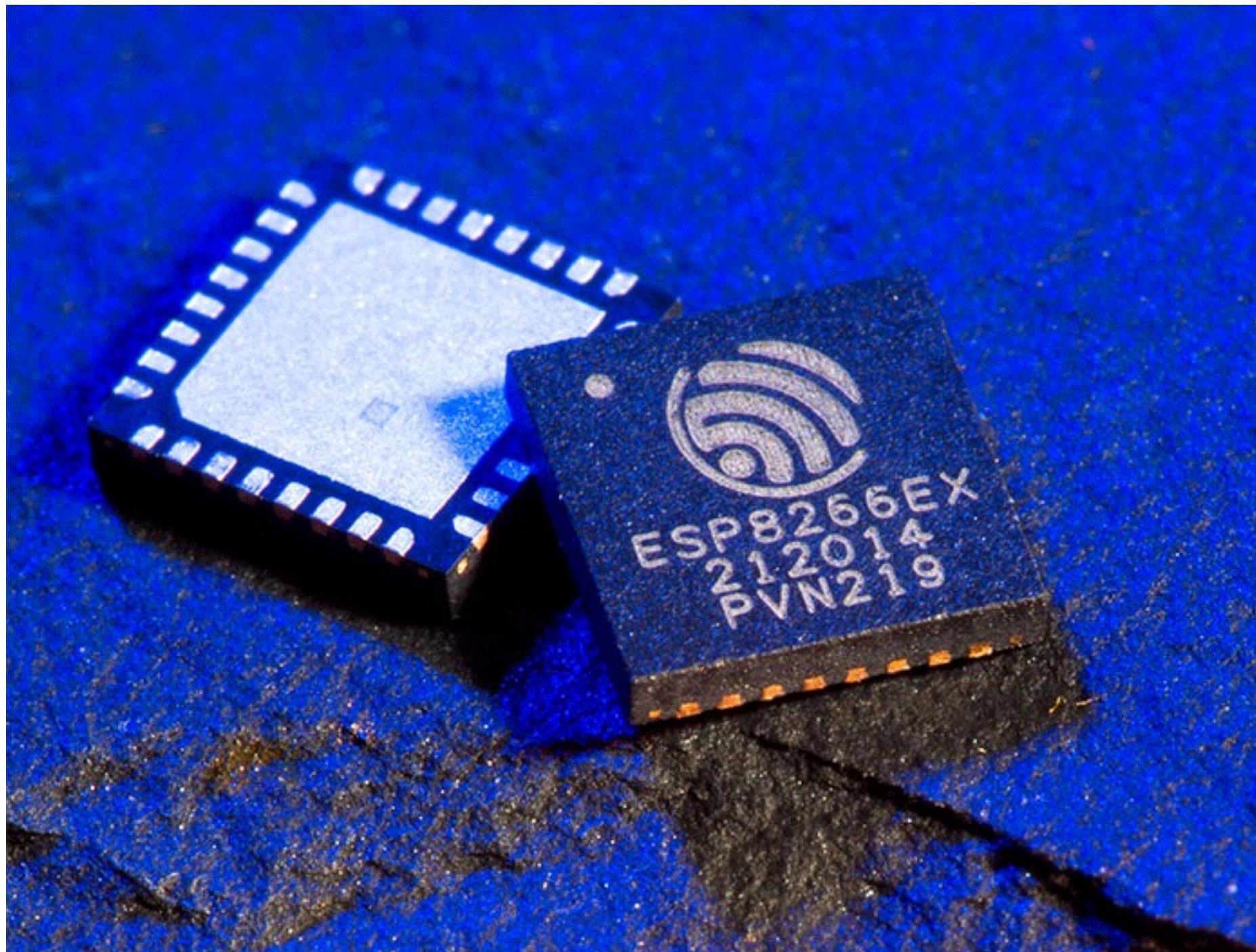
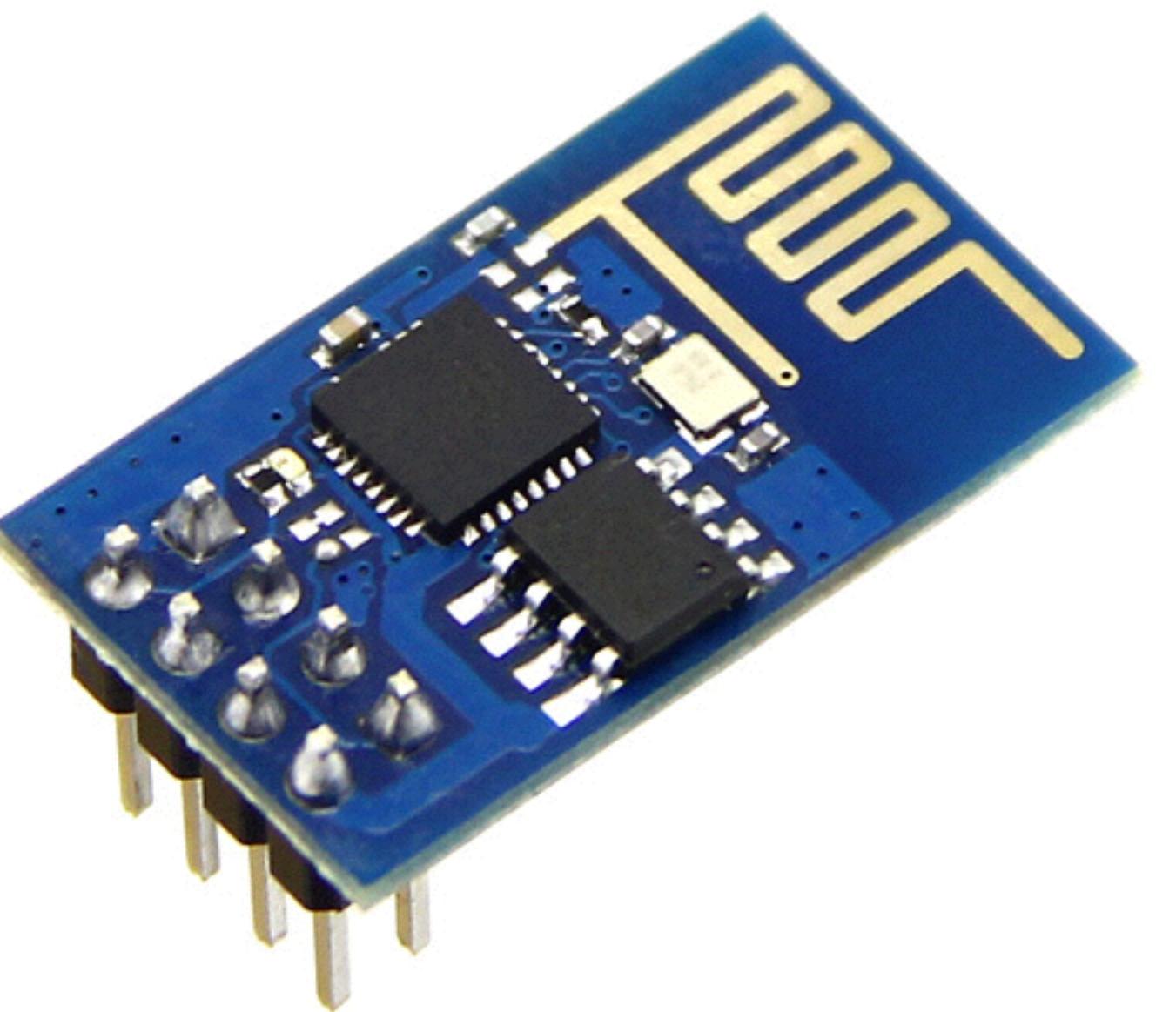
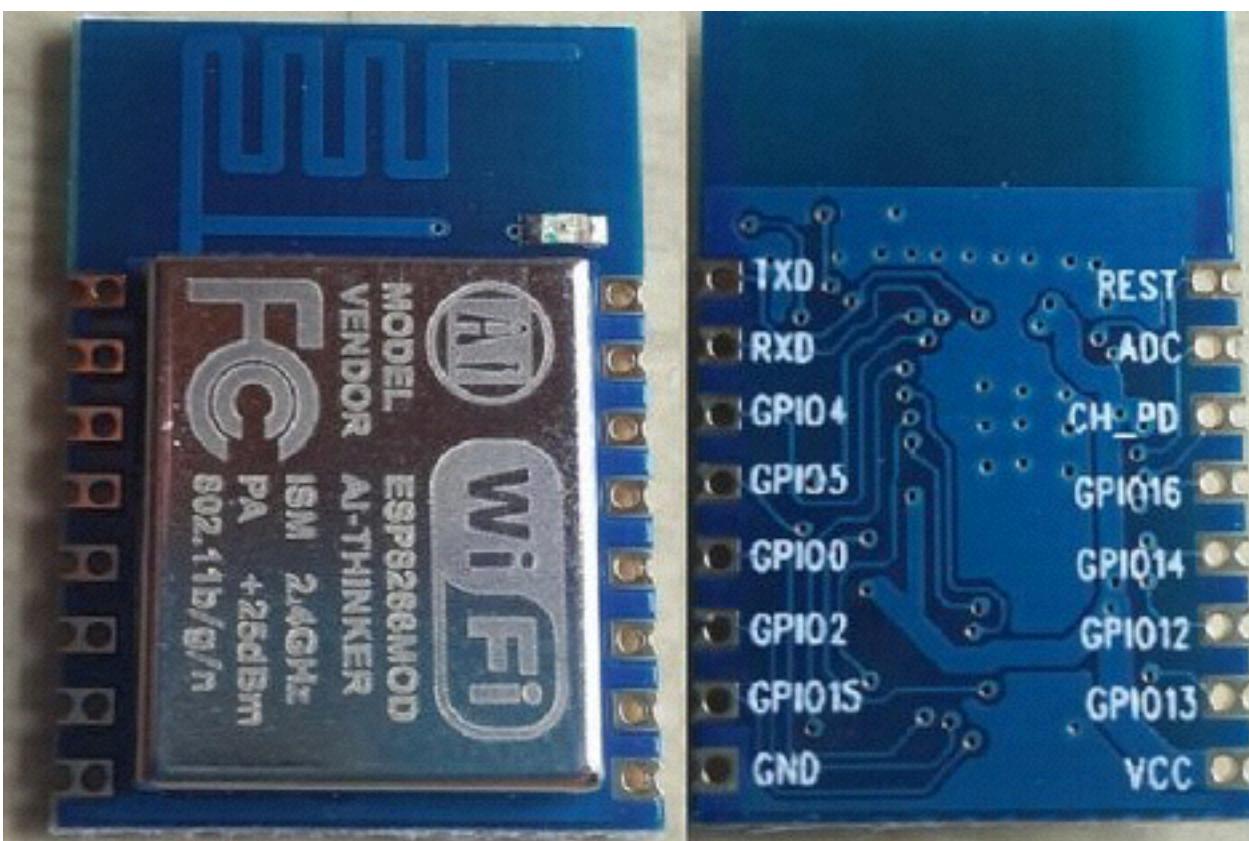


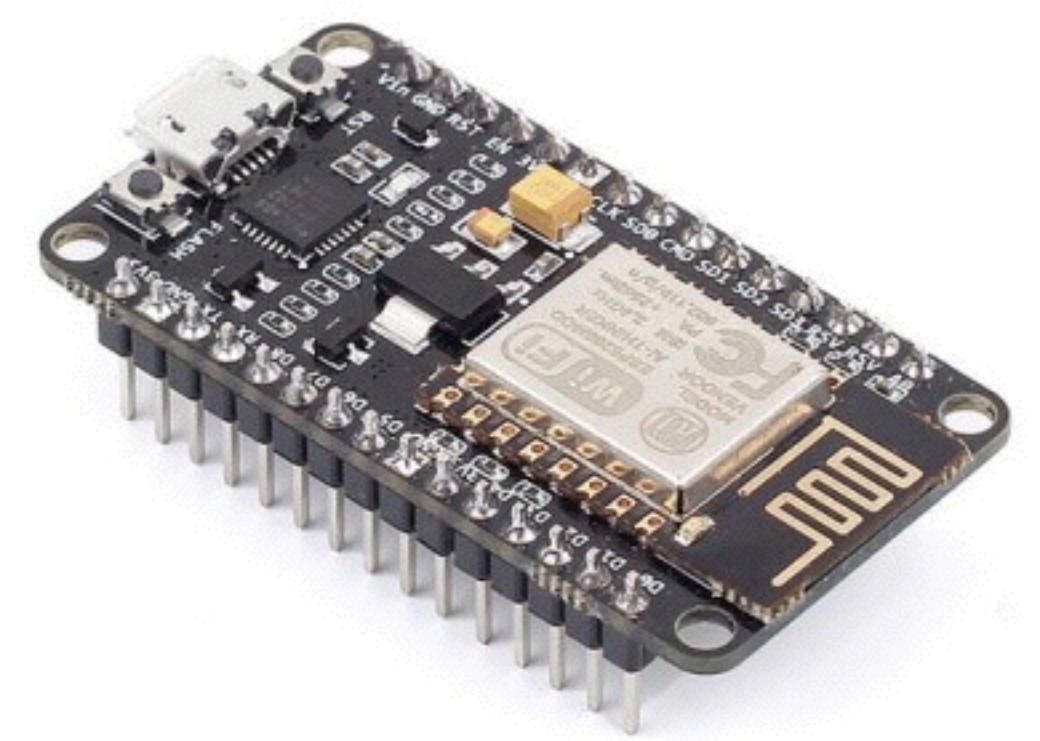
# ESP8266 & Arduino IDE

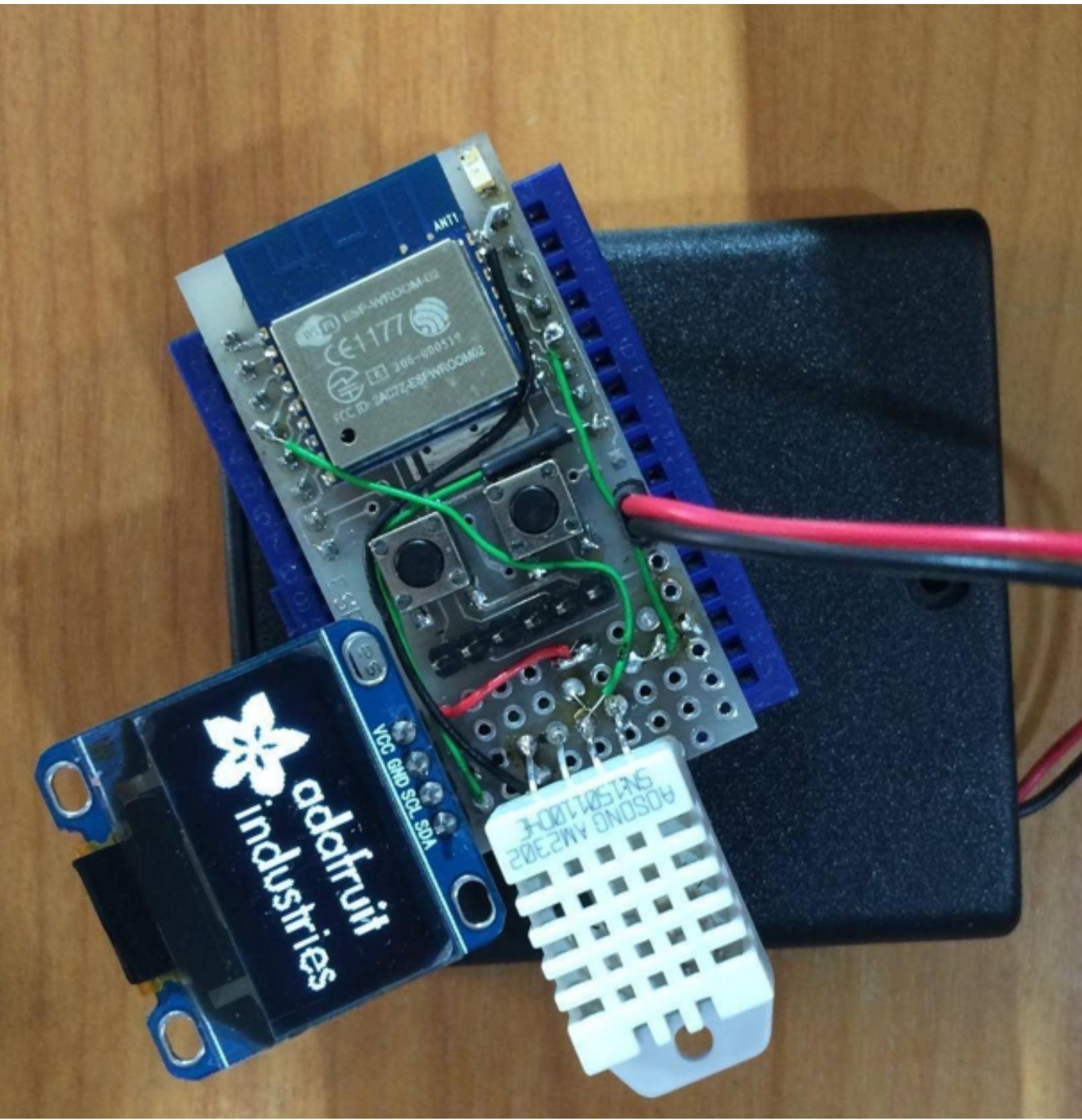
The ESPresso Lite



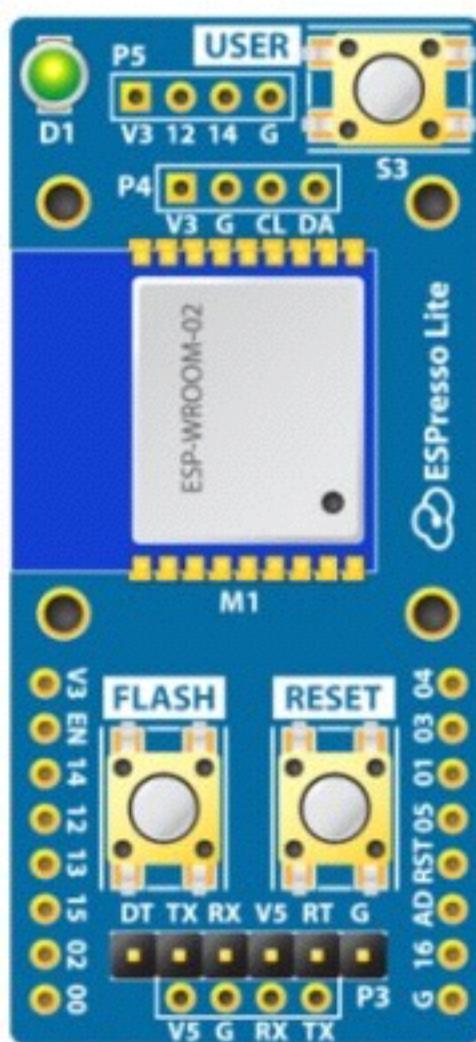




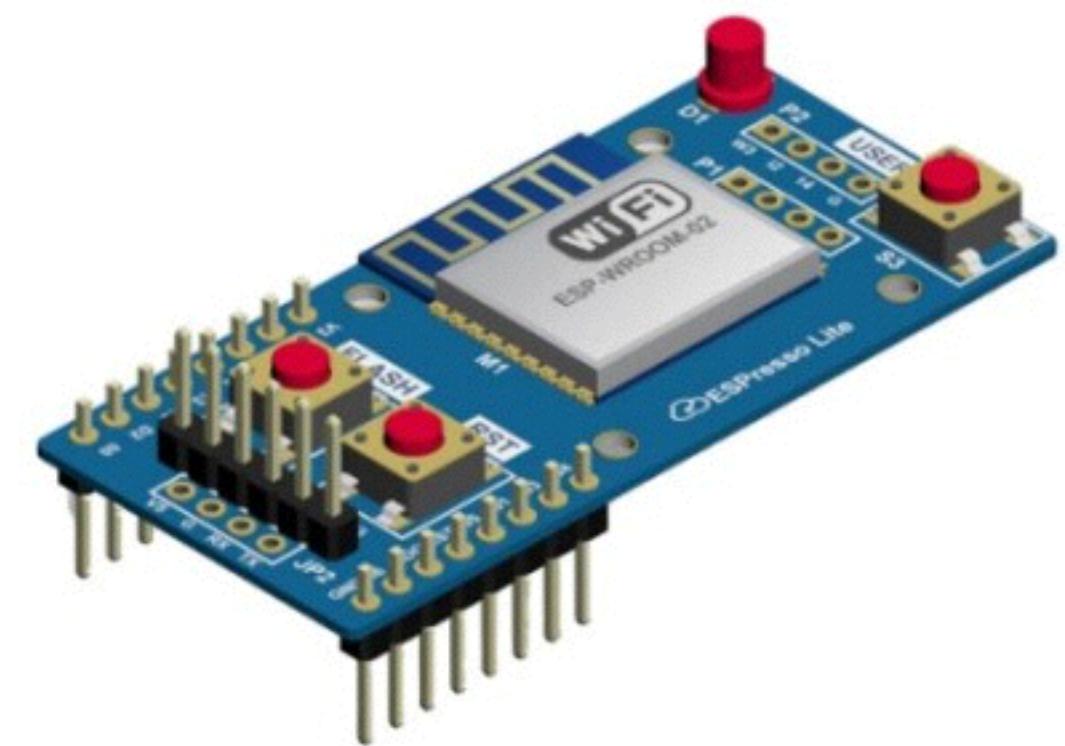


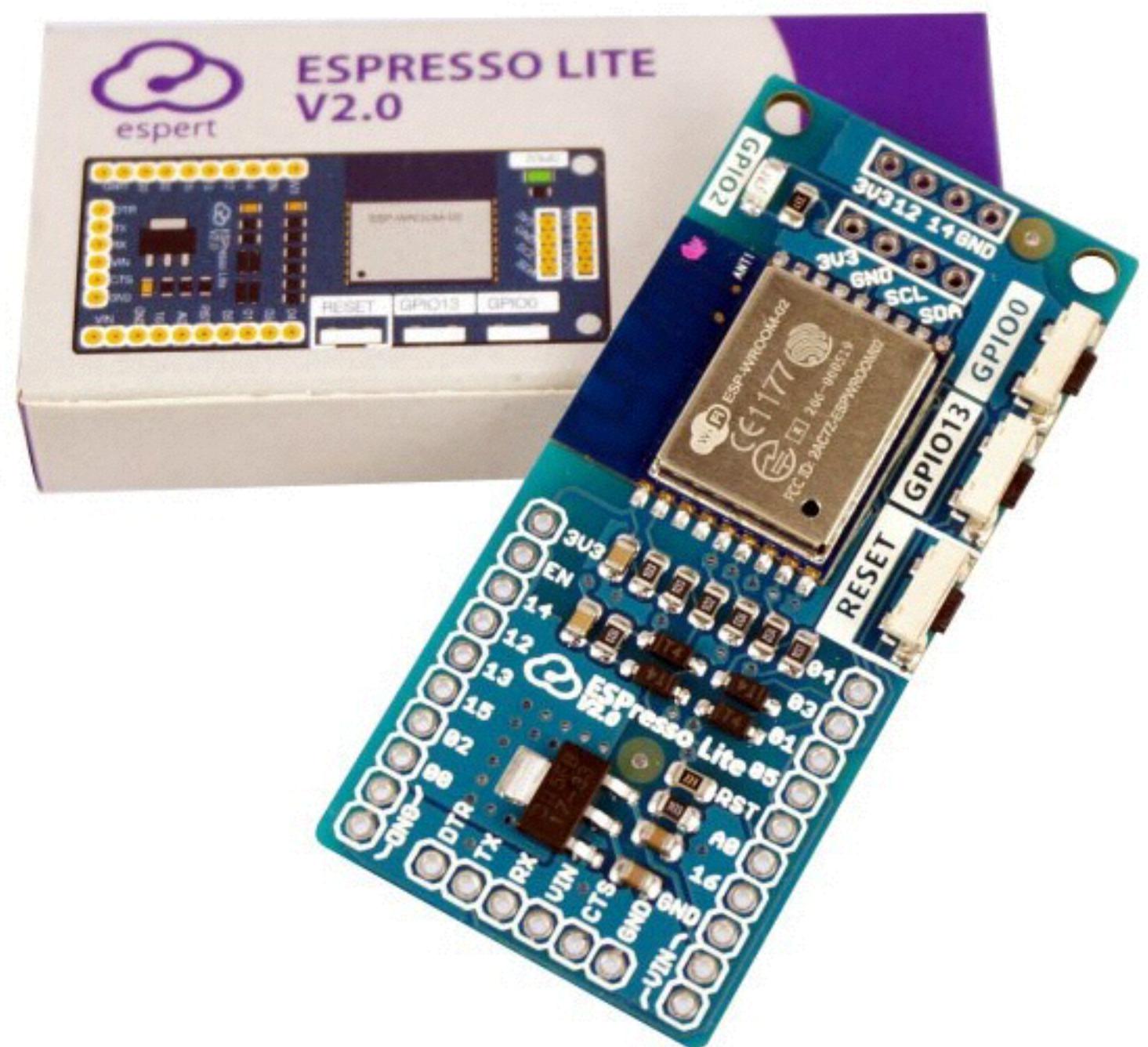


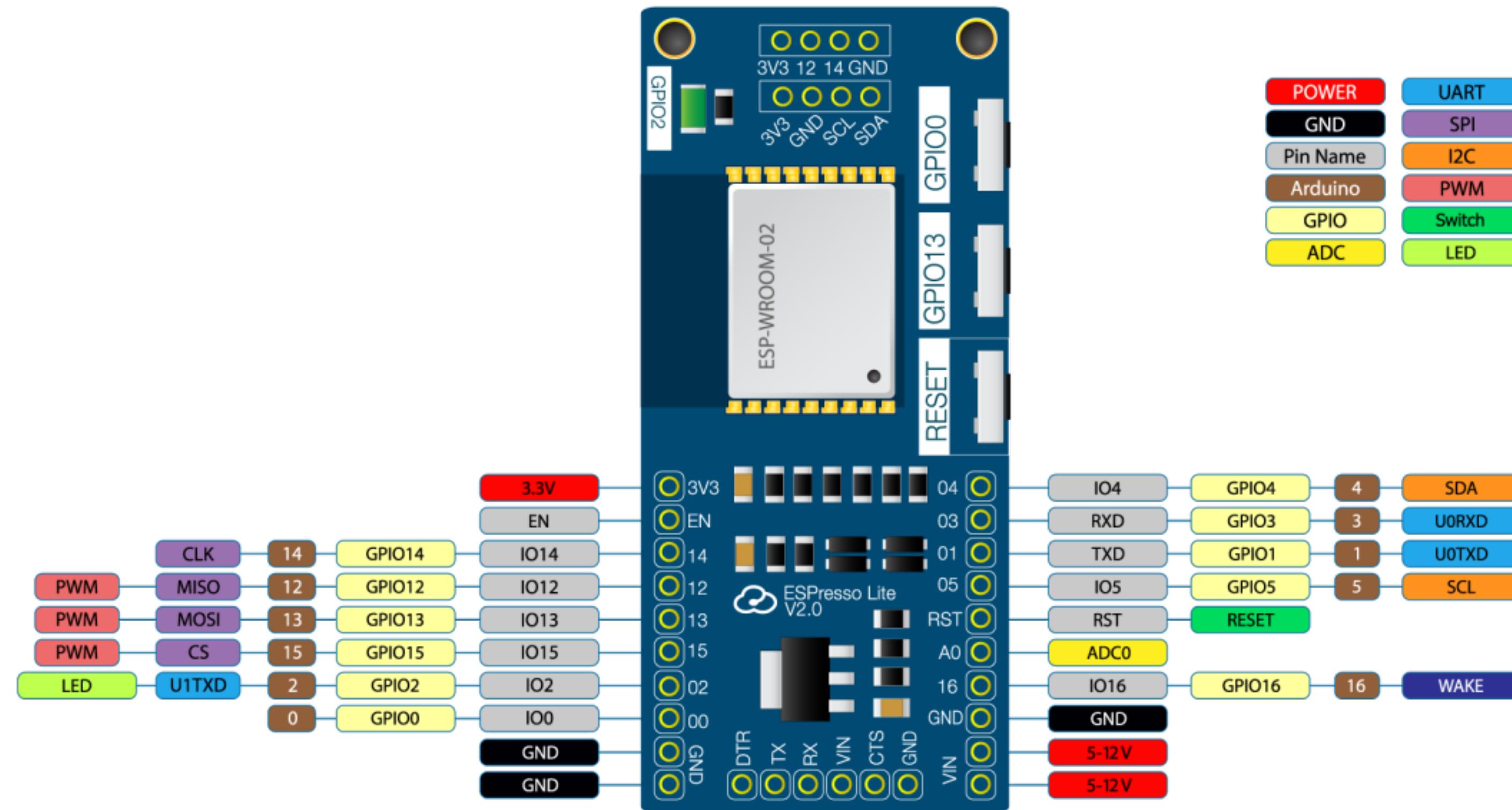
# ESPresso Lite



Latest Arduino-compatible,  
WiFi-enabled (ESP8266)  
development board







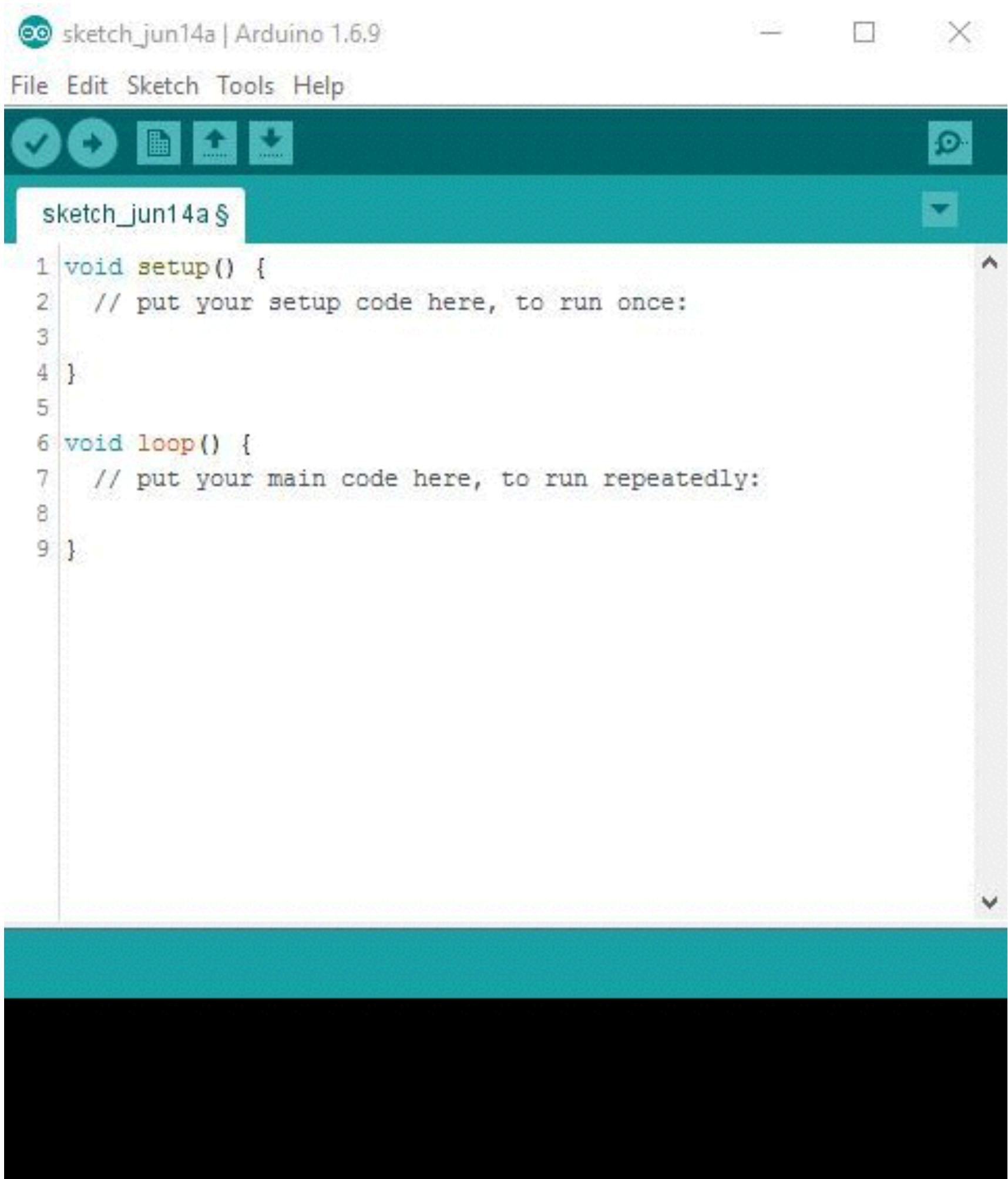
# ESPresso Lite V2.0

- Espressif's certified ESP-WROOM-02 Wi-Fi module  
(which houses the popular 32-bit 80 Mhz ESP8266 SoC with 64kb RAM & 4Mb flash),
- Two user-programmable buttons (connected to pin 0 & 13) and a reset button,
- Green SMD LED indicator,
- On board 3.3V Voltage Regulator with maximum current of 800mA continuous, 1A peak,
- Input voltage Vin: 5 - 12 VDC; operating voltage at 3.3VDC,
- Supports the Arduino IDE with own board manager and libraries,
- Auto program loading from Arduino IDE; no manual resetting required,
- Custom-arranged I2C pads for I2C-compliant sensors or OLED LC display,
- Breakout pins are breadboard-friendly .

## Download the Arduino Software

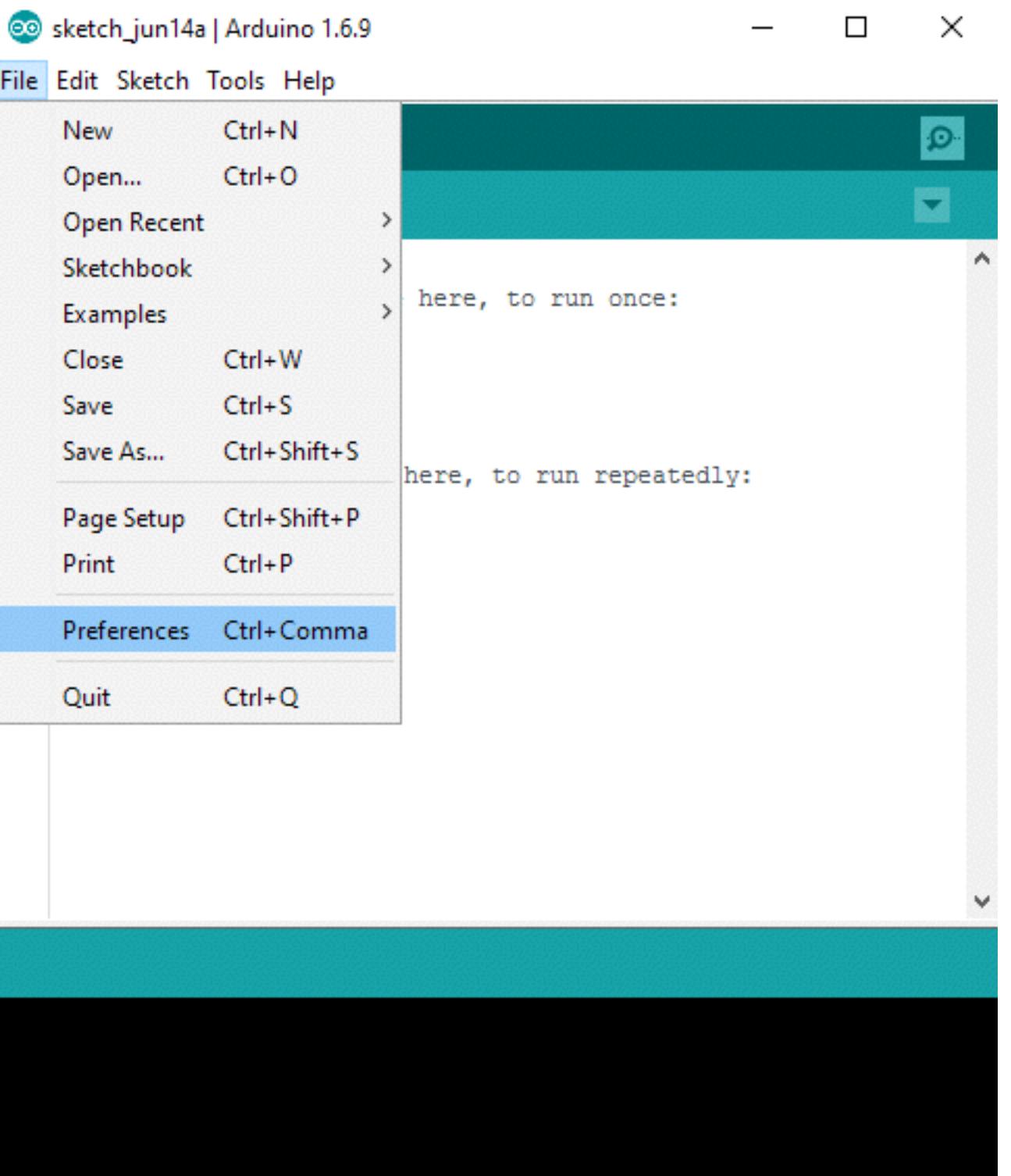


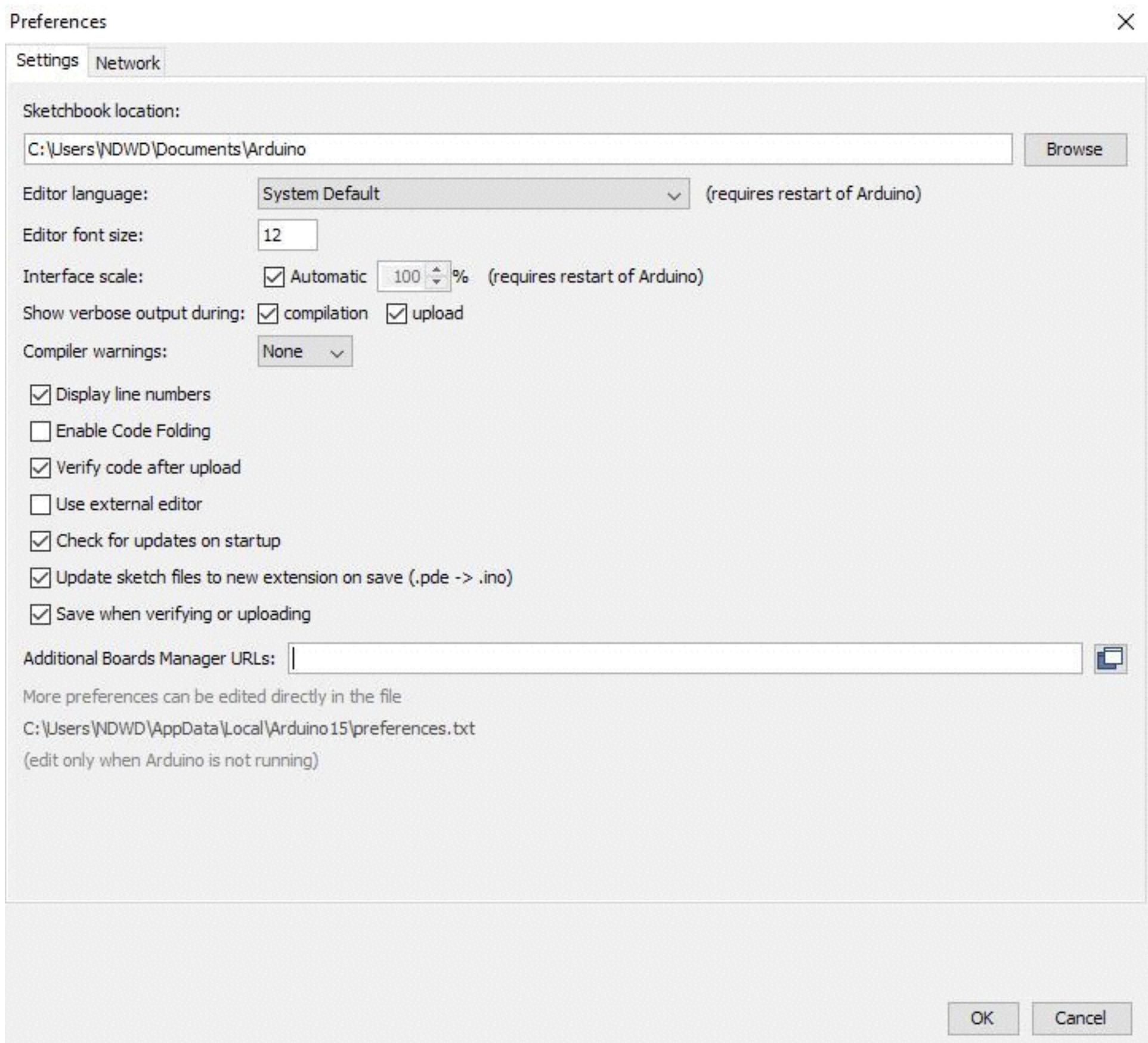


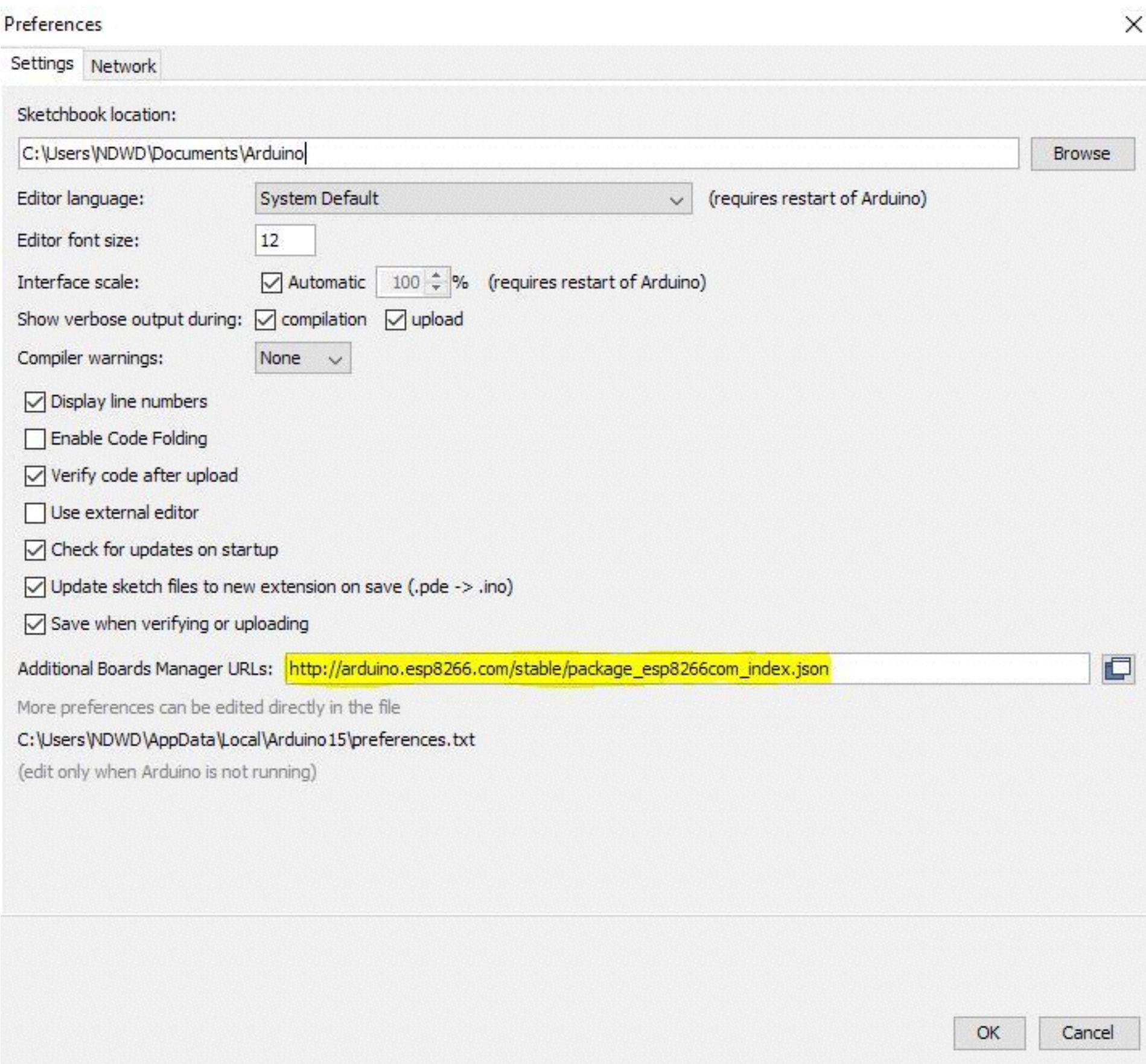


The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_jun14a | Arduino 1.6.9". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for file operations like Open, Save, and Print. The main window displays the code for "sketch\_jun14a":

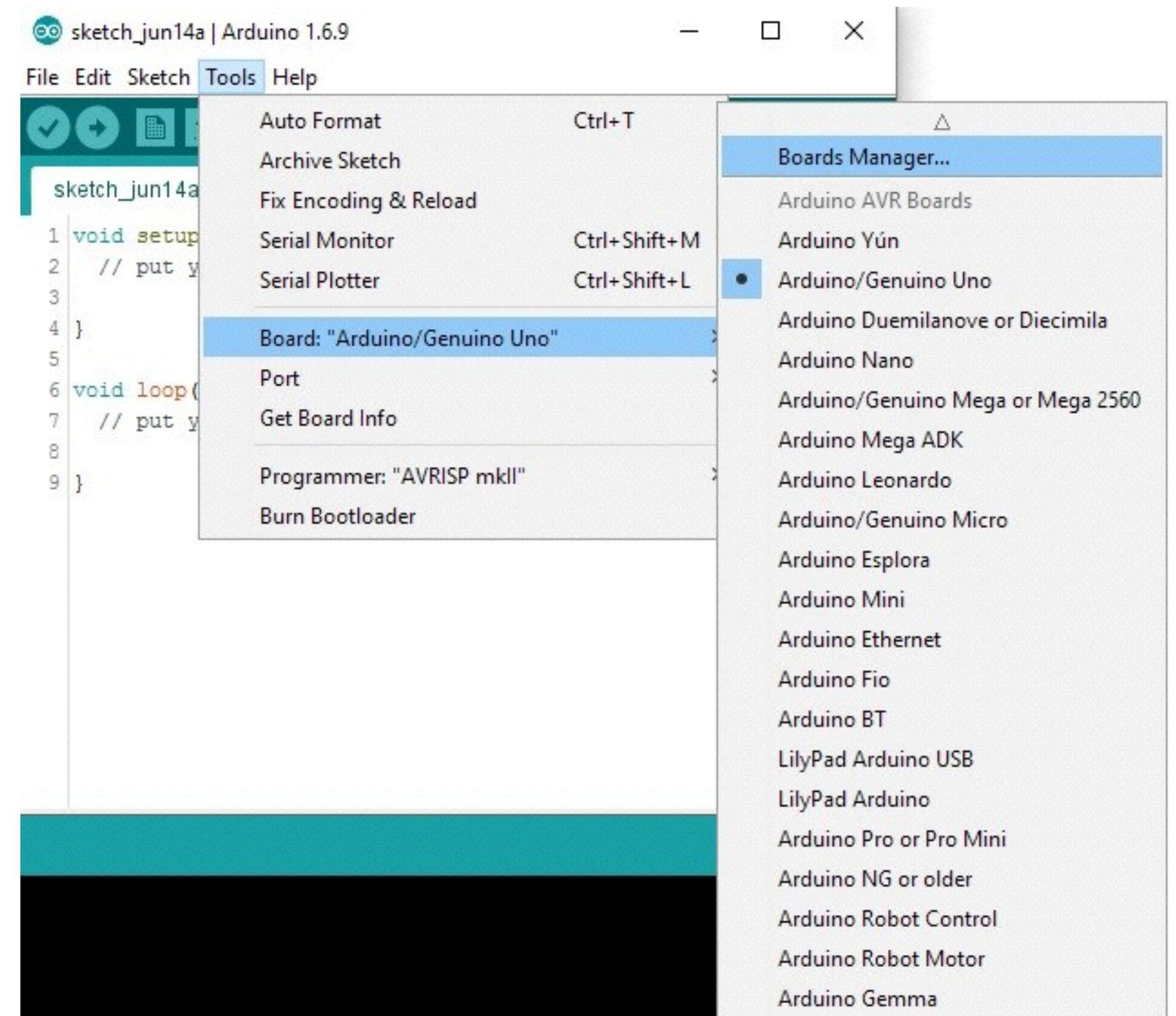
```
1 void setup() {
2 // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8 }
```

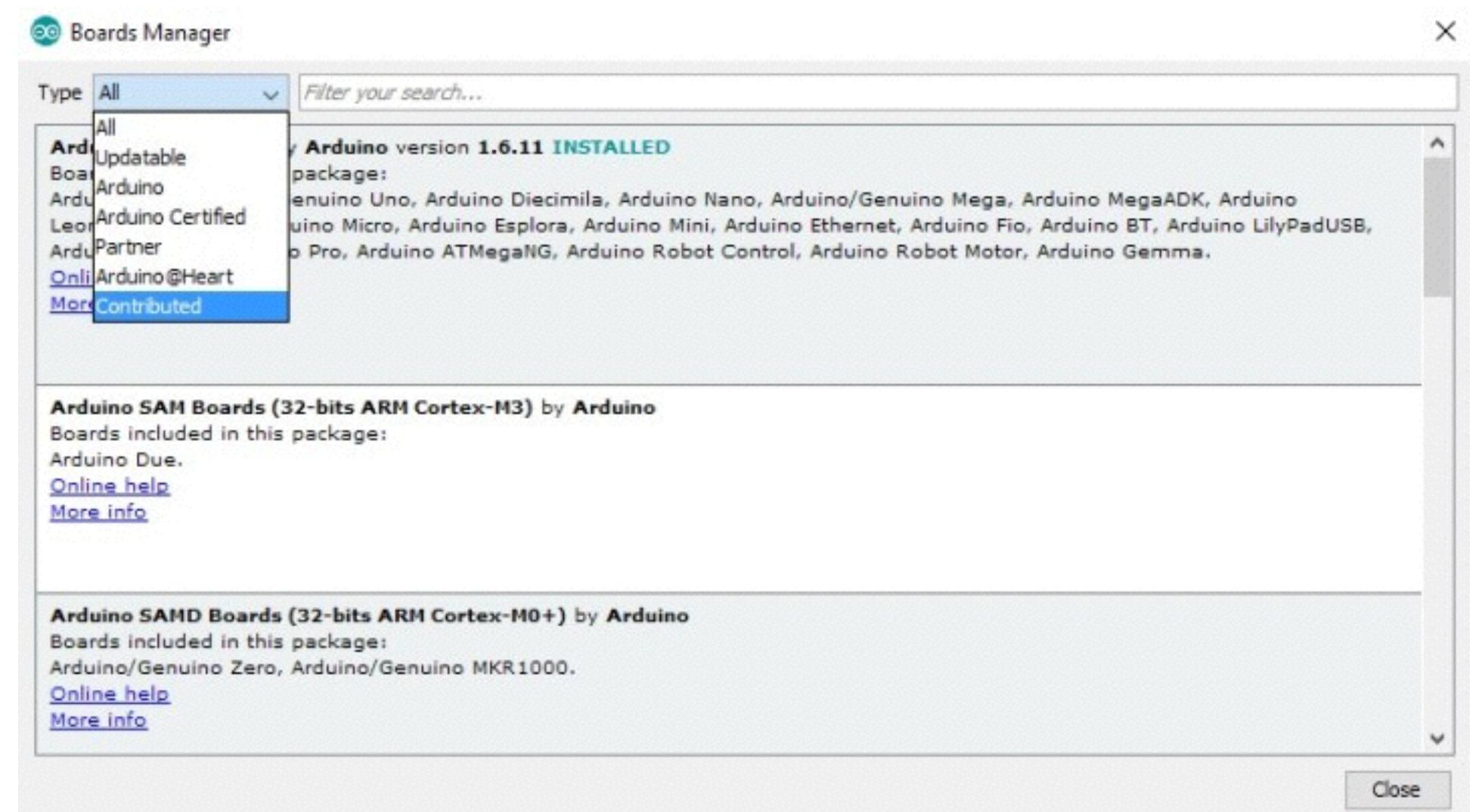


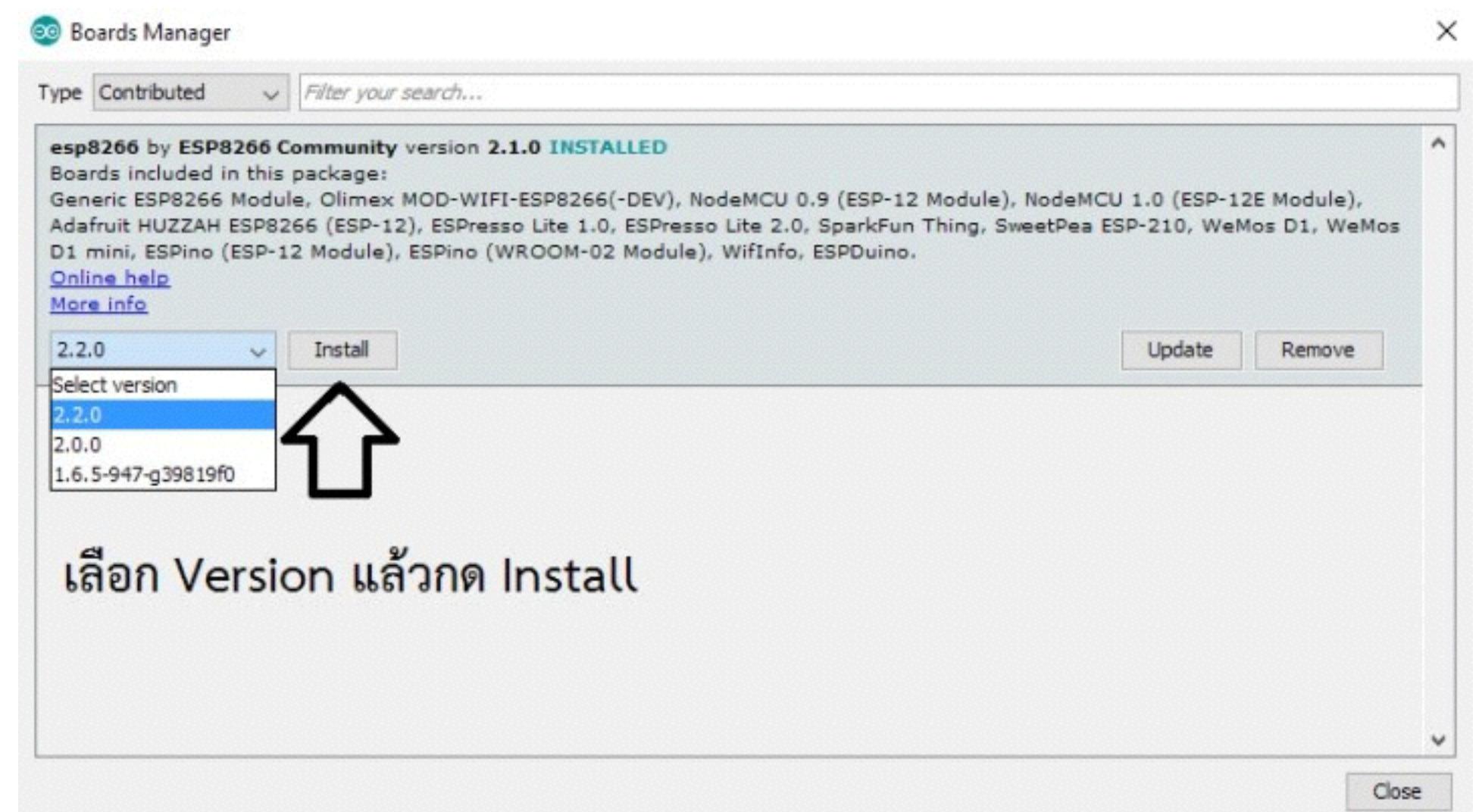




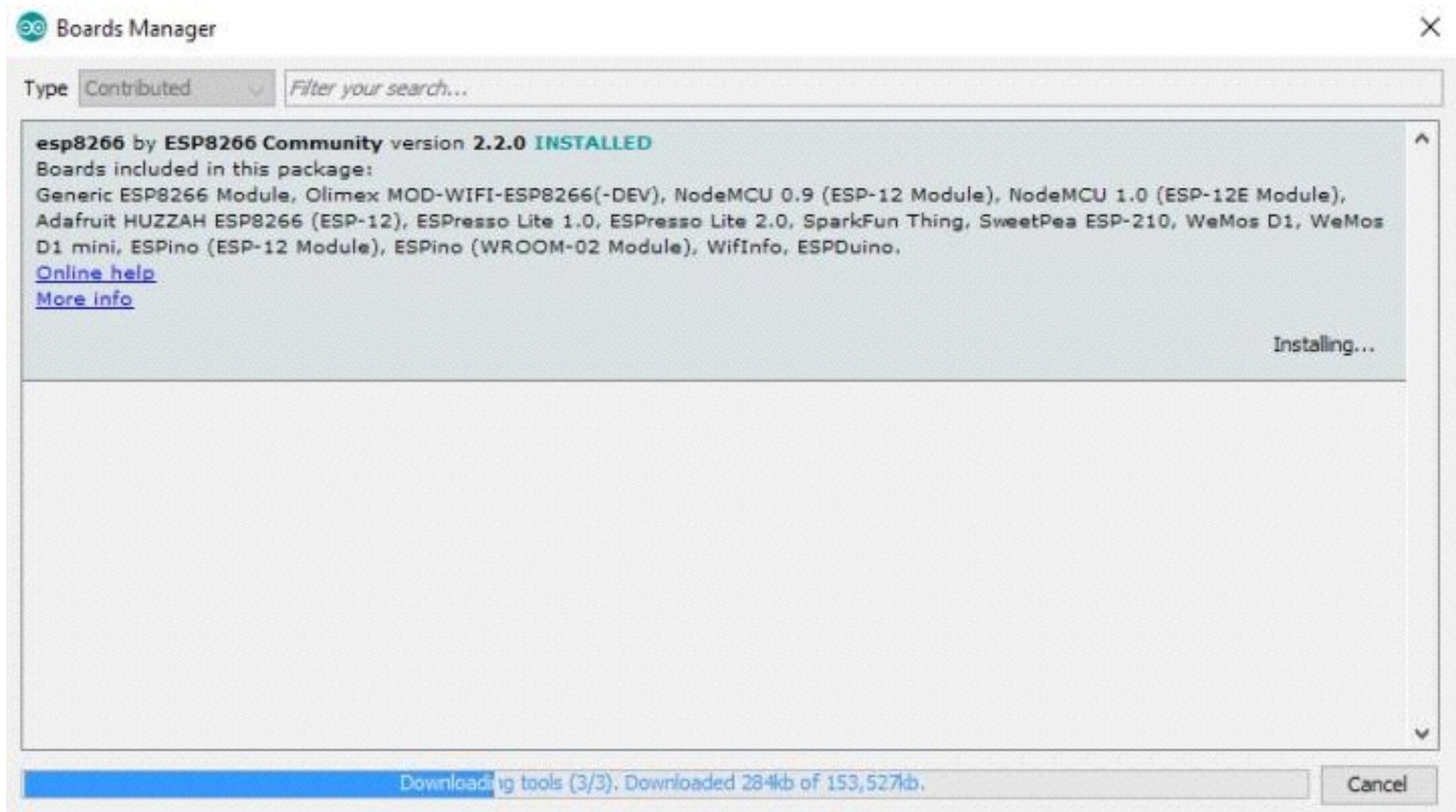
<https://github.com/esp8266/Arduino>

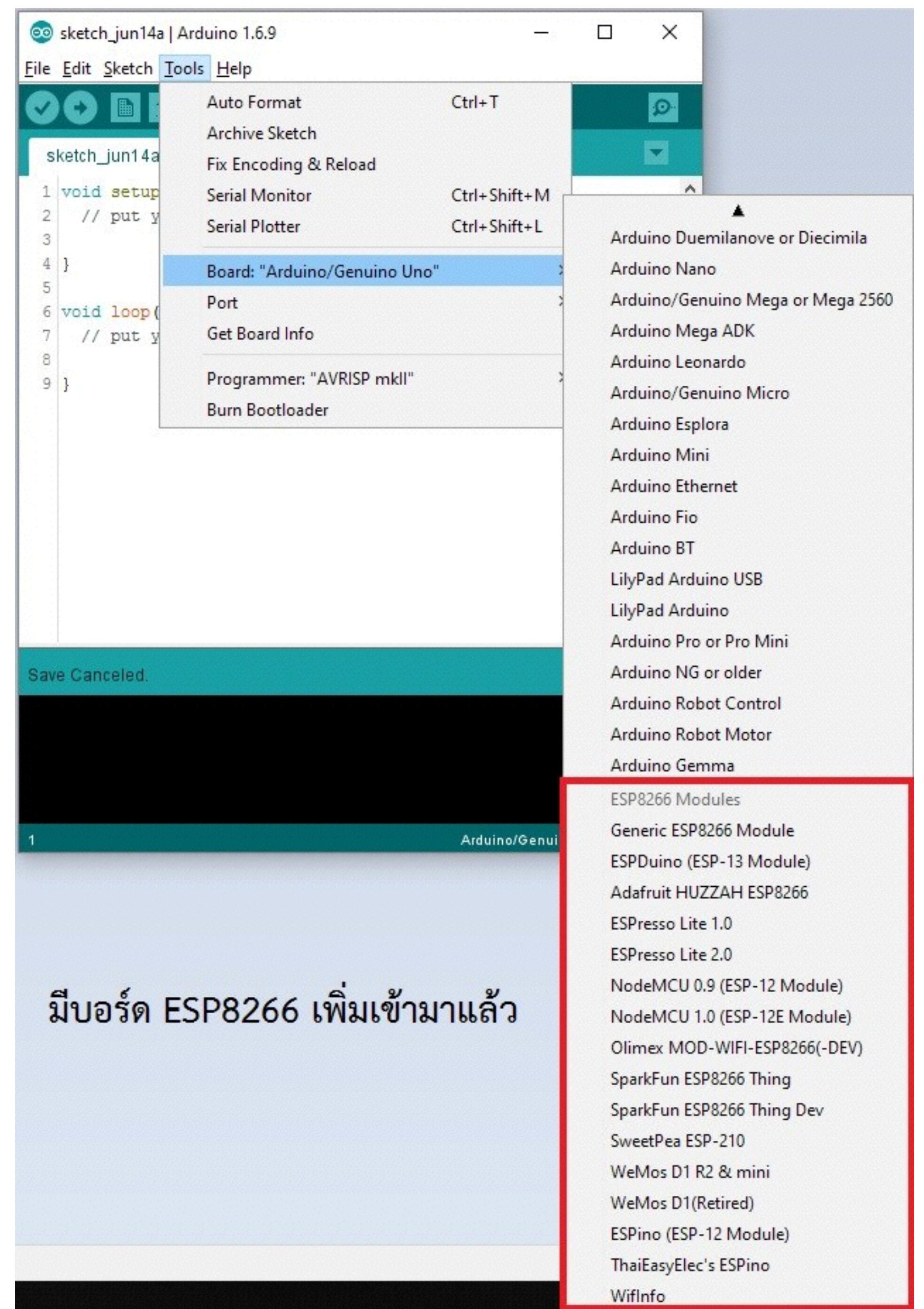




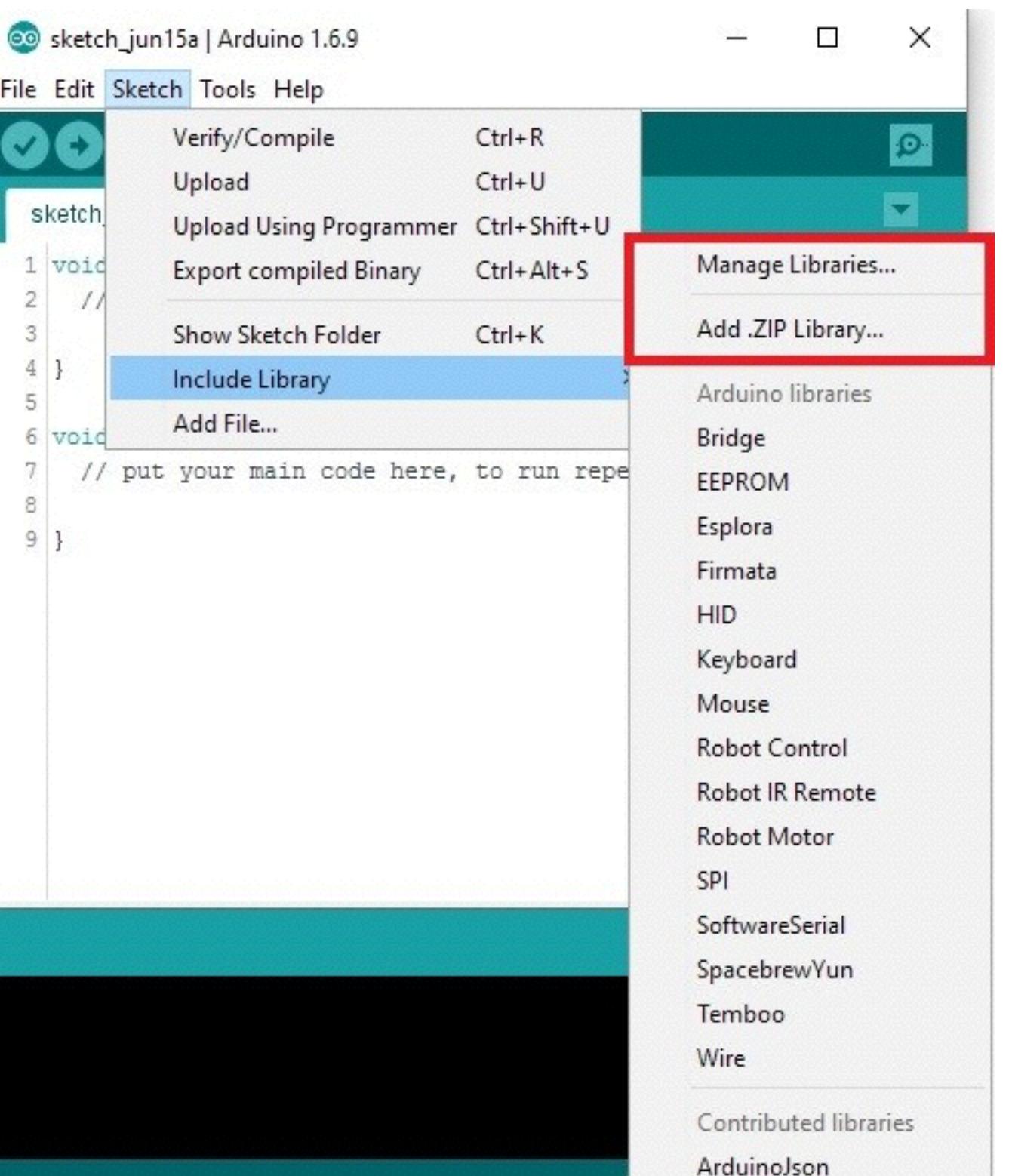


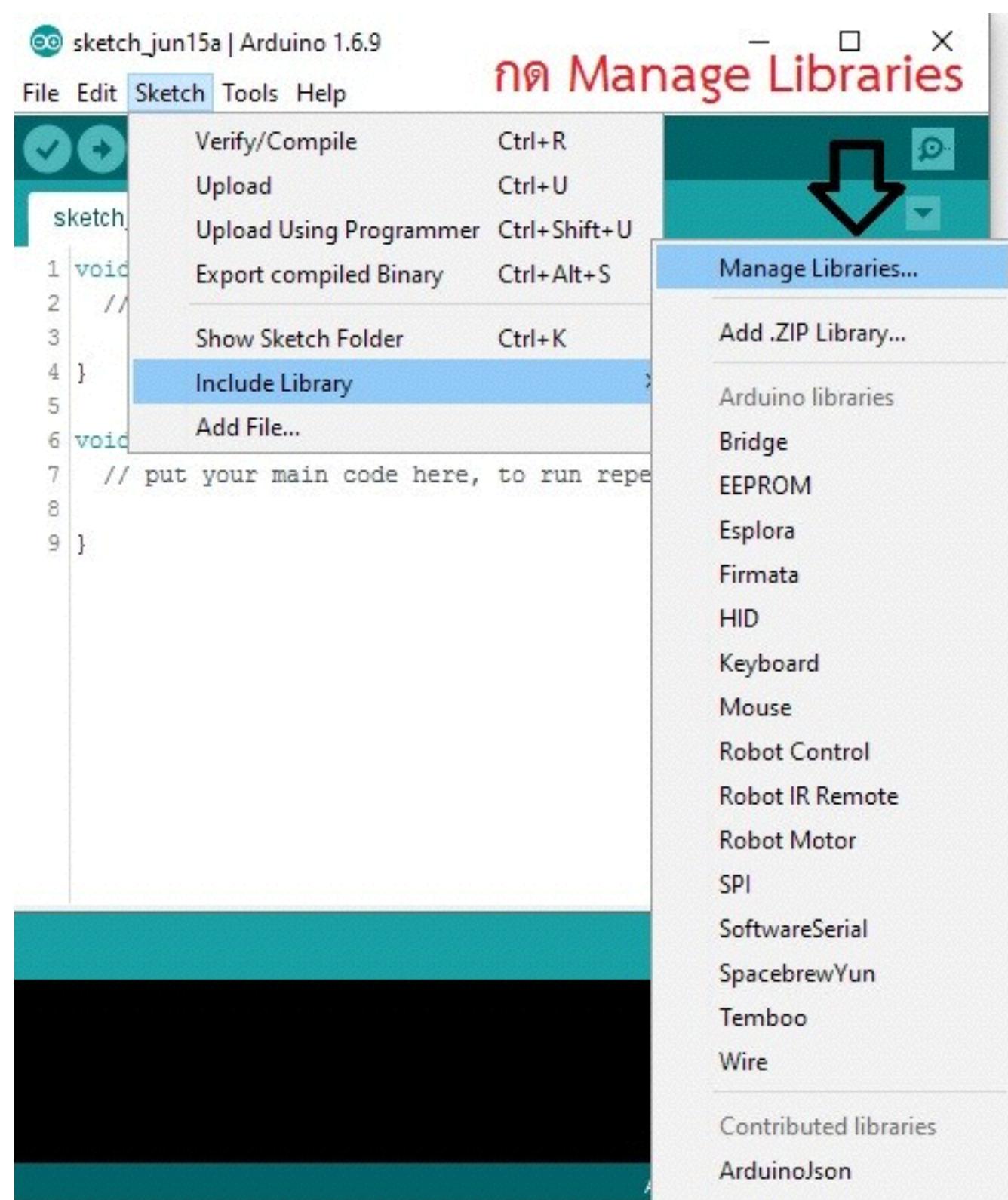
เลือก Version และกด Install

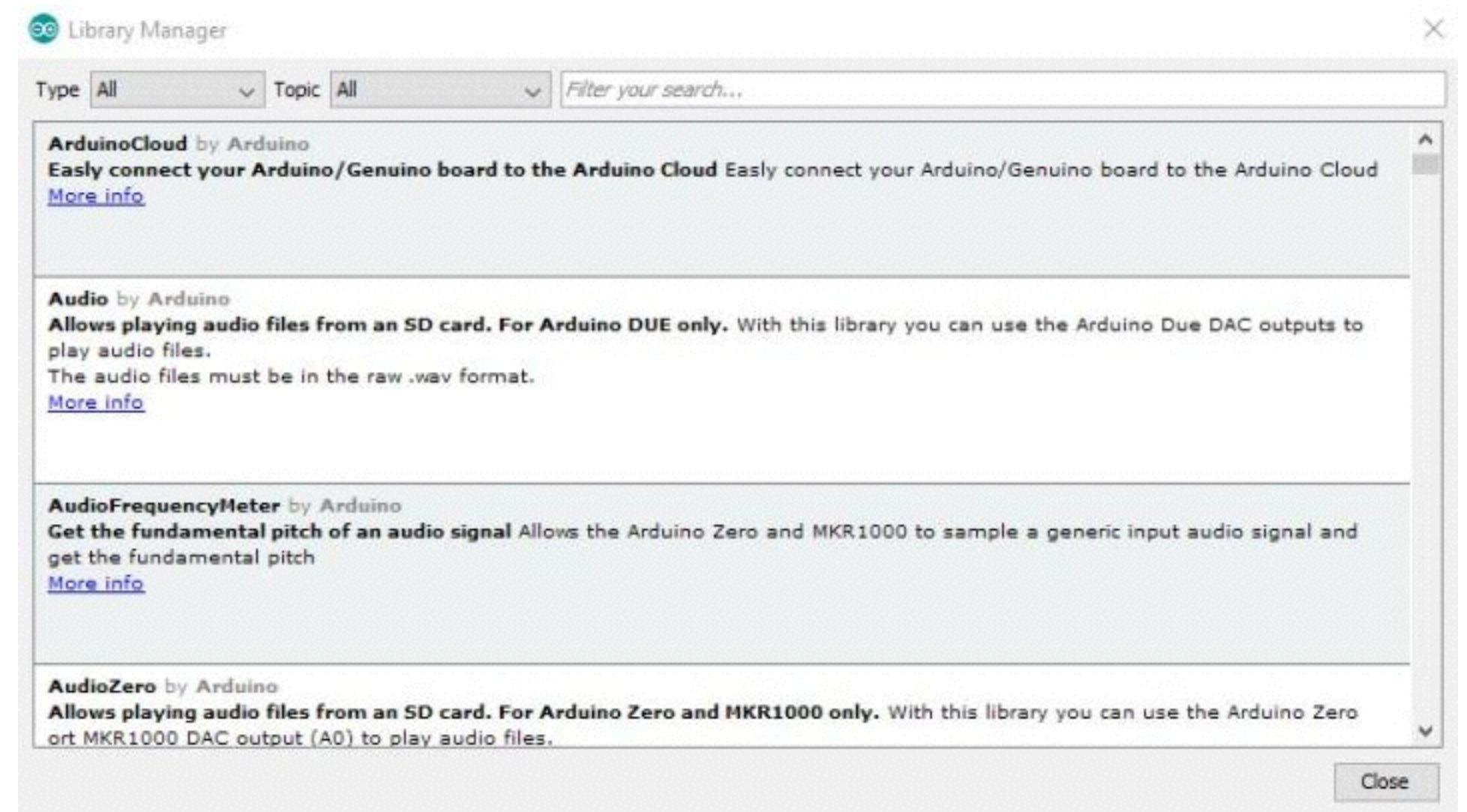


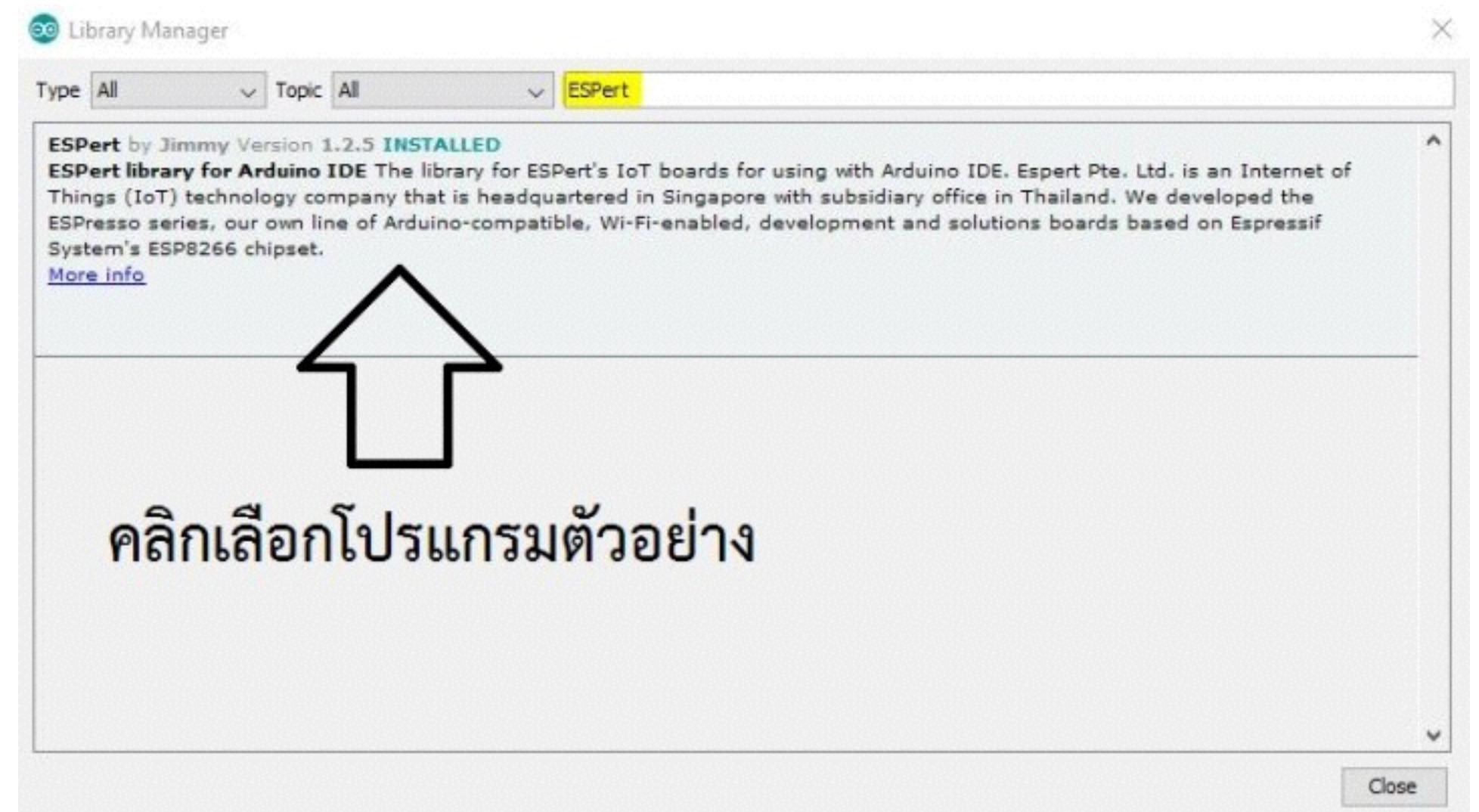


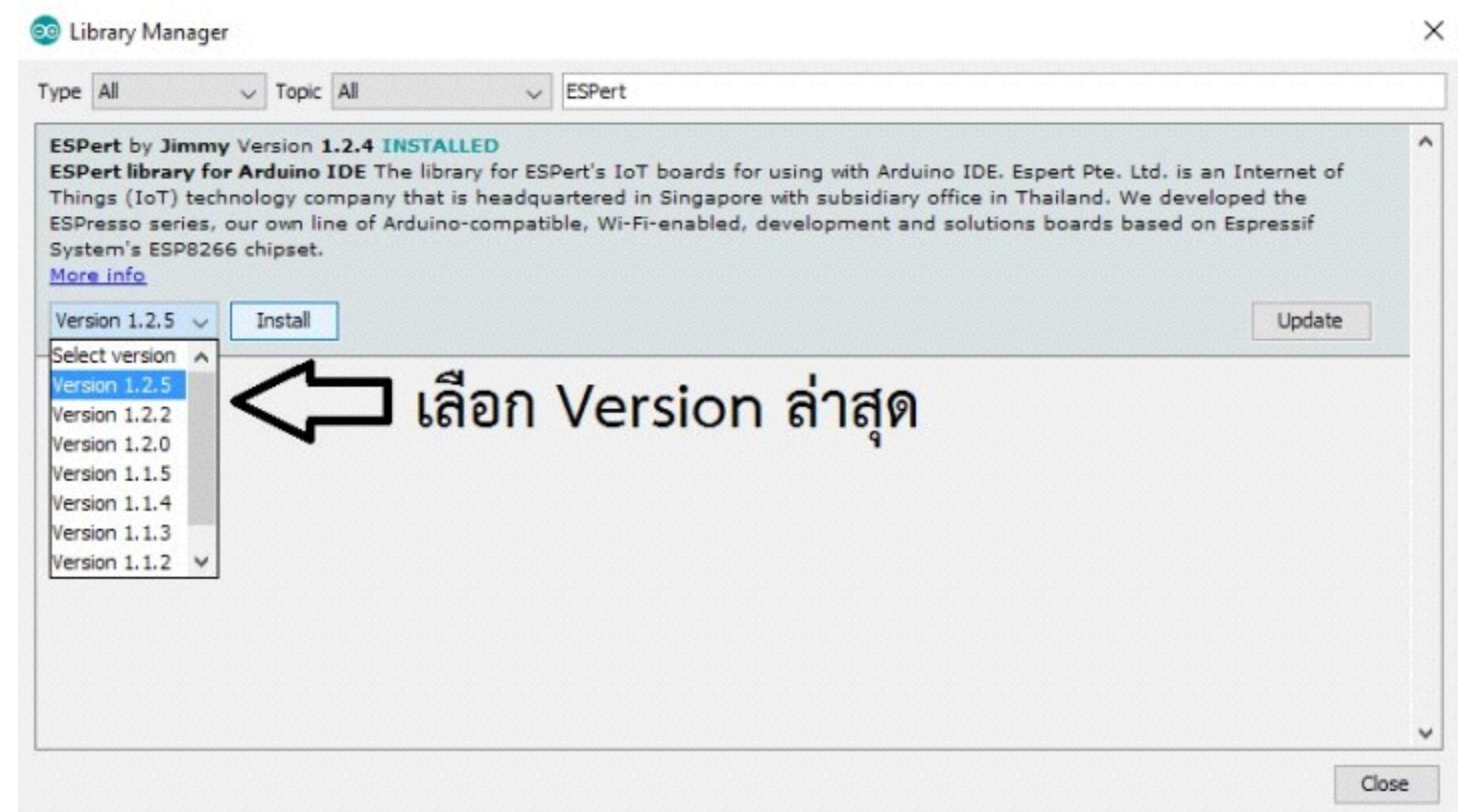
มีบอร์ด ESP8266 เพิ่มเข้ามาแล้ว

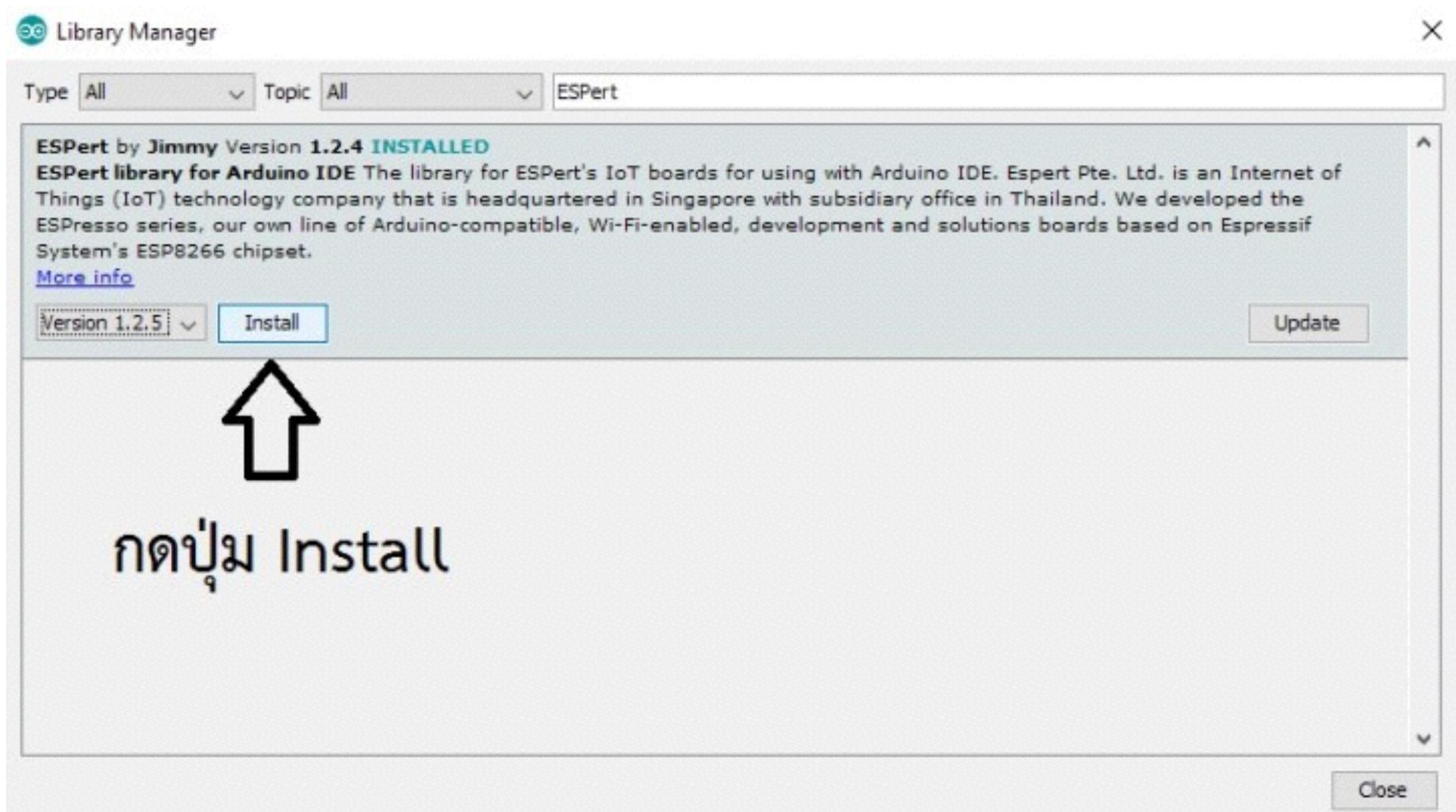


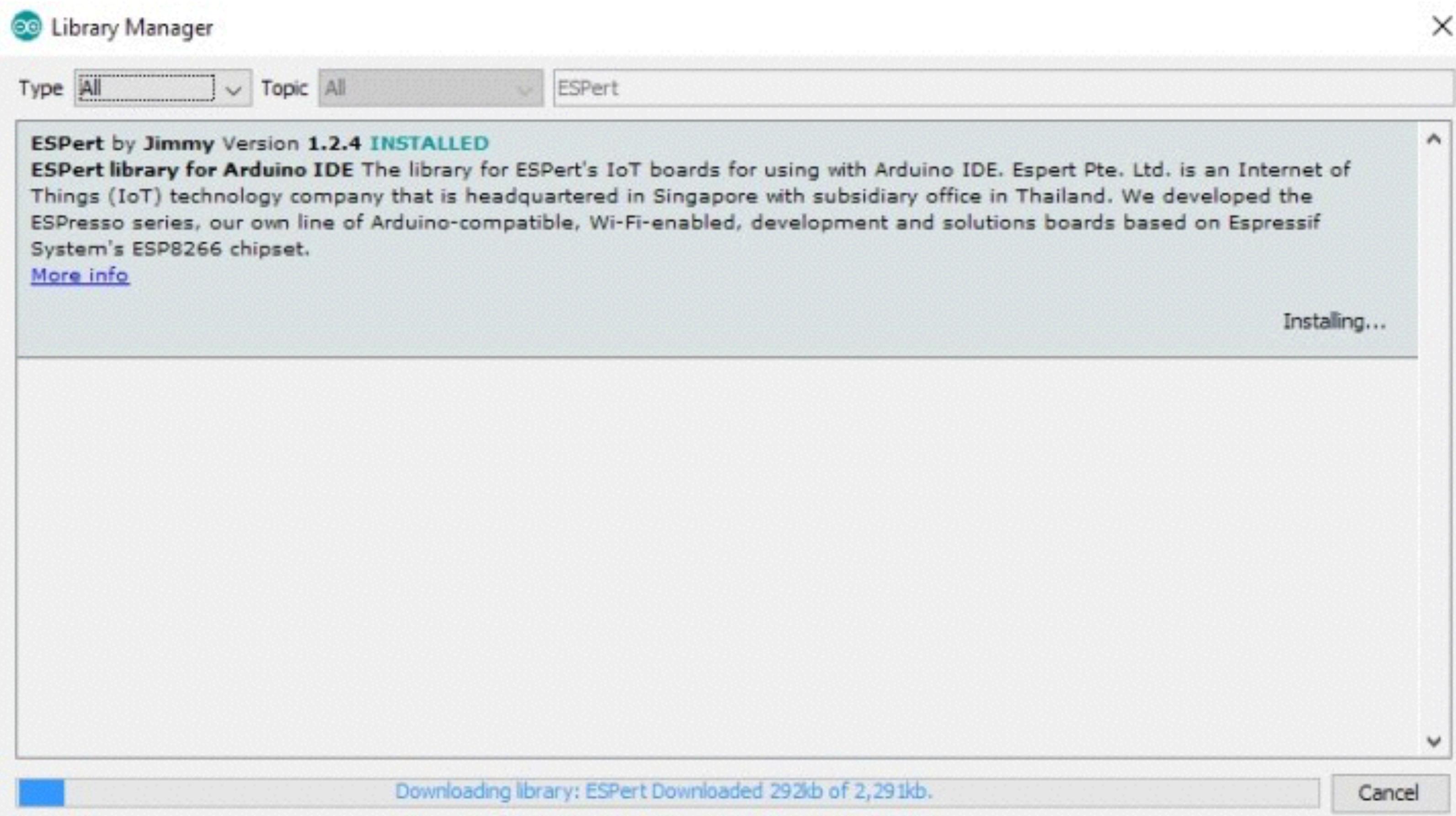


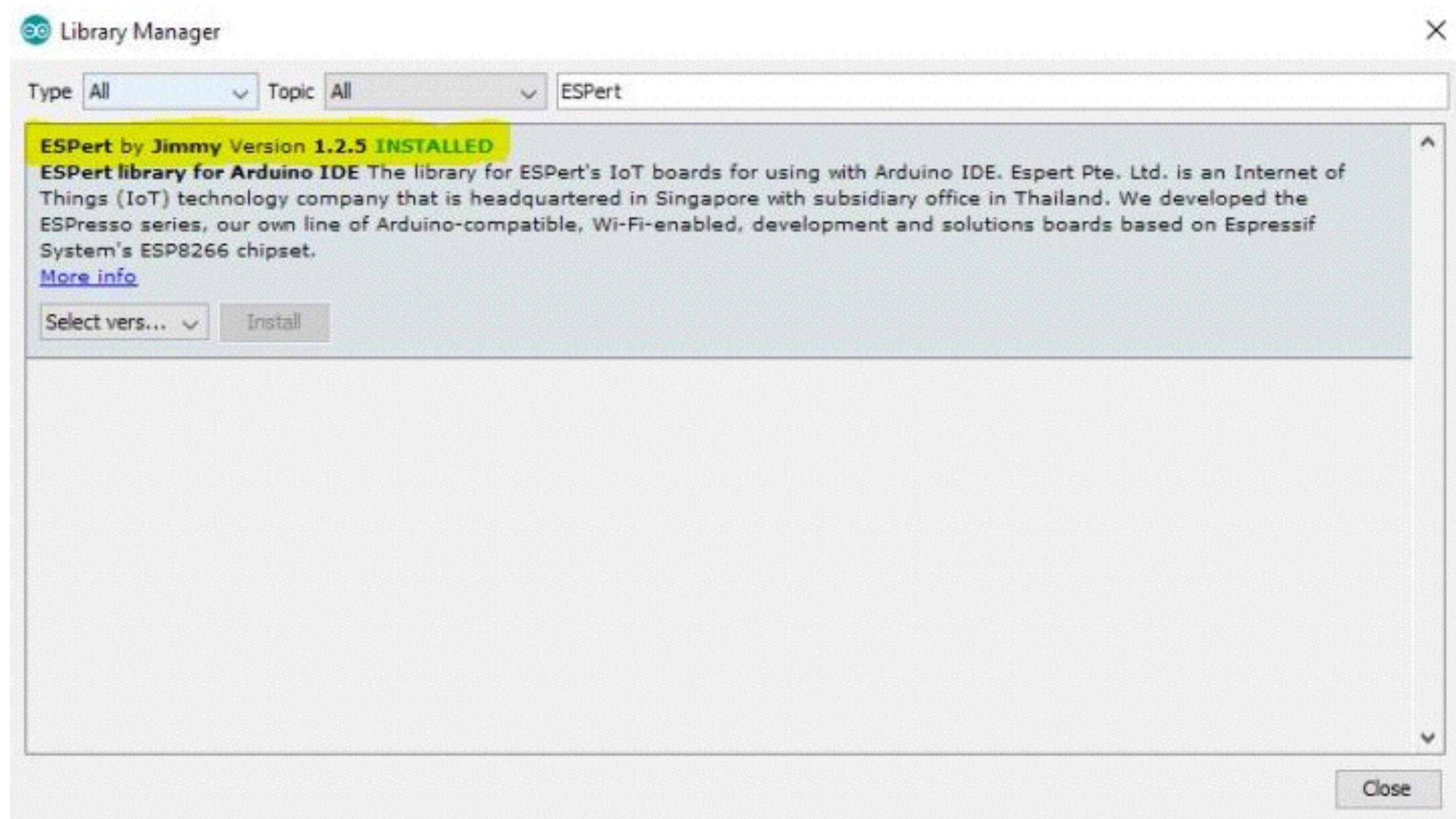












# ESP8266

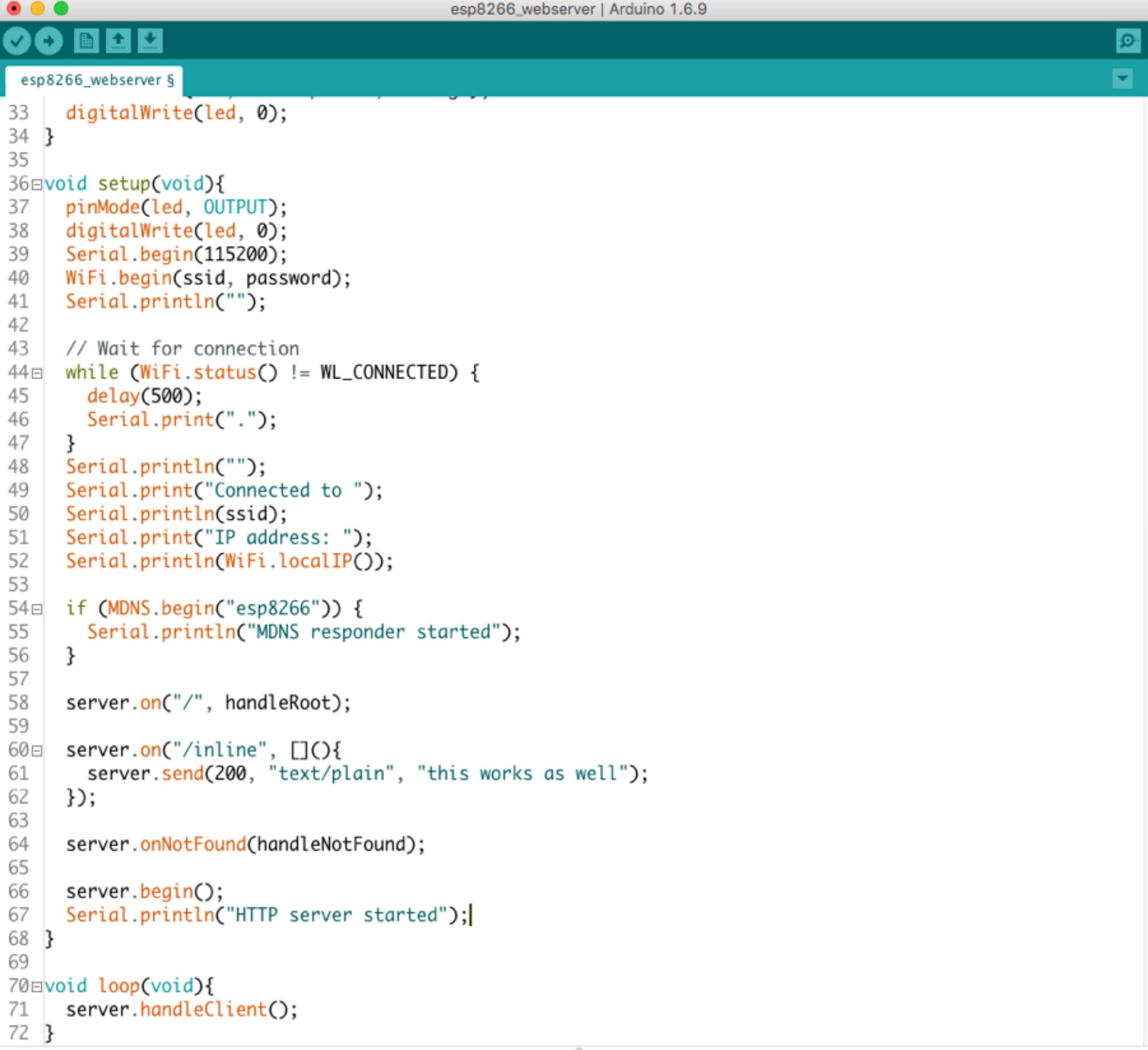
# Arduino Framework

# Connect to WiFi

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** esp8266\_sta | Arduino 1.6.9
- Code Editor:** The code is named `esp8266_sta`. It includes #includes for `ESP8266WiFi.h` and `WiFiClient.h`. It defines constants for SSID and password, and a variable for the LED pin. The `setup()` function initializes the LED, sets up Serial communication at 115200 bps, and attempts to connect to the WiFi network. It prints a message to the Serial monitor while waiting for connection. Once connected, it prints the SSID and local IP address. The `loop()` function is empty.
- Compile Output:** Shows "Done compiling." followed by the library path: "Using library ESP8266WiFi at version 1.7.0 in folder /Users/luke/Downloads/arduino-1.6.9/packages/ESP8266/".
- Sketch Statistics:** Displays memory usage: "Sketch uses 225,873 bytes (21%) of program storage space. Maximum is 1,044,464 bytes." and "Global variables use 31,800 bytes (38%) of dynamic memory, leaving 50,120 bytes for local variables. Max" (partially cut off).
- Bottom Status Bar:** Shows "CHIANG MAI MAKERCLUB" next to a logo, and "22" on the left.

# STA WebServer



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** esp8266\_webserver | Arduino 1.6.9
- Toolbar:** Standard Arduino IDE toolbar with icons for file operations.
- Code Editor:** The main window displays the following C++ code for an ESP8266 webserver:

```
esp8266_webserver §

33     digitalWrite(led, 0);
34 }
35
36 void setup(void){
37     pinMode(led, OUTPUT);
38     digitalWrite(led, 0);
39     Serial.begin(115200);
40     WiFi.begin(ssid, password);
41     Serial.println("");
42
43     // Wait for connection
44     while (WiFi.status() != WL_CONNECTED) {
45         delay(500);
46         Serial.print(".");
47     }
48     Serial.println("");
49     Serial.print("Connected to ");
50     Serial.println(ssid);
51     Serial.print("IP address: ");
52     Serial.println(WiFi.localIP());
53
54     if (MDNS.begin("esp8266")) {
55         Serial.println("MDNS responder started");
56     }
57
58     server.on("/", handleRoot);
59
60     server.on("/inline", [](){
61         server.send(200, "text/plain", "this works as well");
62     });
63
64     server.onNotFound(handleNotFound);
65
66     server.begin();
67     Serial.println("HTTP server started");
68 }
69
70 void loop(void){
71     server.handleClient();
72 }
```
- Status Bar:** Done Saving.

# AP WebServer



# Basic HTTP Get

esp8266\_basic\_httpget | Arduino 1.6.9

```
27 Serial.println("");
28 Serial.print("Connected to ");
29 Serial.println(ssid);
30 Serial.print("IP address: ");
31 Serial.println(WiFi.localIP());
32
33 }
34
35 void loop() {
36   HTTPClient http;
37
38   Serial.print("[HTTP] begin...\n");
39   http.begin("http://192.168.1.12/test.html"); //HTTP
40
41   Serial.print("[HTTP] GET...\n");
42   // start connection and send HTTP header
43   int httpCode = http.GET();
44
45   // httpCode will be negative on error
46   if (httpCode > 0) {
47     // HTTP header has been send and Server response header has been handled
48     Serial.printf("[HTTP] GET... code: %d\n", httpCode);
49
50     // file found at server
51     if (httpCode == HTTP_CODE_OK) {
52       String payload = http.getString();
53       Serial.println(payload);
54     }
55   } else {
56     Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
57   }
58
59   http.end();
60
61
62   delay(10000);
63 }
```

No valid hardware definitions found in folder esp8266com.  
WARNING: Error loading hardware folder /Users/Nat/Documents/Arduino/hardware/esp8266com  
No valid hardware definitions found in folder esp8266com.  
/Users/Nat/labesp8266/basic-esp8266/esp8266\_basic\_httpget/esp8266\_basic\_httpget.ino

20 ESPRESSO LITE 2.0, 80 MHz, 115200, 4M (3M SPIFFS), ck, Disabled, None on /dev/cu.usbserial-A7CX2S3 CHANG MAI MAKERCLUB