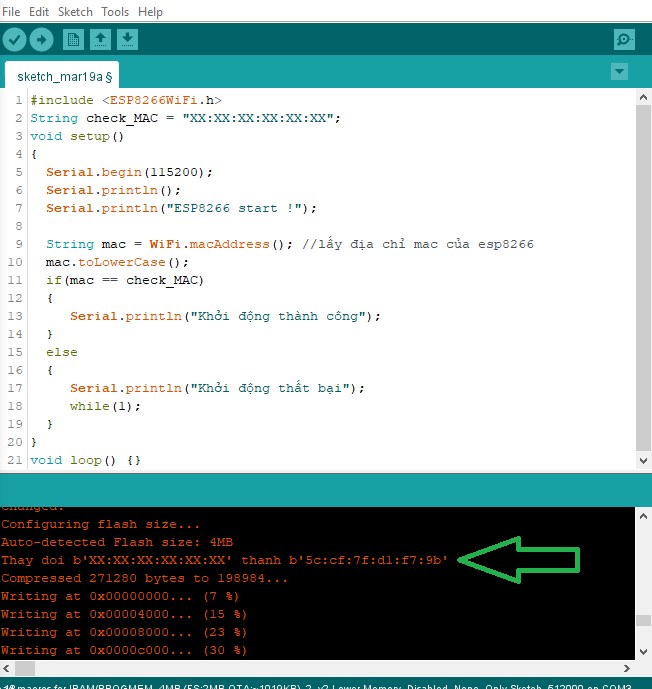
**Bảo mật chương trình cho esp8266**  
Chia sẻ cho các bạn 1 trick nhỏ để bảo mật firmware cho esp8266, với cách này, dù các bạn có gửi file .bin cho người khác thì cũng sẽ không chạy  
Ý tưởng: Do mỗi esp8266 có 1 mac riêng, nên trước khi khởi động chúng ta sẽ so sánh mac của chip với 1 đoạn mac được khai báo sẵn trong code, nếu không đúng thì không cho chạy code. Phép màu ở đây là chúng ta thêm 1 ít code python để sửa lại value trong biến check\_MAC thành mac của esp8266 mỗi khi upload code  
Cụ thể, khi bạn ấn biên dịch, 1 file .bin sẽ được tạo ra, đây chính là file code ở dạng mã máy của bạn. Arduino IDE sử dụng file **esptool.py**để upload, các bạn mở file này lên và chèn thêm vào 5 dòng code python như sau:  
Tìm hàm ***read\_mac***rồi copy 2 dòng này vào cuối hàm:  
*global user\_mac\_str*

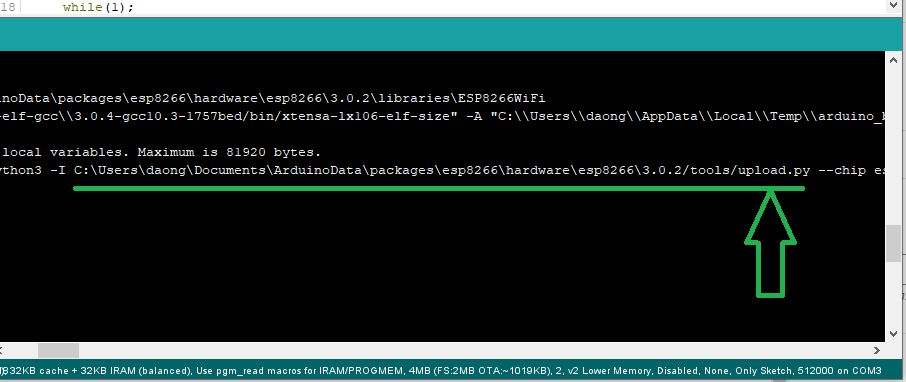
*user\_mac\_str = (':'.join(map(lambda x: '%02x' % x, mac))).lower()*  
  
Tìm hàm ***write\_flash***rồi copy 3 dòng này vào dưới lệnh  
**image = \_update\_image\_flash\_params(esp, address, args, image)**  
  
*in\_arduino\_string= b'XX:XX:XX:XX:XX:XX'*

*print("Thay doi %s thanh %s" % (in\_arduino\_string , user\_mac\_str) )*

*image = image.replace(in\_arduino\_string,bytes(user\_mac\_str, 'ascii'))*  
  
Sau đó ấn upload, dưới của số upload sẽ có thêm dòng thông báo thay đổi mac là thành công !  
  
Hạn chế: Từ khóa XX:XX:XX:XX:XX:XX không được trùng với data nào khác trong code, bạn có thể thay nó thành 1 từ khóa khác đặc biệt hơn để đảm bảo  
Ngoài ra, nên kết hợp với 1 thuật toán mã hóa MAC để "che" đi MAC của bạn và phối hợp so sánh thêm nhiều key khác nhau để làm cracker nản nòng khi dump code của bạn  
  
Chi tiết hơn các bạn có thể tham khảo tại đây:  
[http://iot47.com/bao-mat-chuong-trinh-cho-esp8266/](http://iot47.com/bao-mat-chuong-trinh-cho-esp8266/?fbclid=IwAR1cDDDiEydzZV66w6WnqiHC_aMyQRTpDKQbuZ-9nTmJBzLQVt_T2l7dCnU)

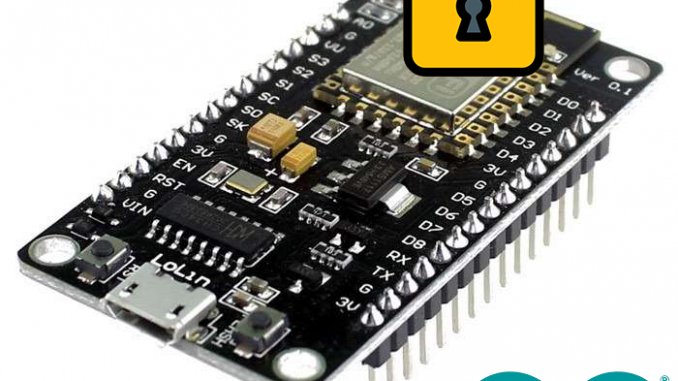


Các bạn có thể tìm thấy đường dẫn tới file esptool.py trong đoạn log in ra arduino ide mỗi khi upload chương trình. Nếu không thấy thì hãy bật log upload lên trong FILE -> Preferences và tích vào upload. FIle esptool.py nằm trong thư mục esptool cùng cấp với file upload.py



**Bảo mật chương trình cho esp8266**

[20 Tháng Ba, 2023](http://iot47.com/2023/03/) [Đào Nguyện](http://iot47.com/author/daonguyen/) [IoT tutorial](http://iot47.com/category/iot-tutorial/), [Lập trình nhúng](http://iot47.com/category/lap-trinh-nhung/), [WIFI-ESP8266](http://iot47.com/category/iot-tutorial/giao-tiep-esp8266/) [0](http://iot47.com/bao-mat-chuong-trinh-cho-esp8266/#mh-comments)

bảo mật code cho esp8266

Trong bài viết này, mình sẽ chia sẻ 1 trig nhỏ để bảo mật chương trình cho esp8266, với phương pháp này, dù bạn có gửi thẳng file .bin của bạn cho người khác thì họ nạp vào chip của họ cũng không chạy được

Các bạn theo dõi đoạn code sau



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | #include &lt;ESP8266WiFi.h>  String check\_MAC = "XX:XX:XX:XX:XX:XX";  void setup()  {    Serial.begin(115200);    Serial.println();    Serial.println("ESP8266 start !");      String mac = WiFi.macAddress(); //lấy địa chỉ mac của esp8266    mac.toLowerCase();    if(mac == check\_MAC)    {       Serial.println("Khởi động thành công");    }    else    {       Serial.println("Khởi động thất bại");       while(1);    }  }  void loop() {} |

Trong chương trình trên, khi khởi động mình sẽ so sánh mac của thiết bị với biến check\_MAC nếu giống nhau thì mới cho phép chạy

**PHÉP THUẬT**

Giờ hãy làm 1 chút ảo thuật nào

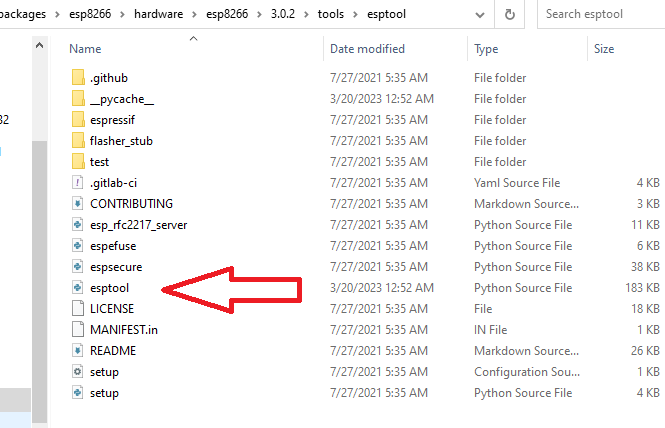
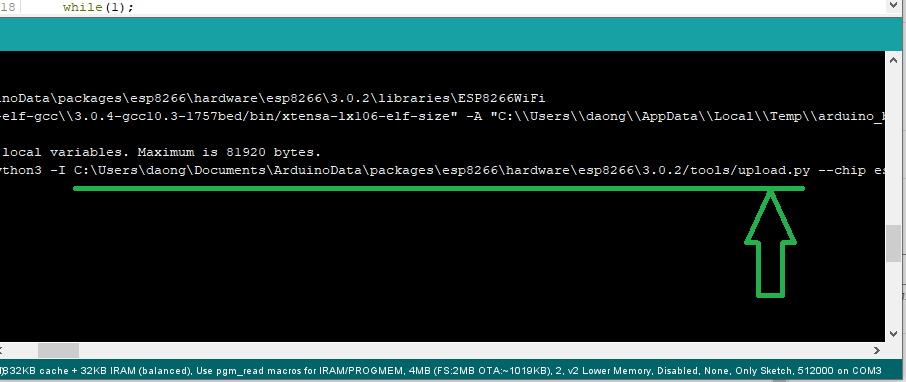
Khi bạn ấn nút nạp trên arduino, Arduino IDE sẽ sinh ra 1 file .bin, chính là file firmware của bạn đã được biên dịch thành mã máy. Bên trong file bin đó có chứa 1 đoạn dữ liệu chính là XX:XX:XX:XX:XX:XX

Sau khi file .bin được tạo thành công, arduino IDE sẽ nạp chương trình xuống cho esp8266 bằng 1 script

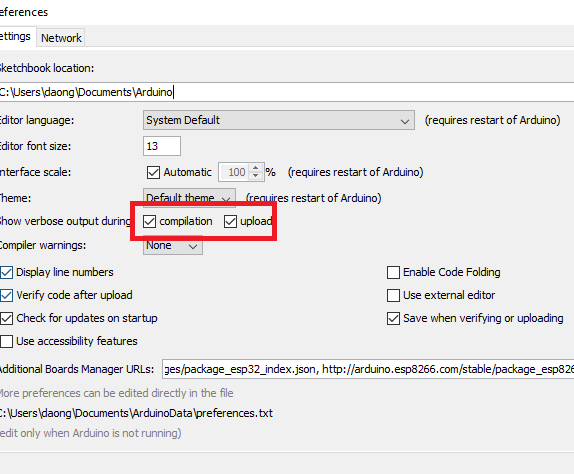
Cụ thể, arduino ide sử dụng 1 chương trình nạp được có tên là **esptool.py**. đây là 1 đoạn mã python, điều tuyệt vời là nó là mã nguồn mở, có nghĩa là ta có thể chèn thêm vào 1 vài dòng code để thêm chức năng cho nó.

Cụ thể, trong quá trình nạp, chúng ta sẽ đọc MAC của esp8266 lên và thay thế nó vào đoạn dữ liệu **XX:XX:XX:XX:XX:XX** trong file .bin kia. Nghe thì dòng dài nhưng chỉ cần thêm đúng 5 dòng code thôi 😀

Các bạn mở file **esptool.py** lên, có thể mở bằng nodepad hay bất kì phần mềm chỉnh sửa văn bản nào. Nếu không biết vị trí của file này ở đâu thì hãy ấn nút upload trên arduino IDE, trước khi nạp arduino IDE sẽ in ra 1 dòng lệnh, trong đó có chưa vị trí của file **upload.py**, file **esptool.py** nằm trong thư mục esptool cùng cấp với **upload.py**



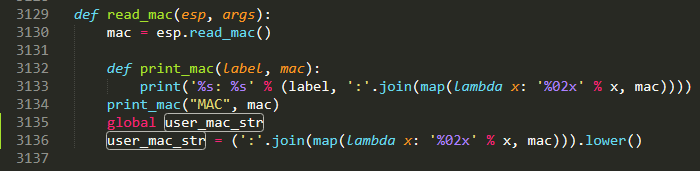
Nếu arduino IDE của bạn không hiển thị, thì là do bạn chưa bật chức năng show log upload, hãy vào **FILE -> Preferences** và tích vào **upload**



Khi đã mở được file **esptool.py** các bạn dùng chức năng tìm kiếm **( Ctrl + F )** để tìm hàm ***read\_mac*** sau đó chèn 2 dòng mã này vào cuối hàm



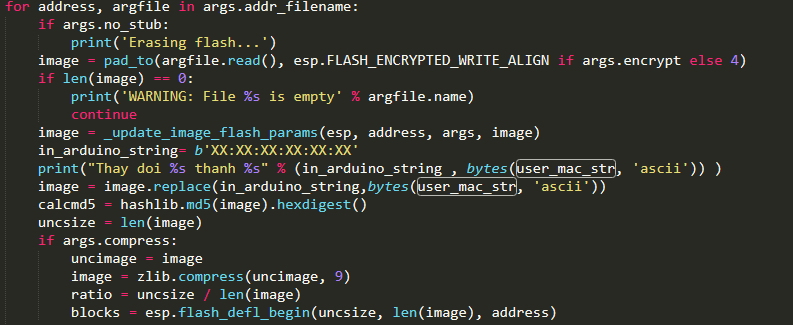
|  |  |
| --- | --- |
| 1  2 | global user\_mac\_str  user\_mac\_str = (':'.join(map(lambda x: '%02x' % x, mac))).lower() |



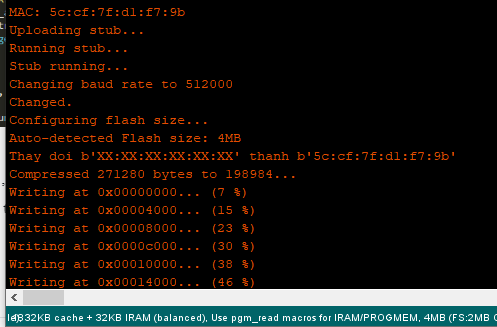
Tiếp tục tìm tới hàm ***write\_flash*** và chèn thêm 3 dòng code này vào dưới dòng ***image = \_update\_image\_flash\_params(esp, address, args, image)***



|  |  |
| --- | --- |
| 1  2  3 | in\_arduino\_string= b'XX:XX:XX:XX:XX:XX'  print("Thay doi %s thanh %s" % (in\_arduino\_string , user\_mac\_str) )  image = image.replace(in\_arduino\_string,bytes(user\_mac\_str, 'ascii')) |



Vậy là xong, lưu lại và nạp code thử , từ giờ khi upload trên arduino sẽ có in thêm dòng Thay doi **XX:XX:XX:XX:XX:XX** thành mac của chip là thành công



**Nâng cao khả năng bảo mật**

Việc lưu trực tiếp ID vào flash vẫn có thể bị hacker phát hiện ra khi dump code và phân tích, vì vậy hãy thêm 1 số thuật toán mã hóa để “che” đi MAC của bạn, thậm chí bạn có thể tạo thêm nhiều biến so sánh ở nhều nơi khác nhau trong chương trình để làm hacker “nản lòng”

**Hạn chế**

**XX:XX:XX:XX:XX:XX** là 1 value đặc biệt trong code của bạn. Do đó hãy đảm bảo trong chương trình của bạn không dữ liệu nào có thể chứa **XX:XX:XX:XX:XX:XX** hoặc bạn cũng có thể thay đổi khóa **XX:XX:XX:XX:XX:XX** thành 1 từ khóa đặc biệt khác khó bị trùng hơn