

Lớp Scan trong thư viện ESP8266WiFi

Giới thiệu về lớp Scan

Scan for Networks

scanNetworks

scanComplete

Hiển thị kết quả

SSID

scanDelete

encryptionType

channel

isHidden

RSSI

BSSID

getNetworkInfo

Giới thiệu về lớp Scan

Để kết nối smart phone tới một hotspot, ta mở chức năng setting Wifi trên smart phone, tìm Wifi cần kết nối và chọn kết nối. Với ESP8266, ta có thể làm điều tương tự như thế với hàm `WiFi.scanNetworks()` cho kết quả trả về là số lượng các WIFI (có mật khẩu và không có mật khẩu) mà module có thể kết nối được.

Scan for Networks

scanNetworks

Hàm scanNetworks thực hiện scan các Wifi trong vùng mà module có thể kết nối được và kết quả trả về là số lượng các WIFI (ta lập trình để kết quả hiển thị trên Serial Monitor).

```
WiFi.scanNetworks()
```

Ví dụ:

```
#include <ESP8266WiFi.h>

void setup()
{
    Serial.begin(115200);

    Serial.println("** Scan Networks **");

    int numSsid = WiFi.scanNetworks();

    Serial.print("SSID List:");
    Serial.println(numSsid);

}
void loop()
{
}
}
```

Output:

The screenshot shows the Arduino IDE interface. The left pane displays the code from the previous block. The right pane shows the serial output, which matches the code's print statements: "** Scan Networks **" followed by "SSID List:13". At the bottom, a status bar indicates "Done uploading." and "Uploading 229936 bytes from /tmp/arduino_bui...".

scanComplete

Hàm `WiFi.scanComplete()` có chức năng tương tự như hàm `scanNetworks` nhưng thực hiện các chức năng scan bất đồng bộ (asynchronous scanning)

```
WiFi.scanComplete()
```

Kết quả trả về số lượng các mạng Wifi có thể kết nối được (giống như với hàm scanNetworks)

Nếu việc scan chưa thực hiện xong thì có các kết quả trả về:

- Hàm scan chưa thực hiện xong: -1
- Lỗi scan: -2

Ví dụ:

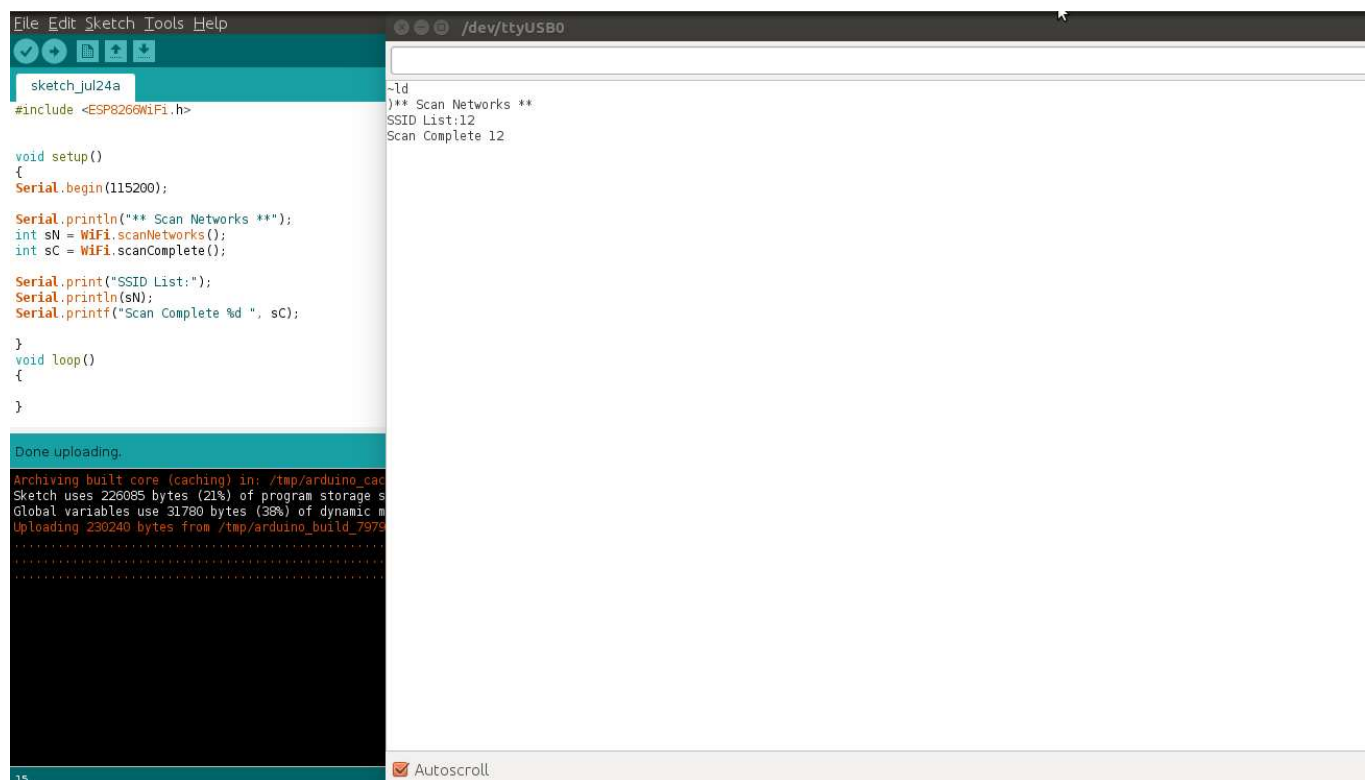
Thực hiện chạy song song 2 hàm scanNetworks và scanComplete, kiểm tra, so sánh kết quả.

```
#include <ESP8266WiFi.h>

void setup()
{
  Serial.begin(115200);
  Serial.println("** Scan Networks **");
  int sN = WiFi.scanNetworks();
  int sC = WiFi.scanComplete();
  Serial.print("**SSID List:**");
  Serial.println(sN);
  Serial.printf("Scan Complete %d ", sC);
}

void loop()
{
}
```

Output:



The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar contains icons for opening, saving, and running the sketch. The sketch is named 'sketch_jul24a'. The code in the editor is the same as shown in the previous block. The serial monitor is open, showing the output:
~ld
)** Scan Networks **
SSID List:12
Scan Complete 12
Below the code editor, a status bar shows 'Done uploading.' and a message: 'Archiving built core (caching) in: /tmp/arduino_cache... Sketch uses 226085 bytes (21%) of program storage space. Global variables use 31780 bytes (38%) of dynamic memory. Uploading 230240 bytes from /tmp/arduino_build_7979...'. At the bottom, there is a checkbox for 'Autoscroll' which is checked.

Hiển thị kết quả

SSID

Sau khi scan network, kết quả trả về là số các network. Để biết SSID của network với số thứ tự tương ứng trong kết quả về, ta dùng hàm `WiFi.SSID(arg 1)`, với arg 1 là networkItem la2 tham số truyền vào là số thứ tự của network cần biết SSID.

```
WiFi.SSID(networkItem);
```

Returned Kết quả trả về là kiểu string, nên ta dùng chuyển đổi: `WiFi.SSID(1).c_str()`

Ta lưu ý là kết quả tên mạng SSID trả về được tính bắt từ 0, tức là giả sử có 10 mạng Wifi thì số thứ tự các mạng sẽ là từ 0 đến 9.

Ví dụ:

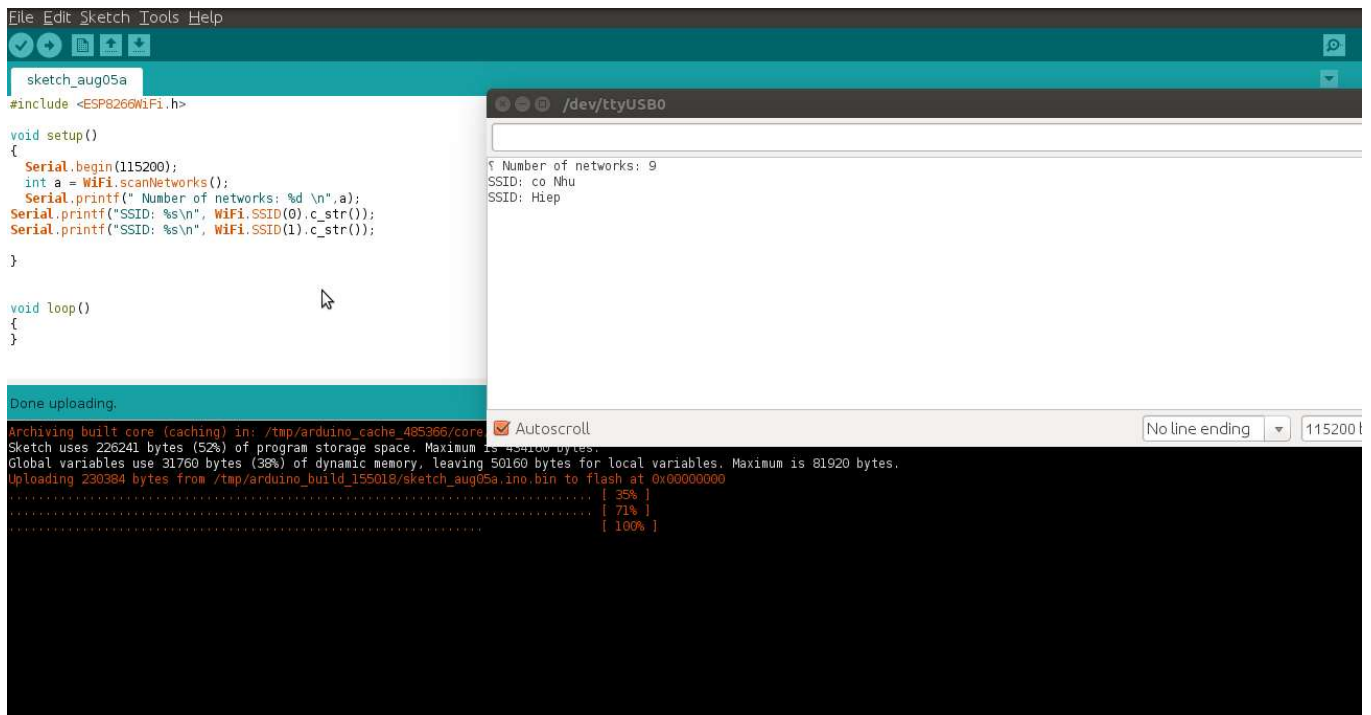
Ta cần trả về tên của 2 mạng WIFI thứ tự 0 và 1

```
#include <ESP8266WiFi.h>
void setup()
{
    Serial.begin(115200);
    int a = WiFi.scanNetworks();
    Serial.printf(" Number of networks: %d \n",a);
    Serial.printf("SSID: %s\n", WiFi.SSID(0).c_str());
    Serial.printf("SSID: %s\n", WiFi.SSID(1).c_str());
}
```

```
<
}

void loop() { }
```

Output:



scanDelete

Xóa kết quả của lần scanNetworks trước khỏi bộ nhớ, do đó kết quả sau khi scan sẽ không thể hiển thị trên màn hình Serial Monitor

```
WiFi.scanDelete()
```

Ví dụ:

Ở đoạn code trước ở phần trước của câu lệnh `WiFi.SSID`, ta thêm dòng lệnh `WiFi.scanDelete();` ở cuối đoạn code phần setup, thì sẽ không có kết quả nào hiển thị trên Serial Monitor, màn hình Serial Monitor trống.

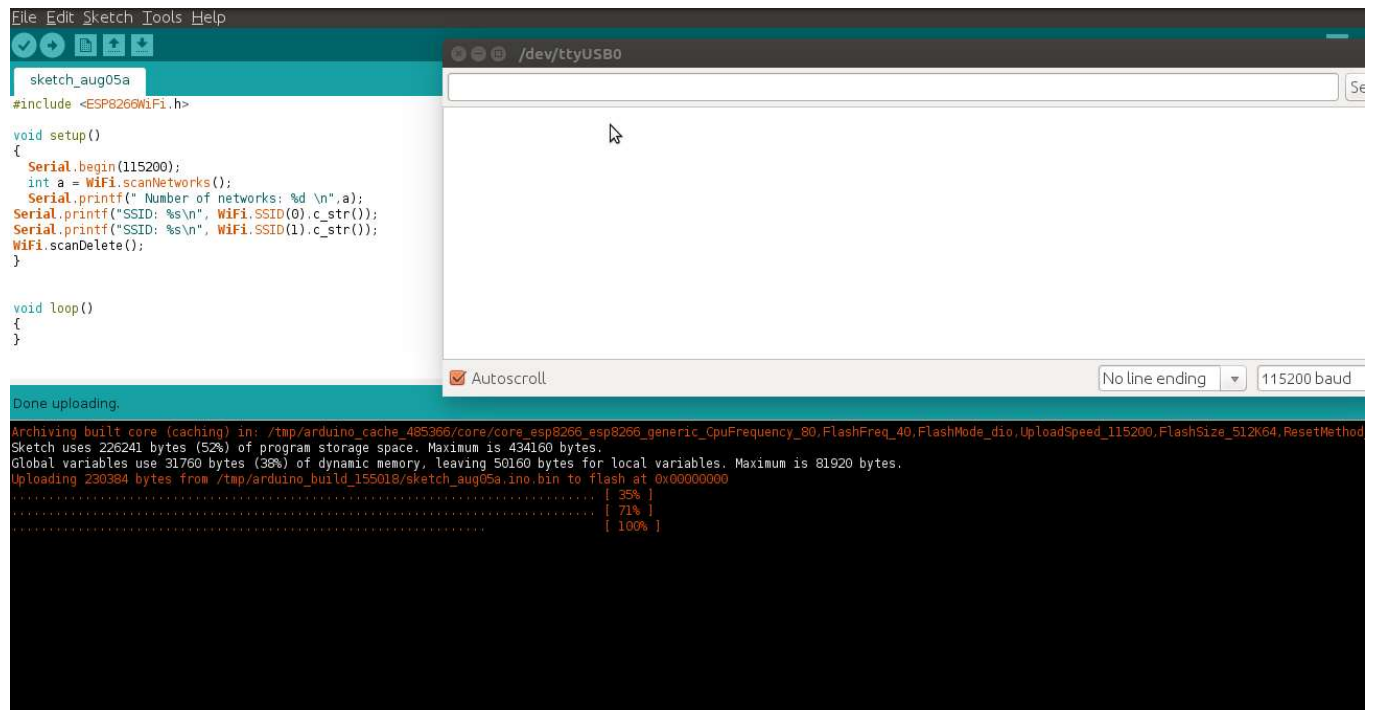
Ví dụ:

```
#include <ESP8266WiFi.h>

void setup()
{
  Serial.begin(115200);
  int a = WiFi.scanNetworks();
  Serial.printf(" Number of networks: %d \n",a);
  Serial.printf("SSID: %s\n", WiFi.SSID(0).c_str());
  Serial.printf("SSID: %s\n", WiFi.SSID(1).c_str());
  WiFi.scanDelete();
}

void loop()
{
}
```

Output:



The screenshot shows the Arduino IDE interface. The sketch editor on the left contains the following code:

```
sketch_aug05a
#include <ESP8266WiFi.h>

void setup()
{
  Serial.begin(115200);
  int a = WiFi.scanNetworks();
  Serial.printf(" Number of networks: %d \n",a);
  Serial.printf("SSID: %s\n", WiFi.SSID(0).c_str());
  Serial.printf("SSID: %s\n", WiFi.SSID(1).c_str());
  WiFi.scanDelete();
}

void loop()
{
}
```

The serial monitor on the right shows the output of the code, which is currently empty. The status bar at the bottom indicates "Done uploading." and "Archiving built core (caching) in: /tmp/arduino_cache_485366/core/core_esp8266_generic_CpuFrequency_80,FlashFreq_40,FlashMode_dio,UploadSpeed_115200,FlashSize_512K64,ResetMethod...".

encryptionType

Việc mã hóa dữ liệu/kiểu bảo mật (encryption) là một phần rất quan trọng trong việc cấu hình, lắp đặt và sử dụng Wifi. Xem thêm [Khái niệm Wifi cơ bản](#) để tìm hiểu cơ bản về các chuẩn bảo mật Wifi.

Hàm `Wifi.encryptionType` trả về kiểu bảo mật mà mạng Wifi sử dụng

```
Wifi.encryptionType(networkItem)
```

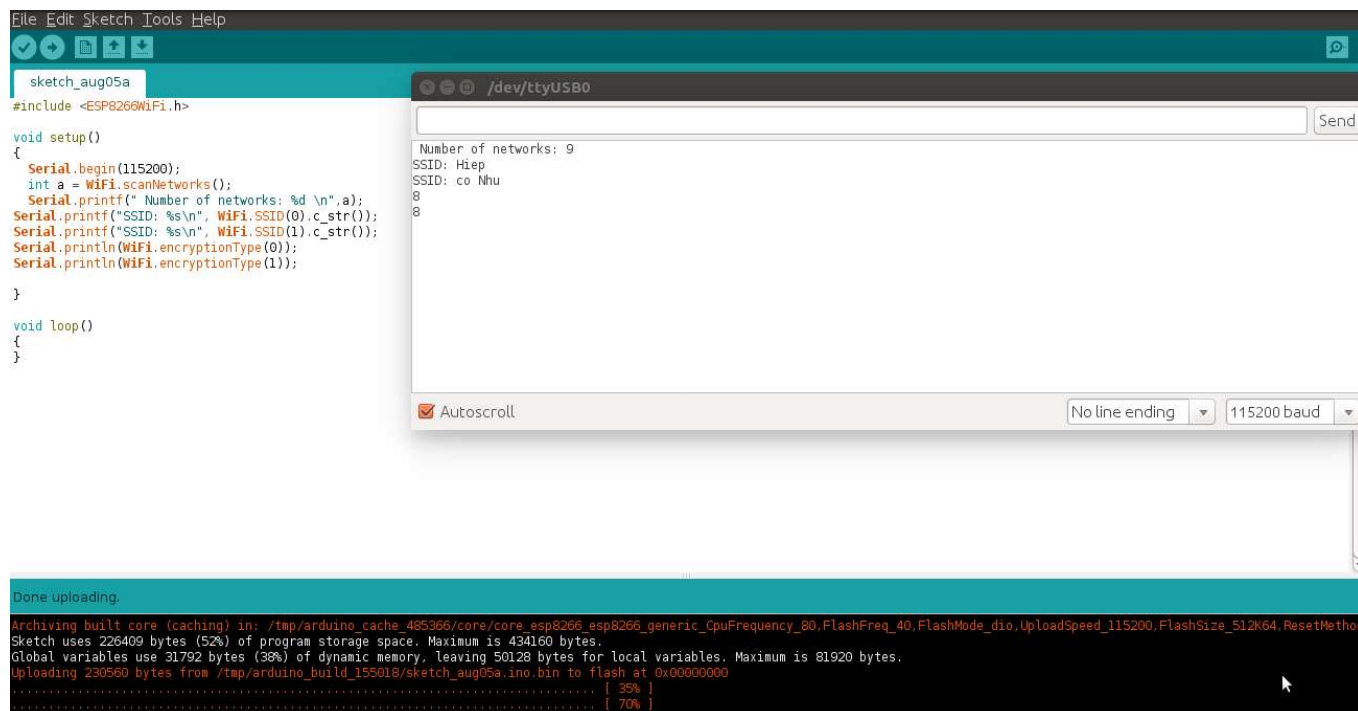
Hàm có các kết quả trả về sau đây:

- 0 : Mạng không tồn tại hoặc không scan được mạng
- 5 : Bảo mật WEP
- 2 : Bảo mật WPA / PSK
- 4 : Bảo mật WPA2 / PSK
- 7 : Không dùng bảo mật
- 8 : Bảo mật WPA / WPA2 / PSK

Code Demo:

Trả về tên 2 WIFI thứ 0 và 1 và kiểu bảo mật của 2 Wifi này

Output:



The screenshot shows the Arduino IDE interface. The sketch editor on the left contains the following code:

```
#include <ESP8266WiFi.h>

void setup()
{
  Serial.begin(115200);
  int a = WiFi.scanNetworks();
  Serial.printf(" Number of networks: %d \n",a);
  Serial.printf("SSID: %s\n", WiFi.SSID(0).c_str());
  Serial.printf("SSID: %s\n", WiFi.SSID(1).c_str());
  Serial.println(WiFi.encryptionType(0));
  Serial.println(WiFi.encryptionType(1));
}

void loop()
{
}
```

The serial monitor on the right, connected to /dev/ttyUSB0, shows the following output:

```
Number of networks: 9
SSID: Hiep
SSID: co Nhu
8
8
```

At the bottom, the status bar indicates "Done uploading." and a progress bar shows the upload progress.

Ta thấy 2 mạng WIFI số 0 và 1 có kết quả trả về là 8, tức là kiểu bảo mật WPA / WPA 2 / PSK

channel

Trả về kênh (channel) của mạng sau khi scan

```
WiFi.channel(số thứ tự mạng)
```

Kết quả trả về kiểu int 32 bit.

Ví dụ:

Trả về channel của 2 mạng thứ tự 0 và 1 sau khi Scan

```
#include <ESP8266WiFi.h>

void setup()
{
```

```

Serial.begin(115200); int a = WiFi.scanNetworks(); Serial.printf(" Number of networks: %d\n",a); Serial.printf("SSID: %sn", WiFi.SSID(0).c_str()); Serial.printf("SSID: %sn", WiFi.SSID(1).c_str()); Serial.println(WiFi.channel(0)); Serial.println(WiFi.channel(1));

}

void loop() { }

```

Output:

The screenshot shows the Arduino IDE interface. The sketch editor on the left contains the following code:

```

#include <ESP8266WiFi.h>

void setup()
{
  Serial.begin(115200);
  int a = WiFi.scanNetworks();
  Serial.printf(" Number of networks: %d\n",a);
  Serial.printf("SSID: %sn", WiFi.SSID(0).c_str());
  Serial.printf("SSID: %sn", WiFi.SSID(1).c_str());
  Serial.println(WiFi.channel(0));
  Serial.println(WiFi.channel(1));
}

void loop()
{
}

```

The serial monitor on the right, titled "/dev/ttyUSB0", shows the following output:

```

tail 8
chksum 0x2d
csum 0x2d
v09f0c112

Number of networks: 15
SSID: Hiep
SSID: co Nhu
1
1

```

Below the serial monitor, the status bar indicates "Done uploading." and shows the upload progress: "Archiving built core (caching) in: /tmp/arduino_cache_233468/core/core_esp8266_generic_CpuFrequency_80.FlashFreq_40.FlashMode_dio.UploadSpeed_115200.FlashSize_512K64.ResetMethod_Sketch uses 226481 bytes (52%) of program storage space. Maximum is 434160 bytes. Global variables use 31776 bytes (38%) of dynamic memory, leaving 50144 bytes for local variables. Maximum is 81920 bytes. Uploading 230624 bytes from /tmp/arduino_build_721097/sketch_aug05a.ino.bin to flash at 0x00000000".

isHidden

Khi thao tác với hàm `WiFi.scanNetworks()` thì vẫn có một số trường hợp một số mạng WiFi bị ẩn đi và không hiển thị được. Hàm `WiFi.isHidden()` dùng để kiểm tra xem một mạng SSID có phải là một mạng bị ẩn đi hay không. Hàm có kiểu trả về là boolean: true false hoặc 1 0 với 1 ứng với true, và 0 ứng với false.

```
WiFi.isHidden(networkItem)
```

Ví dụ: Kiểm tra xem 2 mạng WiFi 0 1 có phải là mạng bị ẩn không.


```
#include <ESP8266WiFi.h>

void setup()
{
    Serial.begin(115200);
    int a = WiFi.scanNetworks();
    Serial.printf(" Number of networks: %d \n",a);
    Serial.printf("SSID: %s\n", WiFi.SSID(0).c_str());
    Serial.printf("SSID: %s\n", WiFi.SSID(1).c_str());
    Serial.println(WiFi.isHidden(0));
    Serial.println(WiFi.isHidden(1));
}

void loop()
{
}
```

Output:

```
tail 8
chksum 0x2d
csum 0x2d
v09f0c112

Number of networks: 7
SSID: co Nhu
SSID: Hiep
0
0
```

Ta thấy kết quả trả về là 0, tức là false, tức là 2 mạng này không phải là mạng ẩn.

Với kết quả trả về từ `WiFi.scanNetworks()` là 1 số nguyên n mạng, thì các mạng ẩn phải có số thứ tự từ n+1 trở lên. Ta kiểm tra xem mạng Wifi thứ n+1 có được module ESP8266 scan được không, và nếu có thì hiển thị tên mạng.

Ví dụ:

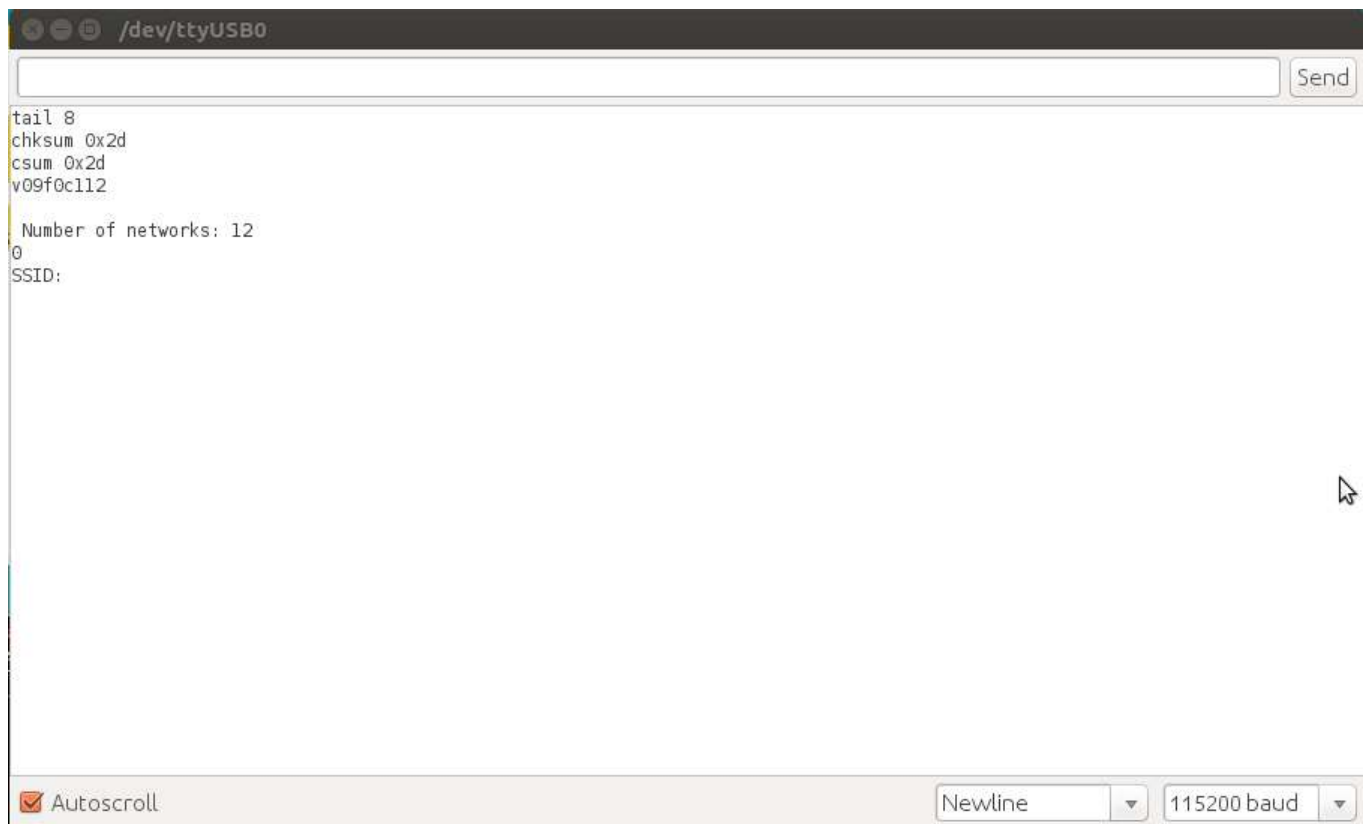
```
#include <ESP8266WiFi.h>

void setup()
{
  Serial.begin(115200);
  int a = WiFi.scanNetworks();
  Serial.printf(" Number of networks: %d \n",a);

  Serial.println(WiFi.isHidden(a+1));
  Serial.printf("SSID: %s\n", WiFi.SSID(a+1).c_str());
}

void loop()
{
}
```

Output:



```
tail 8
chksum 0x2d
csum 0x2d
v09f0c112

Number of networks: 12
0
SSID:
```

Từ kết quả ta thấy mạng WiFi ẩn không có trong trường hợp này hoặc module ESP8266 không tìm được.

RSSI

Trả về cường độ tín hiệu của một mạng Wi-Fi với thứ tự tương ứng từ kết quả của hàm

```
WiFi.scanNetworks()
```

```
WiFi.RSSI("networkItem")
```

Kết quả trả về kiểu int 32 bit.

Ví dụ:

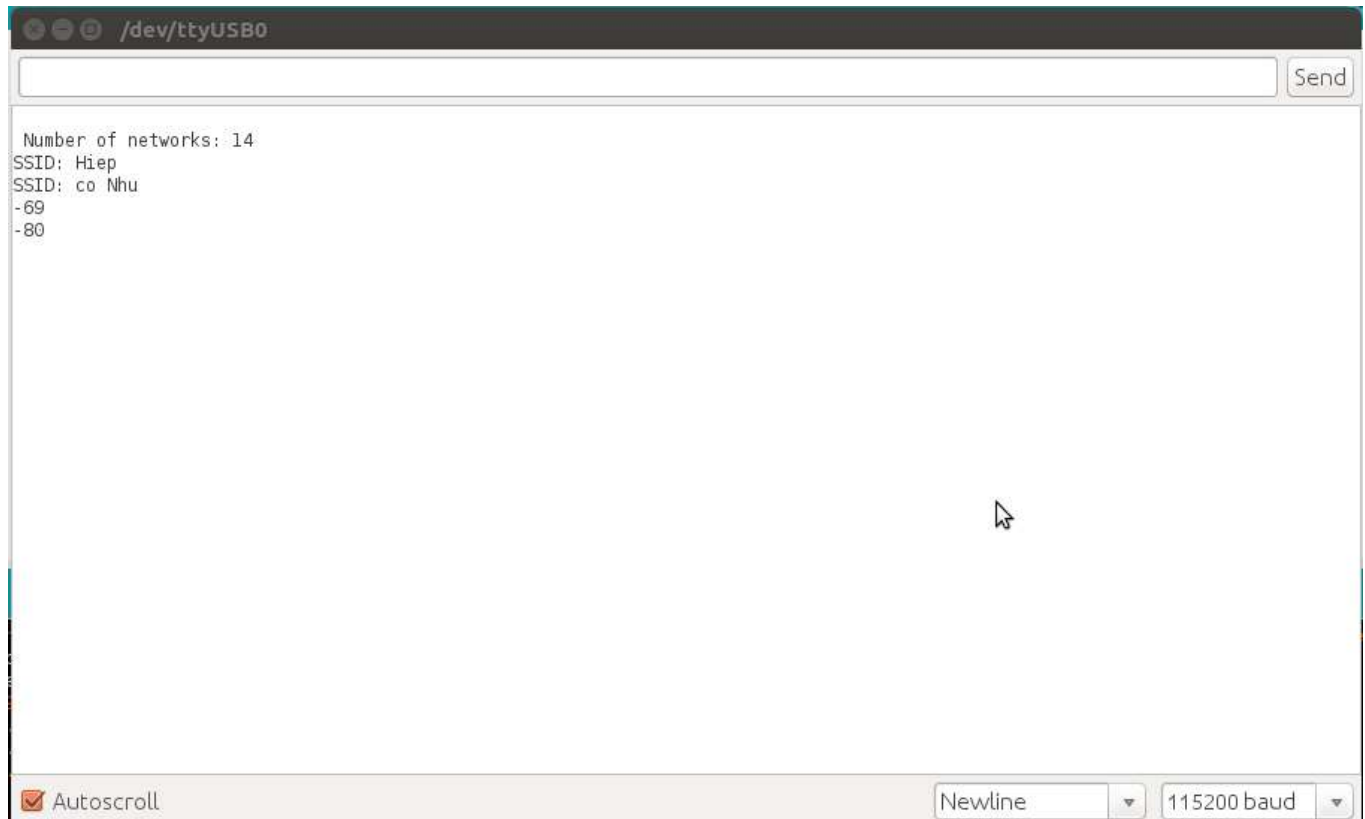
Trả về RSSI của 2 mạng Wifi 0 và 1

```
#include <ESP8266WiFi.h>

void setup()
{
    Serial.begin(115200);
    int a = WiFi.scanNetworks();
    Serial.printf(" Number of networks: %d \n",a);
    Serial.printf("SSID: %s\n", WiFi.SSID(0).c_str());
    Serial.printf("SSID: %s\n", WiFi.SSID(1).c_str());
    Serial.println(WiFi.RSSI(0));
    Serial.println(WiFi.RSSI(1));
}

void loop()
{
}
```

Output:



```
/dev/ttyUSB0

Number of networks: 14
SSID: Hiep
SSID: co Nhu
-69
-80

Autoscroll Newline 115200 baud
```

BSSID

Trả về địa chỉ MAC của một mạng WiFi với thứ tự tương ứng từ kết quả của hàm

`WiFi.scanNetworks()` (BSSID: Basic Service Set Identification là tên gọi khác của địa chỉ MAC).

```
WiFi.BSSID(networkItem)
```

Do giống với hàm `WiFi.BSSID()` trình bày ở phần WiFi Station thì kết quả trả về sẽ là ô nhớ mà địa chỉ BSSID được lưu. Do đó, để hiển thị kết quả dạng string thì ta phải dùng lệnh `WiFi.BSSIDstr()`. Để hiển thị kết quả dạng string trên Serial Monitor, ta dùng thêm lệnh `c_str()`. Do đó, code đầy đủ sẽ là:

```
WiFi.BSSIDstr(thứ tự mạng WiFi).c_str()
```

Ví dụ:

```
#include <ESP8266WiFi.h>

void setup()
{
    Serial.begin(115200);
    int a = WiFi.scanNetworks();
    Serial.printf(" Number of networks: %d \n",a);
    Serial.printf("SSID: %s\n", WiFi.SSID(0).c_str());
    Serial.printf("SSID: %s\n", WiFi.SSID(1).c_str());
    Serial.println(WiFi.BSSIDstr(0).c_str());
    Serial.println(WiFi.BSSIDstr(1).c_str());

}

void loop()
{
}
```

Output:

```
/dev/ttyUSB0
tail 8
chksum 0x2d
csum 0x2d
v09f0c112

Number of networks: 10
SSID: co Nhu
SSID: Hiep
90:17:AC:19:53:94
4C:F2:BF:1F:0D:4C
```

! Ghi chú

Các hàm `WiFi.SSID`, `WiFi.RSSI`, `WiFi.BSSID`, `WiFi.channel` trình bày ở phần Scan class này và ở phần WiFi Station đều là một, nhưng để thao tác ở Scan Class (muốn hiển thị thông tin của mạng Wifi) thì khác với các hàm ở WiFi Station ở chỗ là phải thêm vào đối số : thứ tự mạng Wifi.

getNetworkInfo

Đây là một hàm hữu hiệu nhất trong khi làm việc với thư viện ESP8266WiFi. Hàm trả về tất cả các thông tin cần thiết của một mạng WiFi (trừ password), như đã trình bày ở các hàm ở trên: SSID, Kiểu bảo mật, RSSI, BSSID, channel và nếu WiFi ẩn khi scan. Kiểu trả về của hàm là kiểu boolean (true hoặc false). Kiểu true trả về khi thu được thông tin của mạng WiFi, và kiểu false trả về khi không thu được kết quả. Muốn hiển thị các thông tin như SSID, RSSI,... thì ta dùng hàm `Serial.printf`.

```
WiFi.getNetworkInfo("networkItem", &ssid, &encryptionType, &RSSI, *&BSSID, &channel, &isHidden)
```

Ví dụ:

Trả về thông tin về các mạng WiFi dùng hàm `WiFi.getNetworkInfo()`

```
#include <ESP8266WiFi.h>

void setup()
{
  Serial.begin(115200);
  int n = WiFi.scanNetworks();

  String ssid;
  uint8_t encryptionType;
  int32_t RSSI;
  uint8_t* BSSID;
  int32_t channel;
  bool isHidden;

  for (int i = 0; i < n; i++)
  {
    WiFi.getNetworkInfo(i, ssid, encryptionType, RSSI, BSSID, channel, isHidden);
    Serial.printf("%d: %s, Ch:%d (%ddBm) %s %s\n", i + 1, ssid.c_str(), channel, RSSI,
    encryptionType == ENC_TYPE_NONE ? "open" : "", isHidden ? "hidden" : "");
  }
}

void loop(){}

```

Output:

```
tail 8
chksum 0x2d
csum 0x2d
v09f0c112
1: Hiep, Ch:1 (-72dBm)
2: co Nhu, Ch:1 (-82dBm)
3: DIRECT-vb-BRAVIA, Ch:1 (-83dBm)
4: PhiCong, Ch:2 (-86dBm)
5: SV Maria, Ch:3 (-89dBm)
6: lêu_lêu, Ch:3 (-91dBm)
7: Hong_Nhung, Ch:6 (-87dBm)
8: Hong_Nhung, Ch:6 (-80dBm)
9: Tenda_0B31C0, Ch:9 (-88dBm) open
10: MeoCon, Ch:11 (-77dBm)
11: KimHuynh2, Ch:11 (-70dBm)
12: Tenda, Ch:11 (-84dBm)
13: Meo Con, Ch:11 (-91dBm)
```

Autoscroll Newline 115200 baud