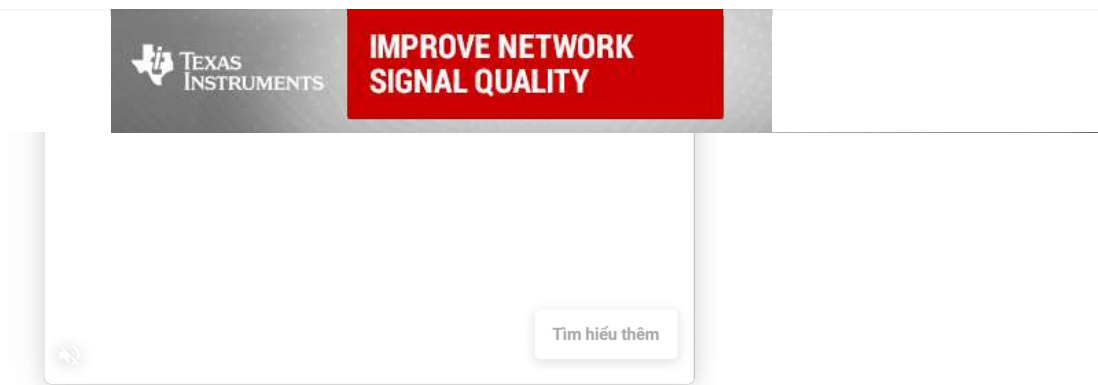
AT+WM<CR><LF>

Phản hồi: OK (Lưu cài đặt)

Nhóm lệnh giao tiếp

Trước khi đi vào các lệnh dưới thì mình sẽ giải thích qua 1 chút. ESP8266 có các chế độ hoạt động sau: AP, Station, AP+Station

Station : ESP sẽ đóng vai trò như 1 thiết bị mạng (giống như điện thoại của bạn). Nó phải truy cập vào WIFI nhà bạn. Các bạn có thể cho nó ping tới 1 trang web, hay truy cập vào 1 trang web như google chẳng hạn để tải toàn bộ mã HTML của nó về. Với chế độ này, các bạn có thể thuê 1 server (trang web) ở nhà cung cấp dịch vụ hay ở trang web cung cấp web free, sau đó cho ESP8266 truy cập vào để tải mã html có chứa trạng thái ON OFF của đèn led. Với cách này chúng ta có thể nhanh chóng tạo 1 ứng dụng điều khiển led đơn giản.

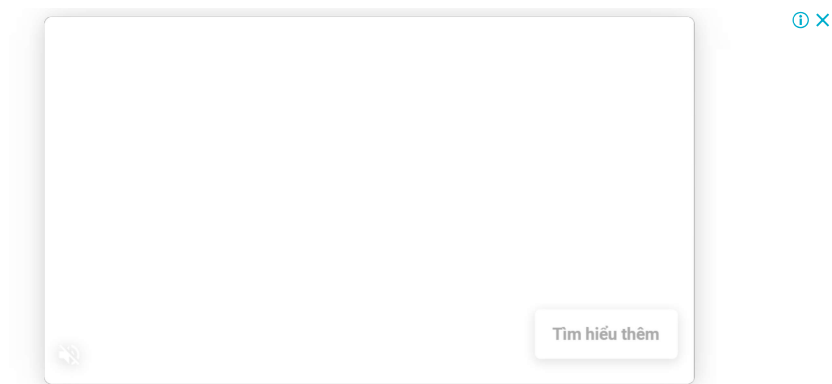


AP (Asset Point – Điểm truy cập) ESP có thể tự tạo ra 1 webserver (1 trang web) cho riêng nó để các thiết bị như điện thoại – laptop – hay 1 esp khác truy cập vào.

Nếu bạn dùng chế độ này khi ESP8266 phát WIFI, thì chỉ có thiết bị nào kết nối tới WIFI của esp mới truy cập vào được.

Nhưng nếu dùng chế độ này khi ESP sử dụng wifi nhà (WIFI do modem mạng phát ra) thì nó sẽ tạo ra 1 mạng có địa chỉ IP dạng 192.169.1.xxx (trong đó xxx là địa chỉ IP do modem cấp cho esp8266). Tạm thời nó cũng mới chỉ hoạt động cục bộ trong mạng LAN được thôi. Nhưng chúng ta có thể NAT PORT để đưa server do esp tạo ra ra ngoài mạng internet. Lúc này các bạn có thể truy cập vào esp8266 ở bất kì đâu.

Ngoài ra ESP8266 còn có thể hoạt động cùng lúc cả 2 mode Station và Asset Point



```
AT+CWJAP="IOT47","12345678"<CR><LF>
```

Phản hồi: WIFI CONNECTED

WIFI GOT IP

OK

(lệnh kết nối vào wifi IOT47 – mật khẩu 12345678. Các bạn tùy chỉnh 2 thông số này nhé)

Chế độ Station/Client

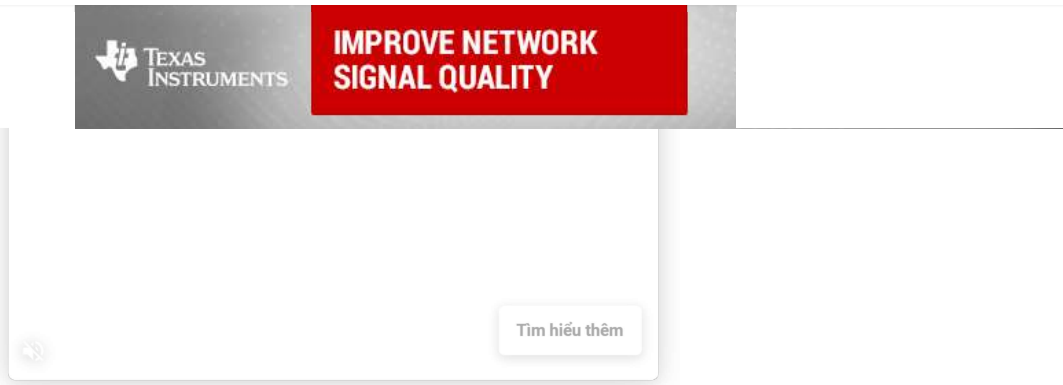
```
AT+CWMODE=1<CR><LF>
```

Phản hồi: OK (lệnh yêu cầu module hoạt động ở chế độ Station/Client)

Bây giờ thử cho ESP8266 upload dữ liệu nhiệt độ lên Server Thingspeak nhé !

```
AT+CIPMUX=0<CR><LF>
```

Phản hồi: OK (chế độ đơn kênh – nếu dùng mode Station thì cứ kích hoạt mode này cho đơn giản nhé)



AT+CIPSTART="TCP","thingspeak.com",80<CR><LF>

Phản hồi: CONNECT OK (khởi tạo giao thức TCP IP đến trang web thingspeak.com, cổng mặc định là 80)

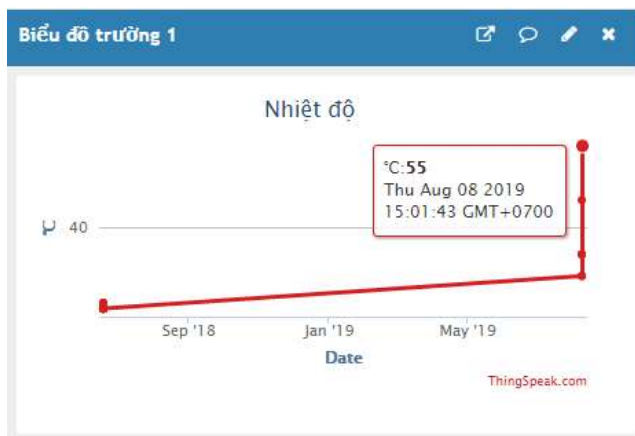
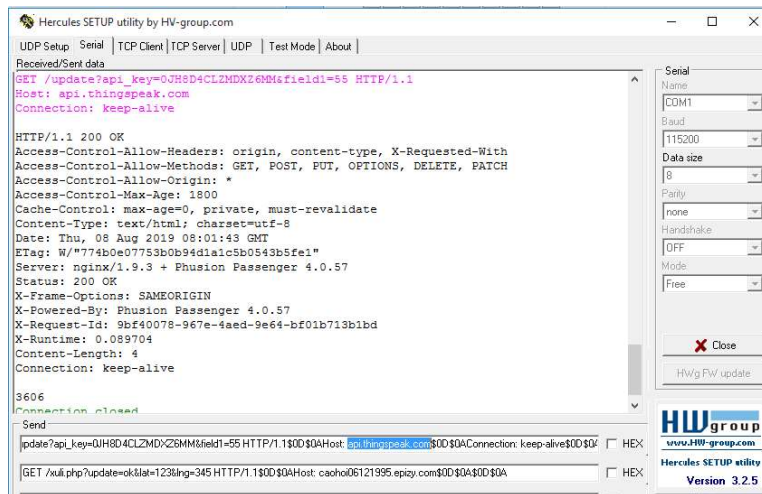
AT+CIPSEND=X<CR><LF>

Phản hồi: > (Chuẩn bị gửi dữ liệu đi, trong đó X chính là độ dài tính theo byte của đoạn dữ liệu sắp gửi)

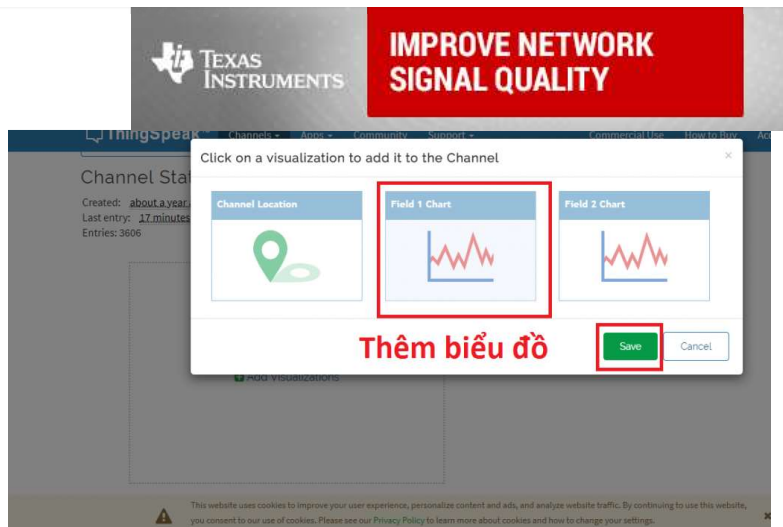
GET /update?api_key=0JH8D4CLZMDXZ6MM&field1=55 HTTP/1.1<CR><LF> Host: api.thingspeak.com<CR><LF> Connection: keep-alive<CR><LF><CR><LF>

Đây chính là dữ liệu mà ESP gửi lên cho thingspeak. Các bạn có thể thấy khóa API Key, và số 55 chính là dữ liệu nhiệt độ mà mình gửi lên. Đây gọi là truy vấn http (bản tin http hay Request HTTP) các bạn có thể tự tìm hiểu cấu trúc của nó trên mạng hoặc mình sẽ nói trong các bài sắp tới. (Hãy chắc chắn tính đúng độ dài của truy vấn để gửi đi trong lệnh CIPSEND nhé)

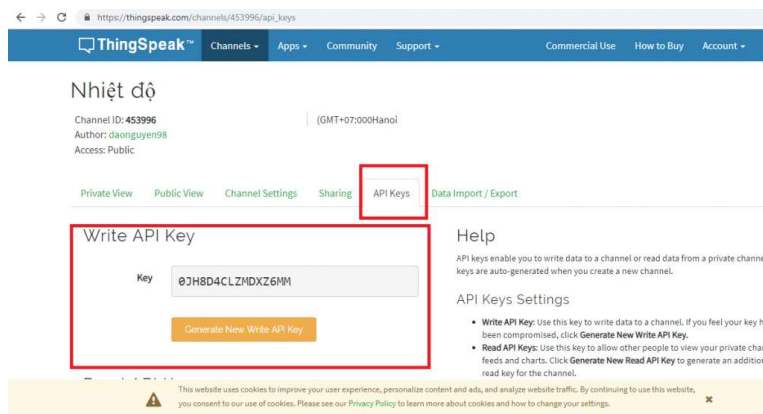
Phản hồi: SEND OK và chuỗi Reponse HTTP (các bạn có thể tham khảo chuỗi Reponse ở dưới ảnh)



Dữ liệu nhiệt độ được cập nhật lên biểu đồ của trang ThingSpeak



ADD biểu đồ vào



Cuối cùng là đóng kết nối, kết thúc 1 lần giao tiếp

`AT+CIPCLOSE<CR><LF>`

Chế độ Asset Point - Điểm truy cập

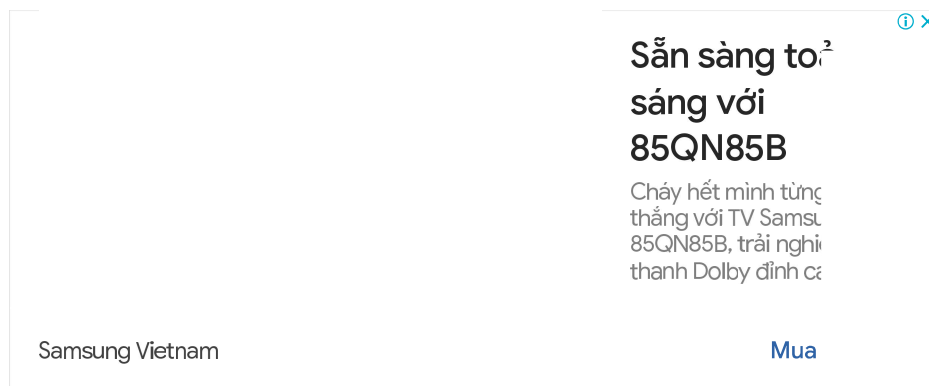
Bây giờ thử cho ESP8266 phát WiFi để điện thoại kết nối đến và cho ESP8266 tự tạo ra server web nhé !

`AT+CWSAP="IOT47","12345678",5,3<CR><LF>`

Phản hồi: OK (yêu cầu esp8266 phát WiFi tên là IOT47 và Pass là 12345678)

`AT+CWMODE=2<CR><LF>`

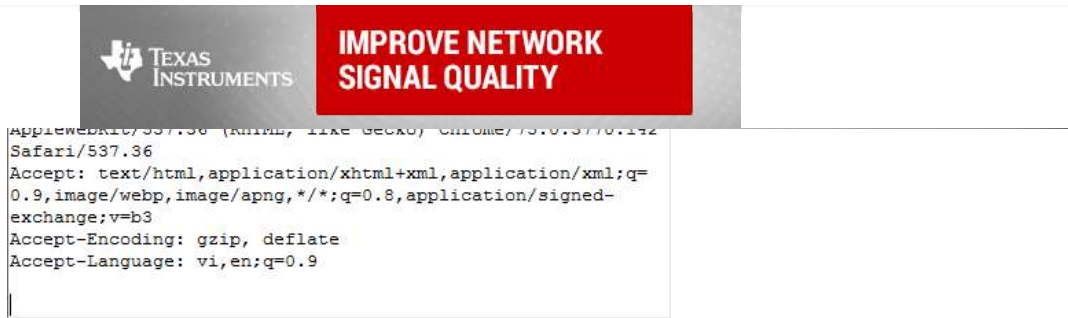
Phản hồi: OK (ESP8266 hoạt động ở chế độ AP)



`AT+CIPSERVER=1<CR><LF>`

Phản hồi: SERVER START (khởi động server, lúc này ta có 1 server hoạt động ở địa chỉ 192.168.4.1, bất kì thiết bị nào cùng mạng LAN đều có thể truy cập)

Lúc này, bạn có thể mở trình duyệt trên laptop hoặc điện thoại, gõ 192.168.4.1 (Enter). Trình duyệt sẽ gửi tới cho esp8266 1 truy vấn. Nó có dạng như này



Hmm... Nhìn cái đồng này khó chịu nhỉ. Thôi cứ tạm kệ nó đi nhé! Giờ hãy gõ lại vào chrome với địa chỉ như sau: 192.168.4.1/LEDON



Hãy nhìn vào phần mình khoanh đỏ. Từ khóa LEDON đã được gửi tới ESP8266. Bây giờ các bạn có thể bắt từ khóa này để bật led. Tương tự với việc tắt LED đi.

Quay lại trình duyệt web, nếu tính ý các bạn sẽ thấy trình duyệt vẫn đang loading, bởi vì nó đang đợi server trả lời. Lúc này hãy trả lời cho trình duyệt bằng lệnh **AT+CIPSEND**

AT+CIPSEND=X<CR><LF> (trong đó X là độ dài chuỗi mà bạn sẽ trả lời)

Ví dụ: **AT+CIPSEND=10<CR><LF>**

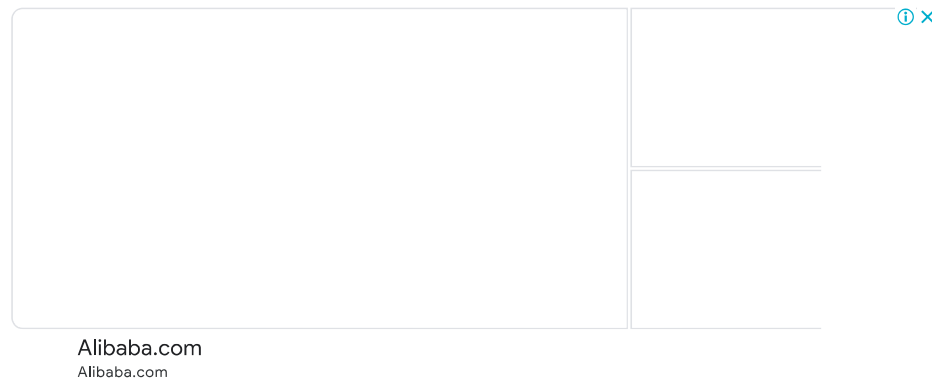
Phản hồi: >

Bây giờ gửi nội dung cần gửi:

Da tat led

Phản hồi: SEND OK

Sau đó có thể đóng kết nối: **AT+CIPCLOSE<CR><LF>**



Trình duyệt lúc này sẽ ngừng loading và trên màn hình hiện:



IMPROVE NETWORK
SIGNAL QUALITY

Đã tắt led

Related posts:

[ENC28J60] Bài 9: Giao thức ICMP

[IoT] Thực hành #1: Xây dựng hệ thống IoT hoàn chỉnh sử dụng AJAX, Http Request, PHP, MySQL

[ENC28J60] Bài 18: Giao thức HTTP Client, khám phá internet và gửi data lên thingspeak

[ENC28J60] Bài 1: Giới thiệu module ethernet enc28j60 - giao tiếp với ENC28J60

Thích

Chia sẻ

Hãy là người đầu tiên trong số bạn bè của bạn thích nội dung này.



ESP8266

IOT

Từ tác giả:

Nếu có bất kì thắc mắc nào trong bài viết, vui lòng để lại comment dưới mỗi bài ! Mình sẽ không trả lời thắc mắc của các bạn ở facebook hay email !

Nếu trong phần code bạn nhìn thấy những thứ kiểu như & thì đó là lỗi hiển thị, cụ thể 3 kí tự < > & bị biến đổi thành như thế
& là &
< là <
> là >



Giới thiệu Đào Nguyễn > 71 bài viết

DIY, chế cháo, viết blog chia sẻ kiến thức về lập trình, điện tử - IoT. Rất mong được giao lưu, kết bạn với các bạn cùng đam mê. Địa chỉ Facebook: <https://www.facebook.com/nguyendao207>

3 BÌNH LUẬN



Sỹ Hoàng

16 THÁNG MƯỜI MỘT, 2019 TẠI 8:53 CHIỀU

@@ Tks bác nhiều. e đg tìm tài liệu học mà đọc được cái này. Bác chỉ tận tình, chi tiết quá

↩ TRẢ LỜI





IMPROVE NETWORK
SIGNAL QUALITY

name to every , because it is actually legal to stream the footage

↩ TRẢ LỜI



Khách

25 THÁNG CHÍN, 2020 TẠI 11:16 SÁNG

Rất hay ạ. Cảm ơn anh

↩ TRẢ LỜI

Để lại bình luận

Nhập bình luận ở đây...

IOT47 BLOG

Blog chuyên chia sẻ kiến thức về IoT, lập trình nhúng, được lập vào tháng 8/2019 bởi Đào Nguyễn

LIÊN HỆ – HỖ TRỢ

✉ daonguyen20798@gmail.com
☎ 0394733311
F fb.com/nguyendao207

LIÊN KẾT

- ▶ Kênh DIY, làm mạch điện tử
- ▶ Kênh tổng hợp video từ blog
- ▶ https://qstech.vn/

ỦNG HỘ

Nếu thấy cái bài viết hữu ích, các bạn có thể ủng hộ 🍷 cho mình tại tài khoản:
VIETCOMBANK CHƯƠNG DƯƠNG
STK: 0541000289275
DAO VAN NGUYEN

