程式：ScanNetworks(掃描AP)

**開啟程式**ScanNetworks

**程式位址：**https://github.com/brucetsao/BruceCourses/blob/master/105ANQU\_IOT/Code/ScanNetworks/ScanNetworks.ino

|  |
| --- |
| /\*  This example prints the Wifi shield's MAC address, and  scans for available Wifi networks using the Wifi shield.  Every ten seconds, it scans again. It doesn't actually  connect to any network, so no encryption scheme is specified.  Circuit:  \* WiFi shield attached  created 13 July 2010  by dlf (Metodo2 srl)  modified 21 Junn 2012  by Tom Igoe and Jaymes Dec  \*/  #include <WiFi.h>  void setup() {  //Initialize serial and wait for port to open:  Serial.begin(9600);  while (!Serial) {  ; // wait for serial port to connect. Needed for native USB port only  }  // check for the presence of the shield:  if (WiFi.status() == WL\_NO\_SHIELD) {  Serial.println("WiFi shield not present");  // don't continue:  while (true);  }  String fv = WiFi.firmwareVersion();  if (fv != "1.1.0") {  Serial.println("Please upgrade the firmware");  }  // Print WiFi MAC address:  printMacAddress();  }  void loop() {  // scan for existing networks:  Serial.println("Scanning available networks...");  listNetworks();  delay(10000);  }  void printMacAddress() {  // the MAC address of your Wifi shield  byte mac[6];  // print your MAC address:  WiFi.macAddress(mac);  Serial.print("MAC: ");  Serial.print(mac[5], HEX);  Serial.print(":");  Serial.print(mac[4], HEX);  Serial.print(":");  Serial.print(mac[3], HEX);  Serial.print(":");  Serial.print(mac[2], HEX);  Serial.print(":");  Serial.print(mac[1], HEX);  Serial.print(":");  Serial.println(mac[0], HEX);  }  void listNetworks() {  // scan for nearby networks:  Serial.println("\*\* Scan Networks \*\*");  int numSsid = WiFi.scanNetworks();  if (numSsid == -1) {  Serial.println("Couldn't get a wifi connection");  while (true);  }  // print the list of networks seen:  Serial.print("number of available networks:");  Serial.println(numSsid);  // print the network number and name for each network found:  for (int thisNet = 0; thisNet < numSsid; thisNet++) {  Serial.print(thisNet);  Serial.print(") ");  Serial.print(WiFi.SSID(thisNet));  Serial.print("\tSignal: ");  Serial.print(WiFi.RSSI(thisNet));  Serial.print(" dBm");  Serial.print("\tEncryptionRaw: ");  printEncryptionTypeEx(WiFi.encryptionTypeEx(thisNet));  Serial.print("\tEncryption: ");  printEncryptionType(WiFi.encryptionType(thisNet));  }  }  void printEncryptionTypeEx(uint32\_t thisType) {  /\* Arduino wifi api use encryption type to mapping to security type.  \* This function demonstrate how to get more richful information of security type.  \*/  switch (thisType) {  case SECURITY\_OPEN:  Serial.print("Open");  break;  case SECURITY\_WEP\_PSK:  Serial.print("WEP");  break;  case SECURITY\_WPA\_TKIP\_PSK:  Serial.print("WPA TKIP");  break;  case SECURITY\_WPA\_AES\_PSK:  Serial.print("WPA AES");  break;  case SECURITY\_WPA2\_AES\_PSK:  Serial.print("WPA2 AES");  break;  case SECURITY\_WPA2\_TKIP\_PSK:  Serial.print("WPA2 TKIP");  break;  case SECURITY\_WPA2\_MIXED\_PSK:  Serial.print("WPA2 Mixed");  break;  case SECURITY\_WPA\_WPA2\_MIXED:  Serial.print("WPA/WPA2 AES");  break;  }  }  void printEncryptionType(int thisType) {  // read the encryption type and print out the name:  switch (thisType) {  case ENC\_TYPE\_WEP:  Serial.println("WEP");  break;  case ENC\_TYPE\_TKIP:  Serial.println("WPA");  break;  case ENC\_TYPE\_CCMP:  Serial.println("WPA2");  break;  case ENC\_TYPE\_NONE:  Serial.println("None");  break;  case ENC\_TYPE\_AUTO:  Serial.println("Auto");  break;  }  } |

ScanNetworks**程式重點解說**

* #include <WiFi.h> 使用網路必要函數
* WiFi.status() == WL\_NO\_SHIELD 檢查有網路供能否
* WiFi.firmwareVersion();檢查網路韌體版本
* listNetworks() 列出可連接到的AP(自訂)
* WiFi.scanNetworks(); 取得可連接到的AP並存入(-1為沒有AP可連接)
* WiFi.SSID(n) 可連接到的AP(n)的名字
* WiFi.RSSI(n) 可連接到的AP(n)的RSSI
* WiFi.encryptionTypeEx(n)可連接到的AP(n)的加密方式I