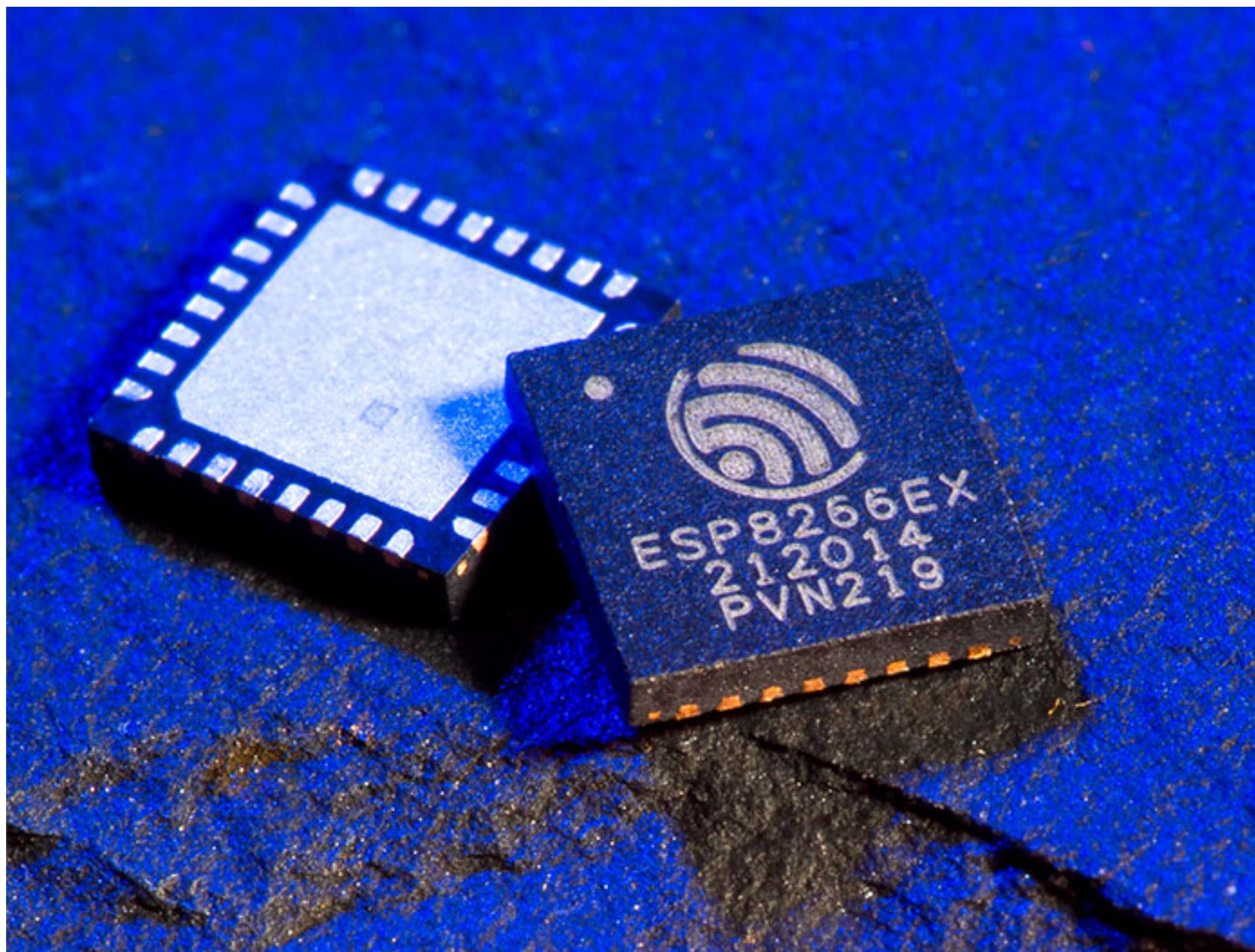
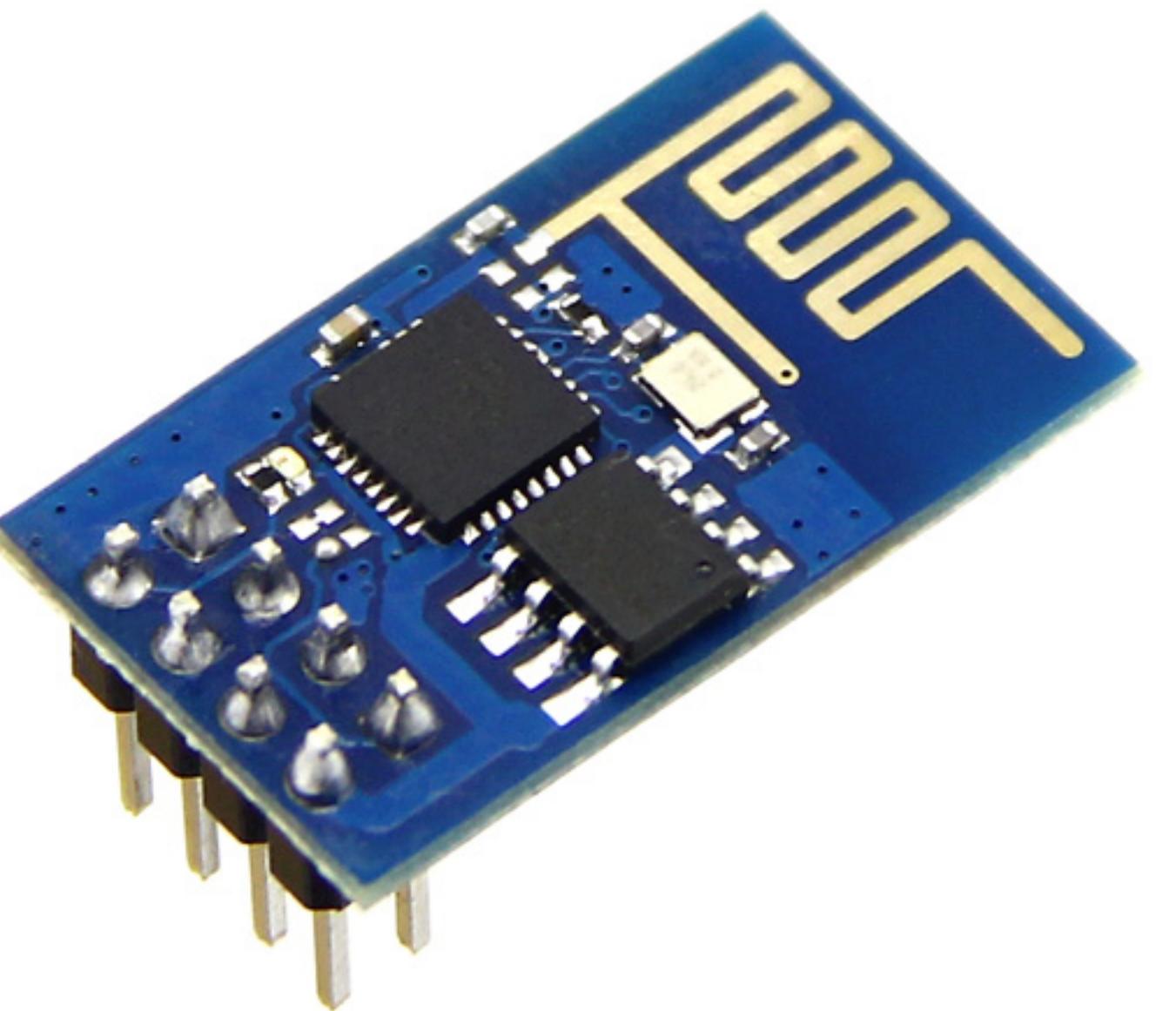
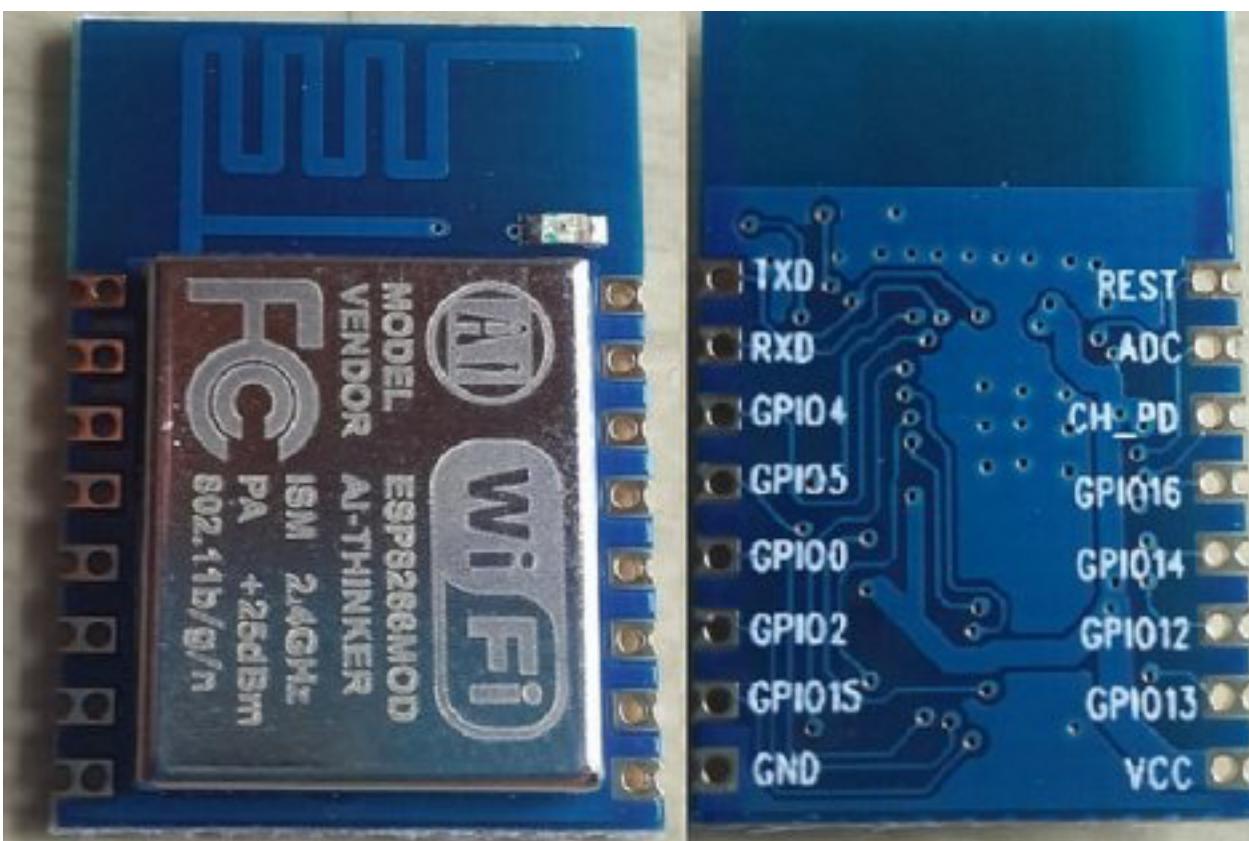


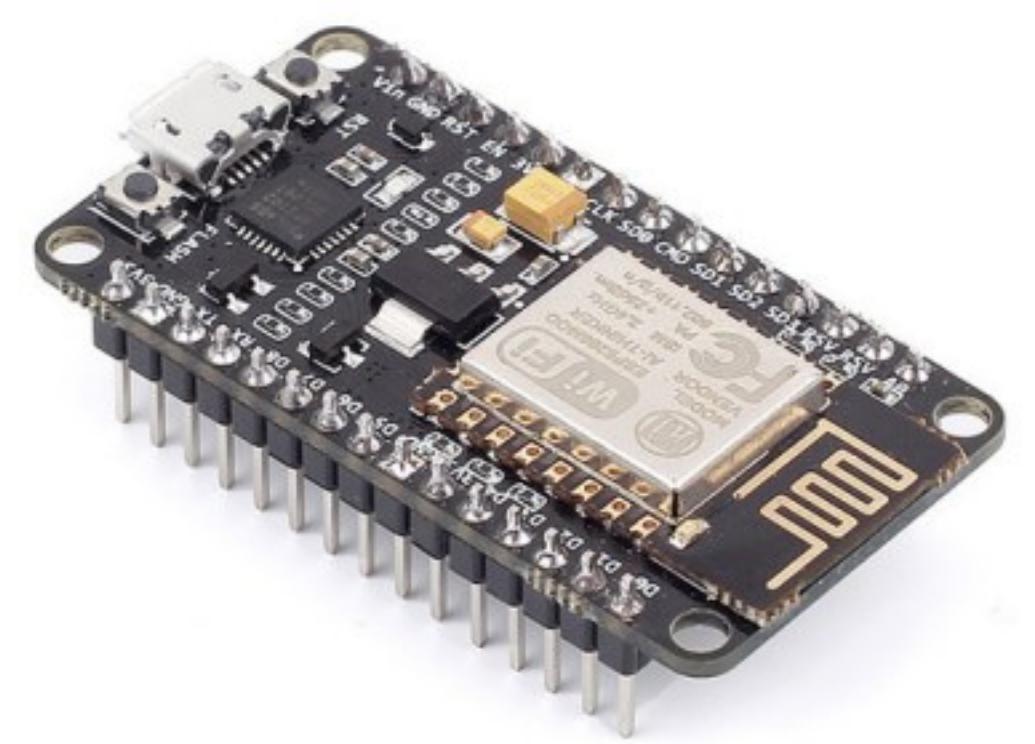
ESP8266 & Arduino IDE

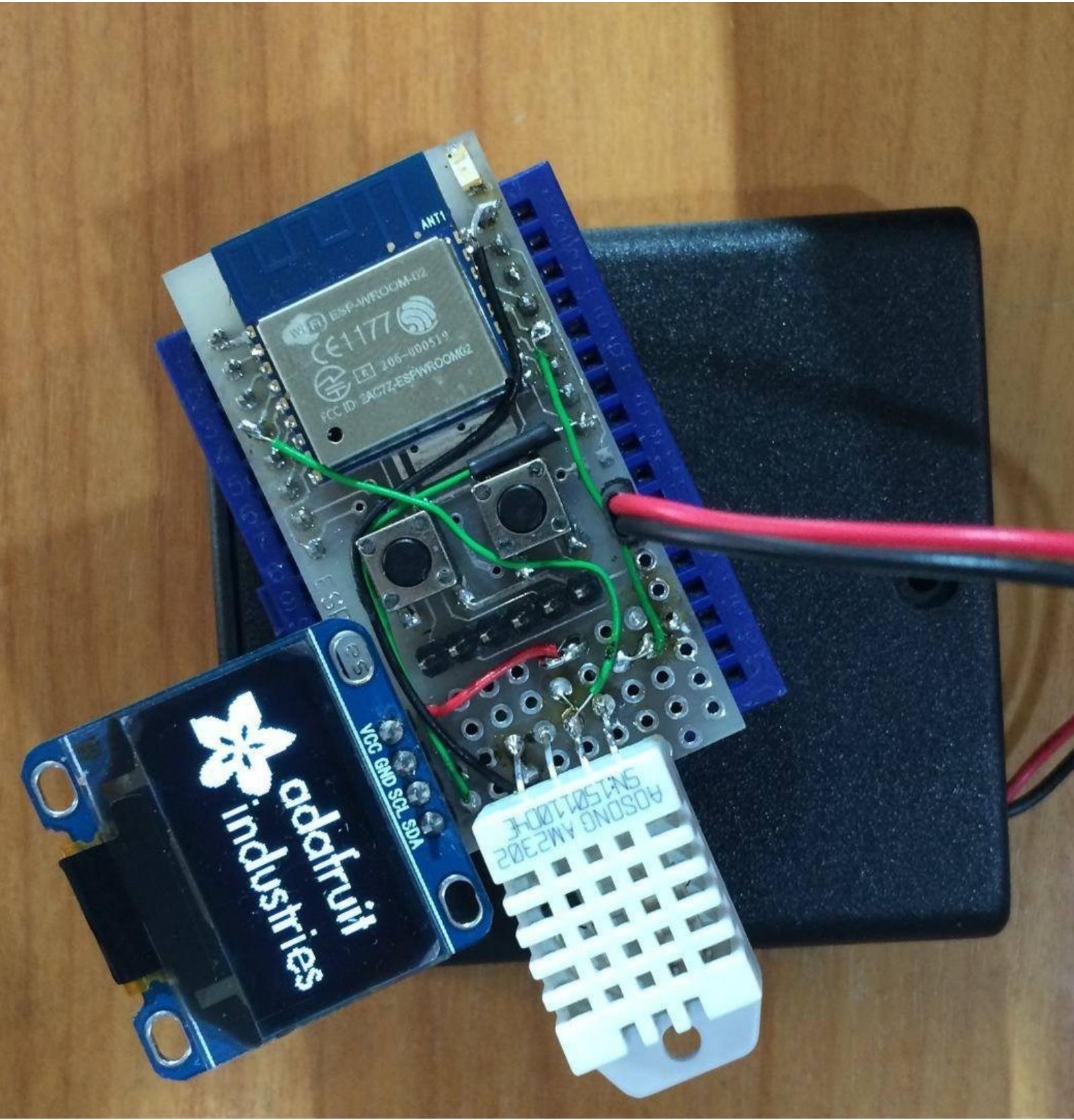
The ESPresso Lite



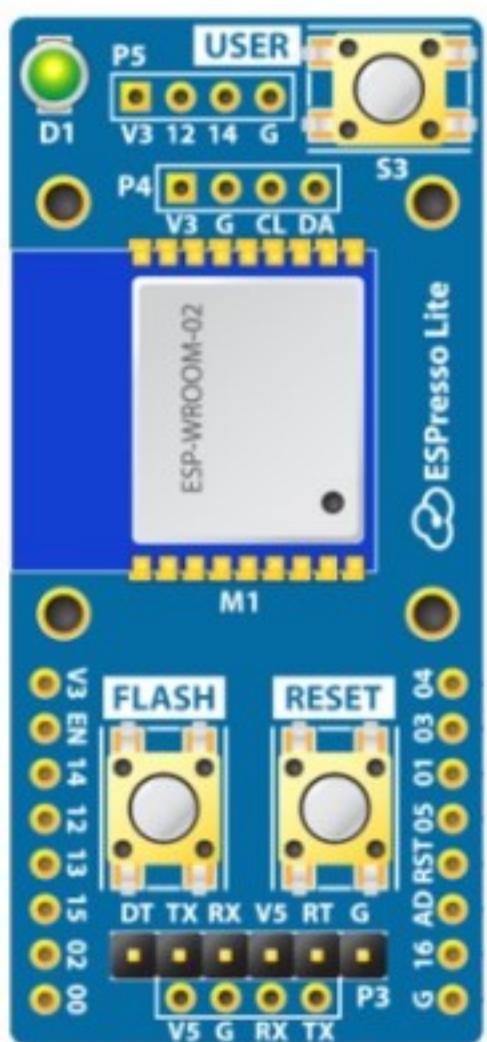




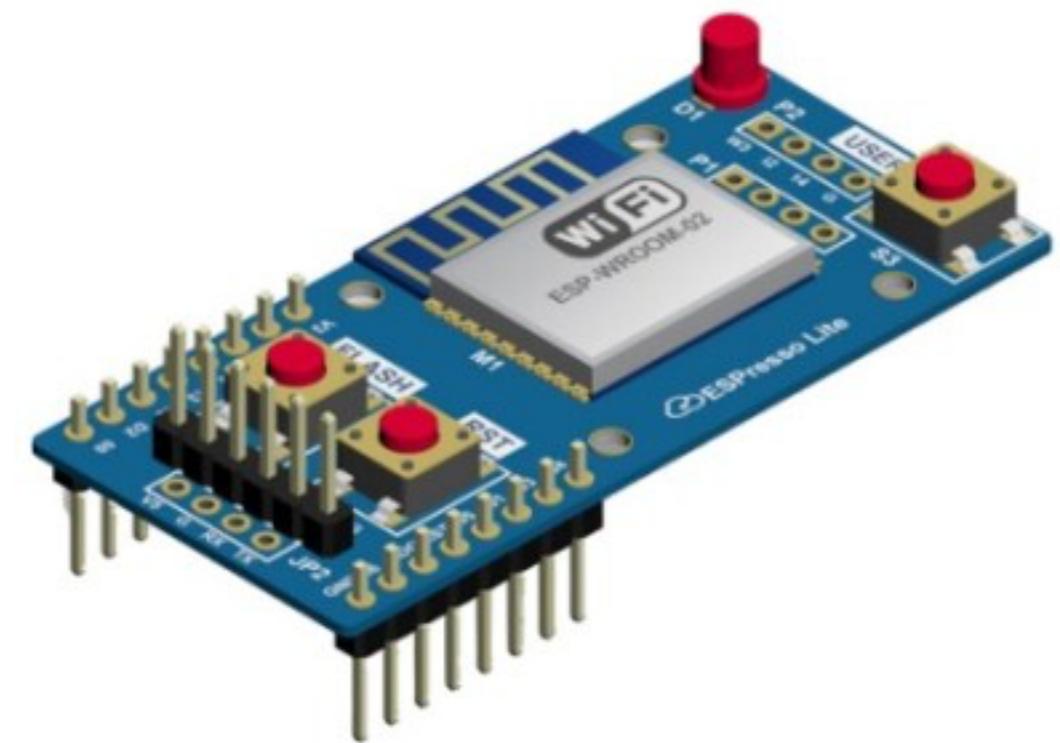


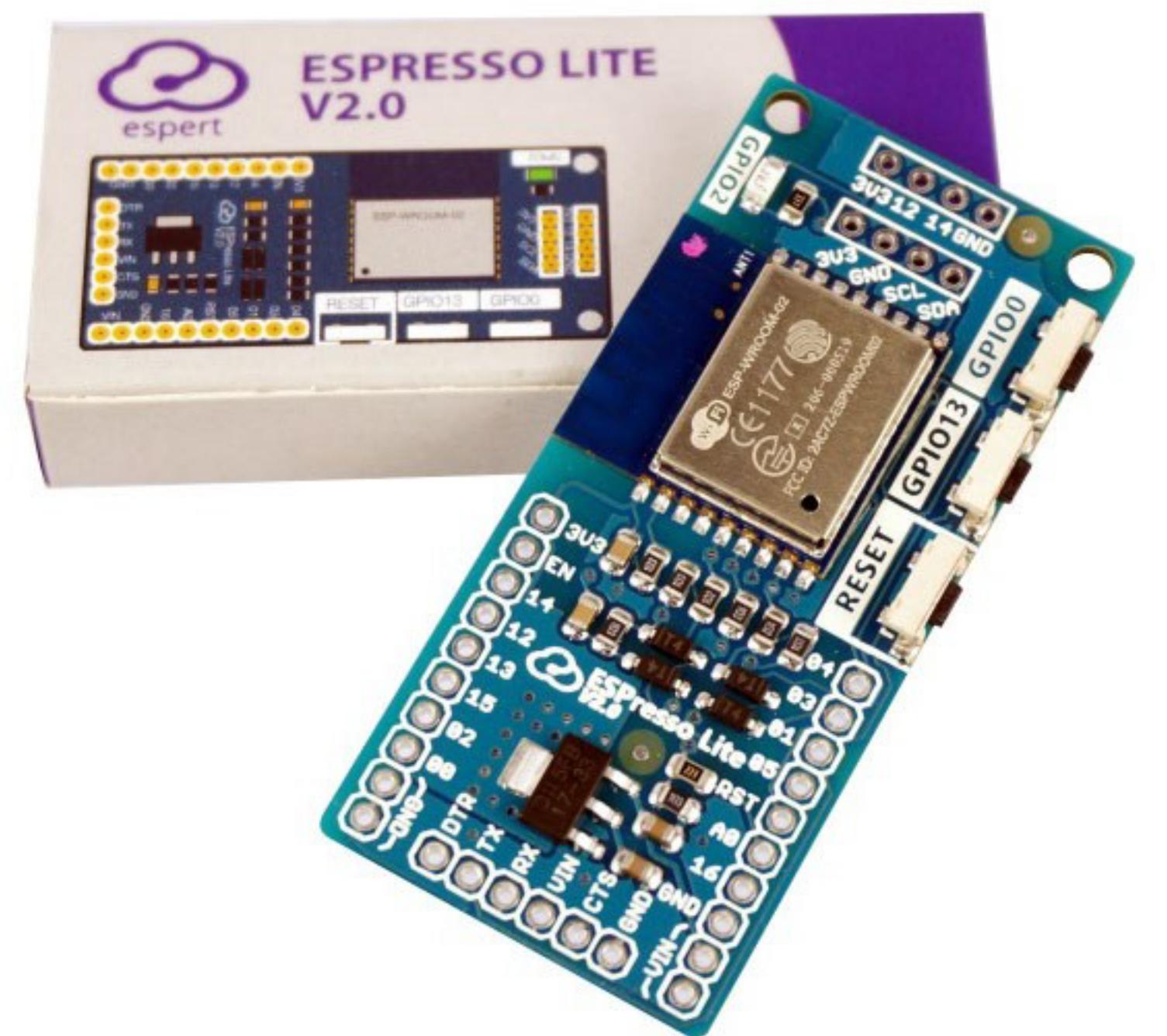


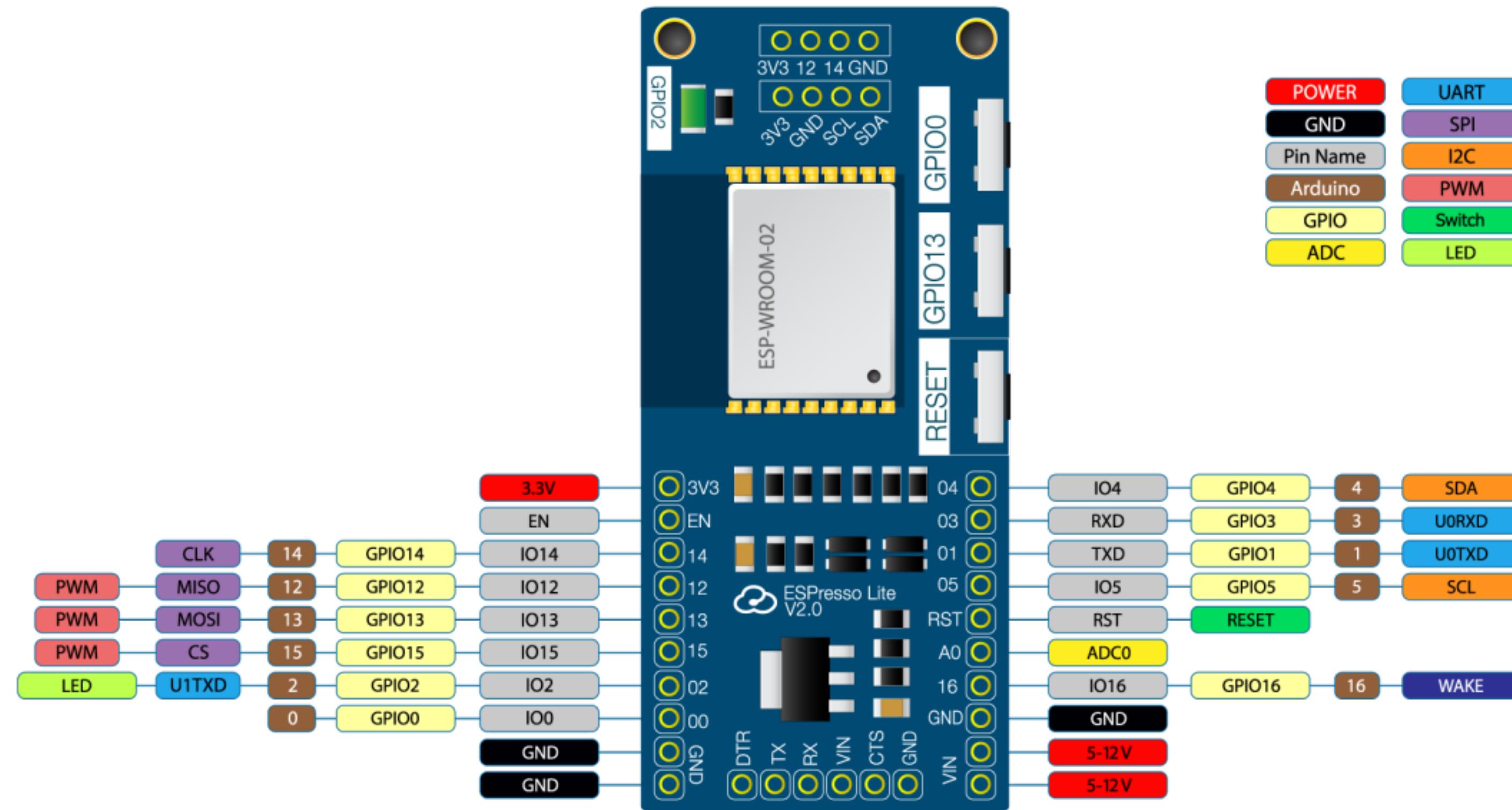
ESPresso Lite



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







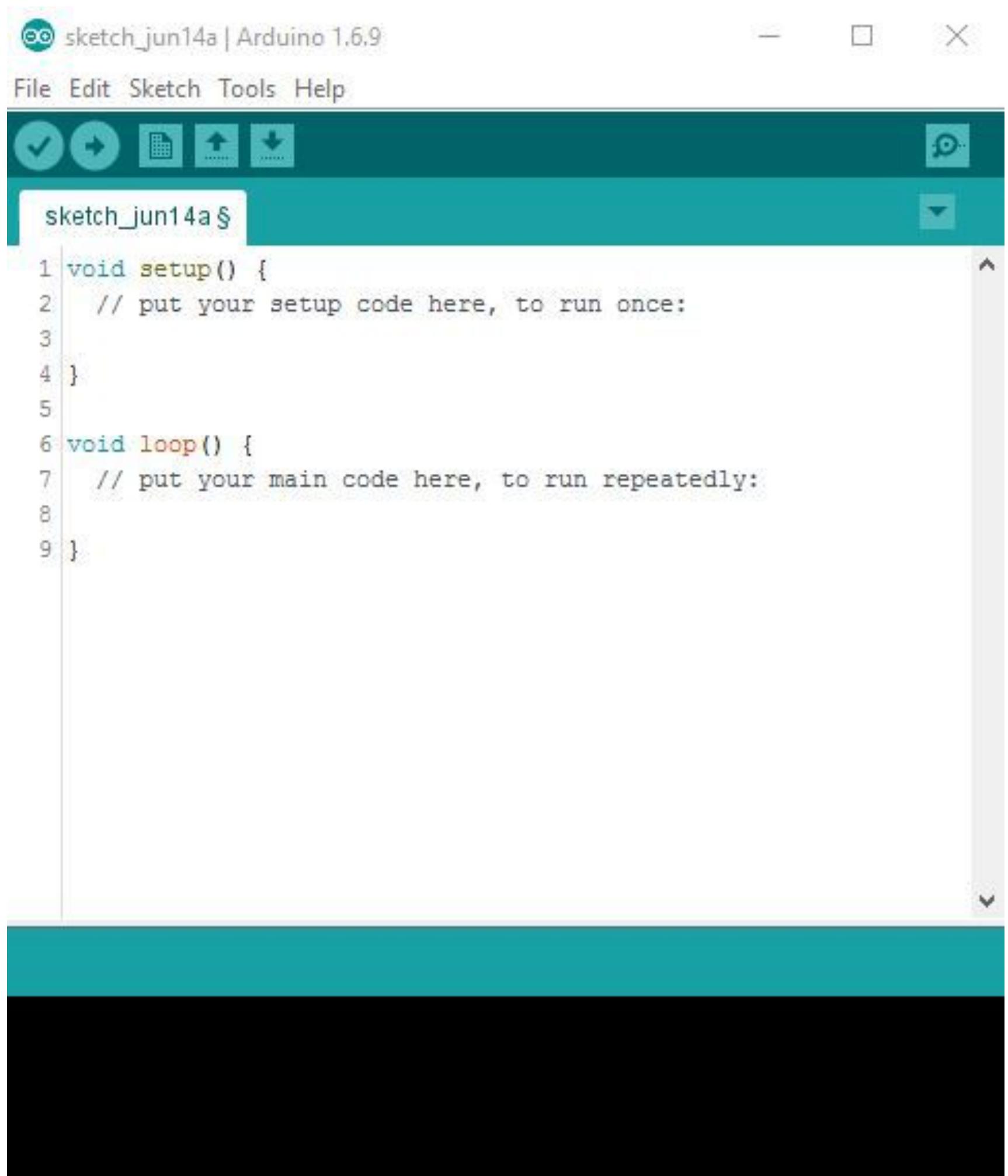
ESPresso Lite V2.0

- 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

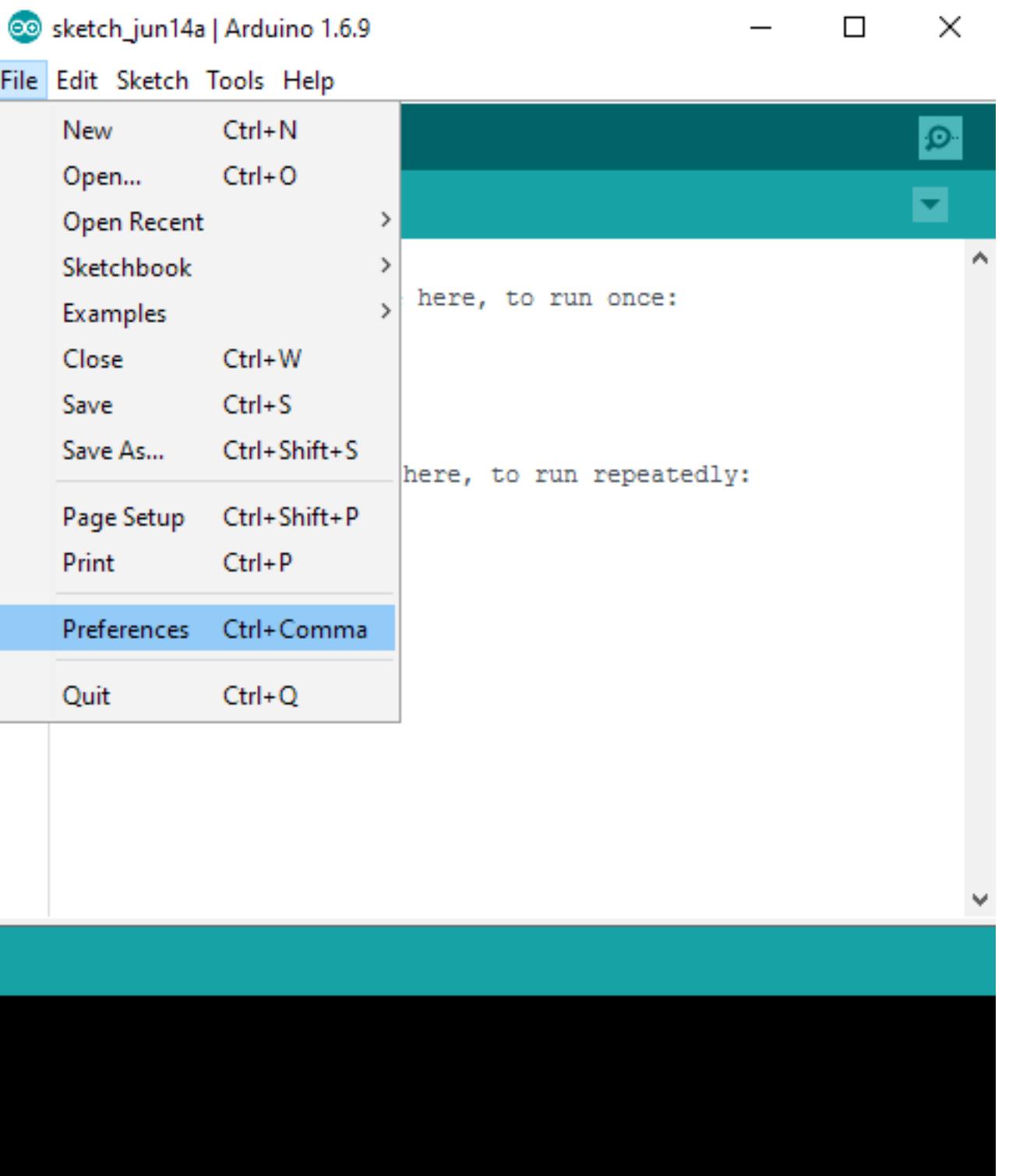


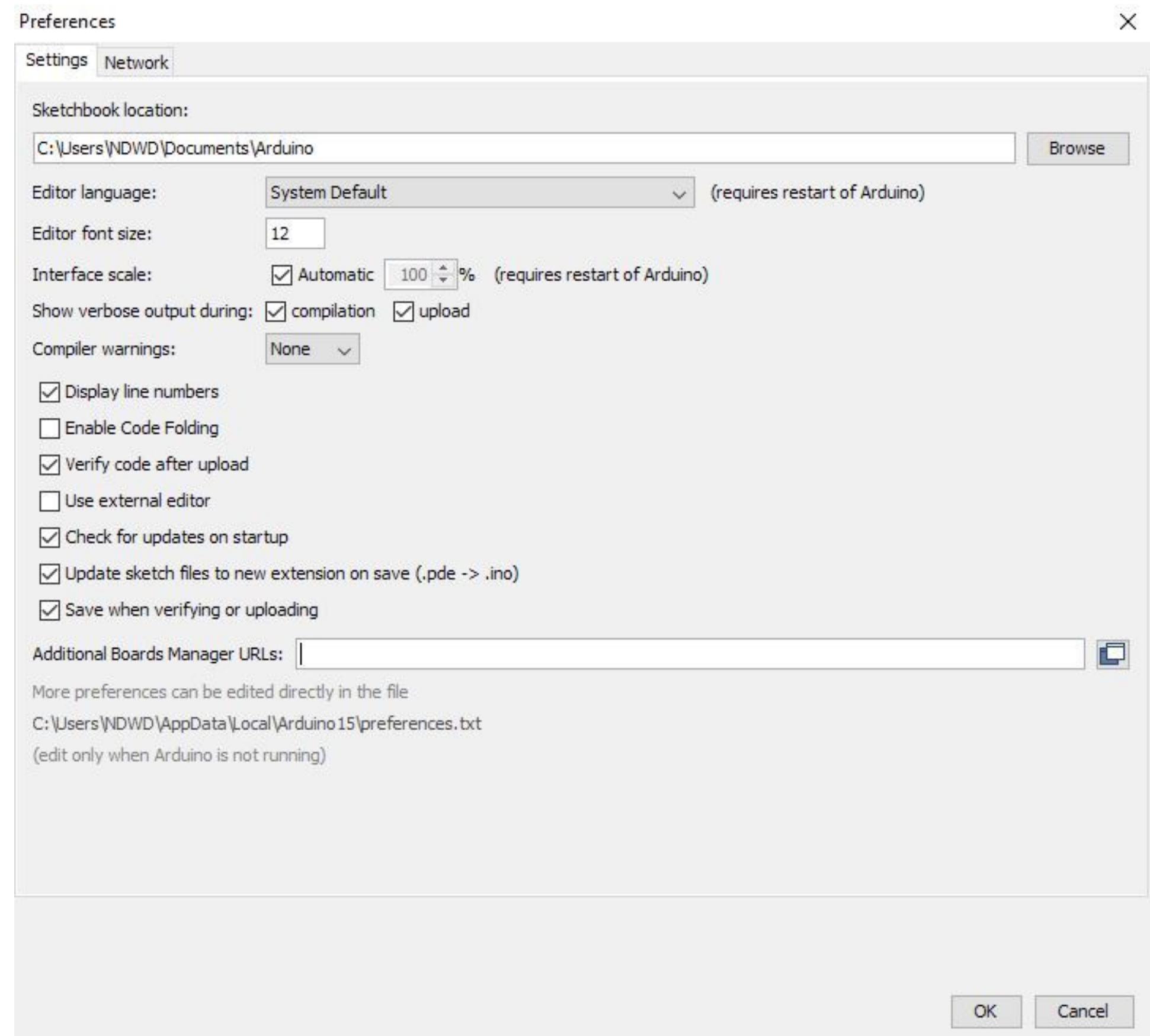


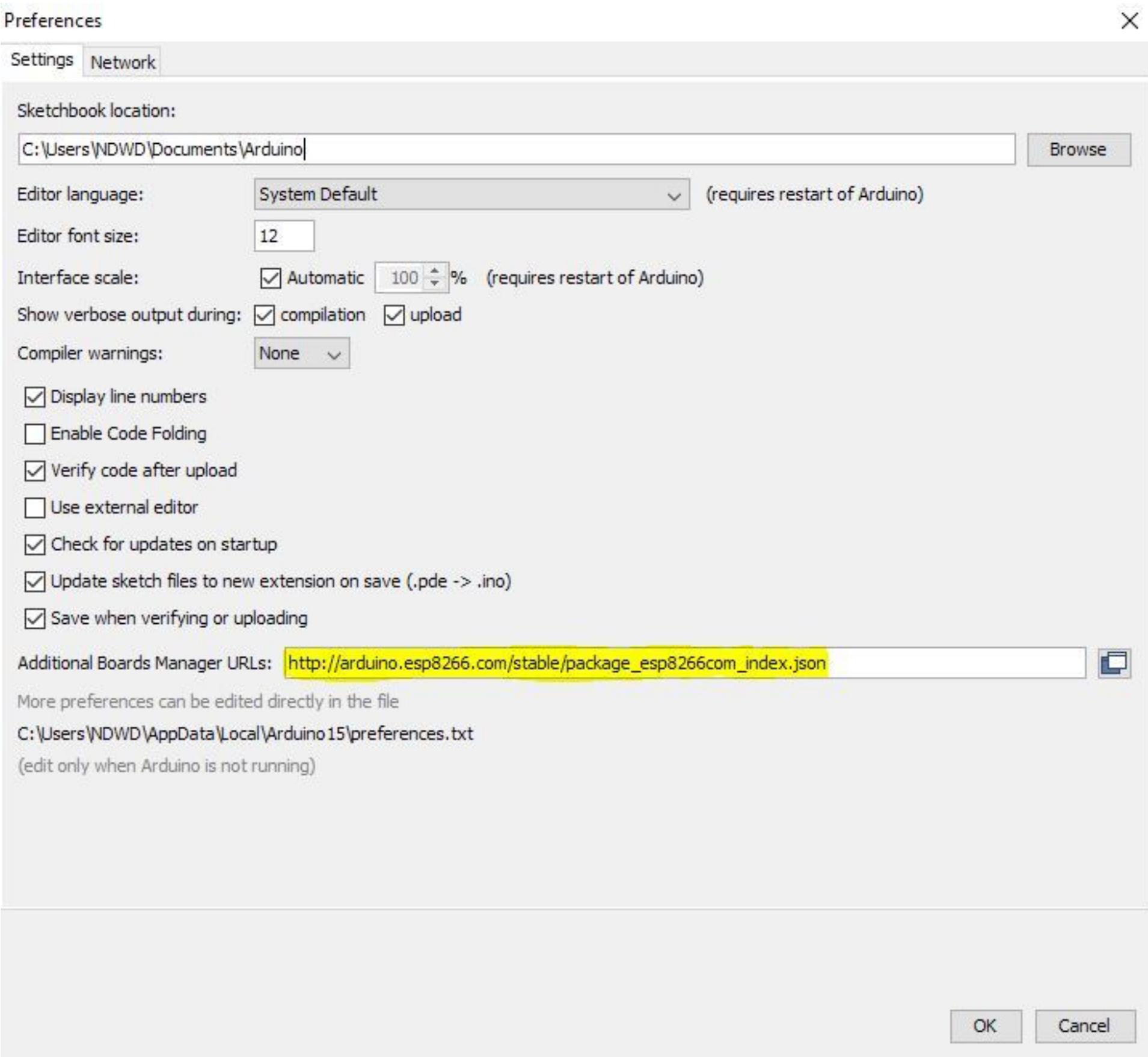


The screenshot shows the Arduino IDE interface. The title bar reads "sketch_jun14a | Arduino 1.6.9". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for file operations like Open, Save, and Print. The main code editor window contains the following code:

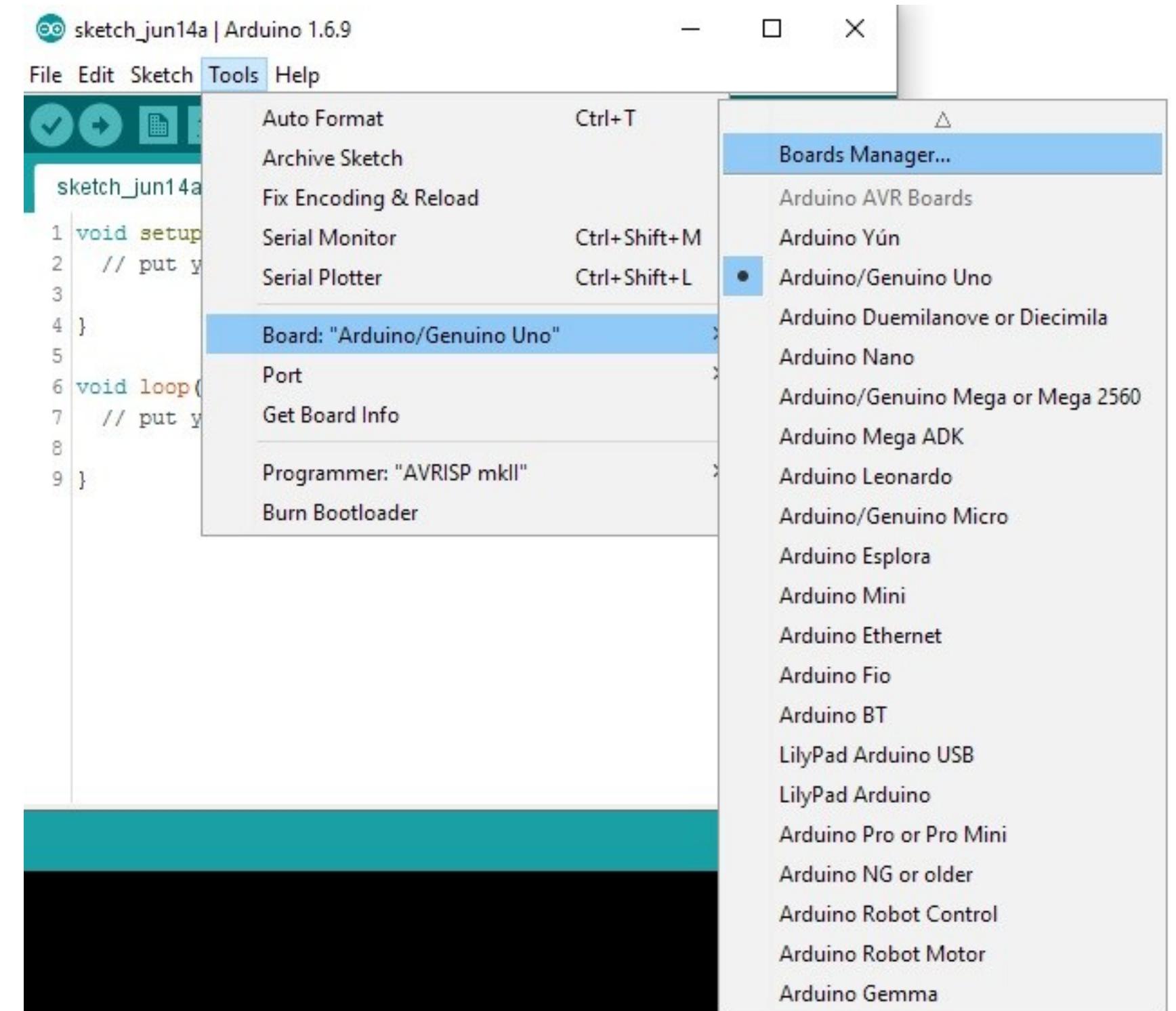
```
1 void setup() {  
2     // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7     // put your main code here, to run repeatedly:  
8 }  
9 }
```

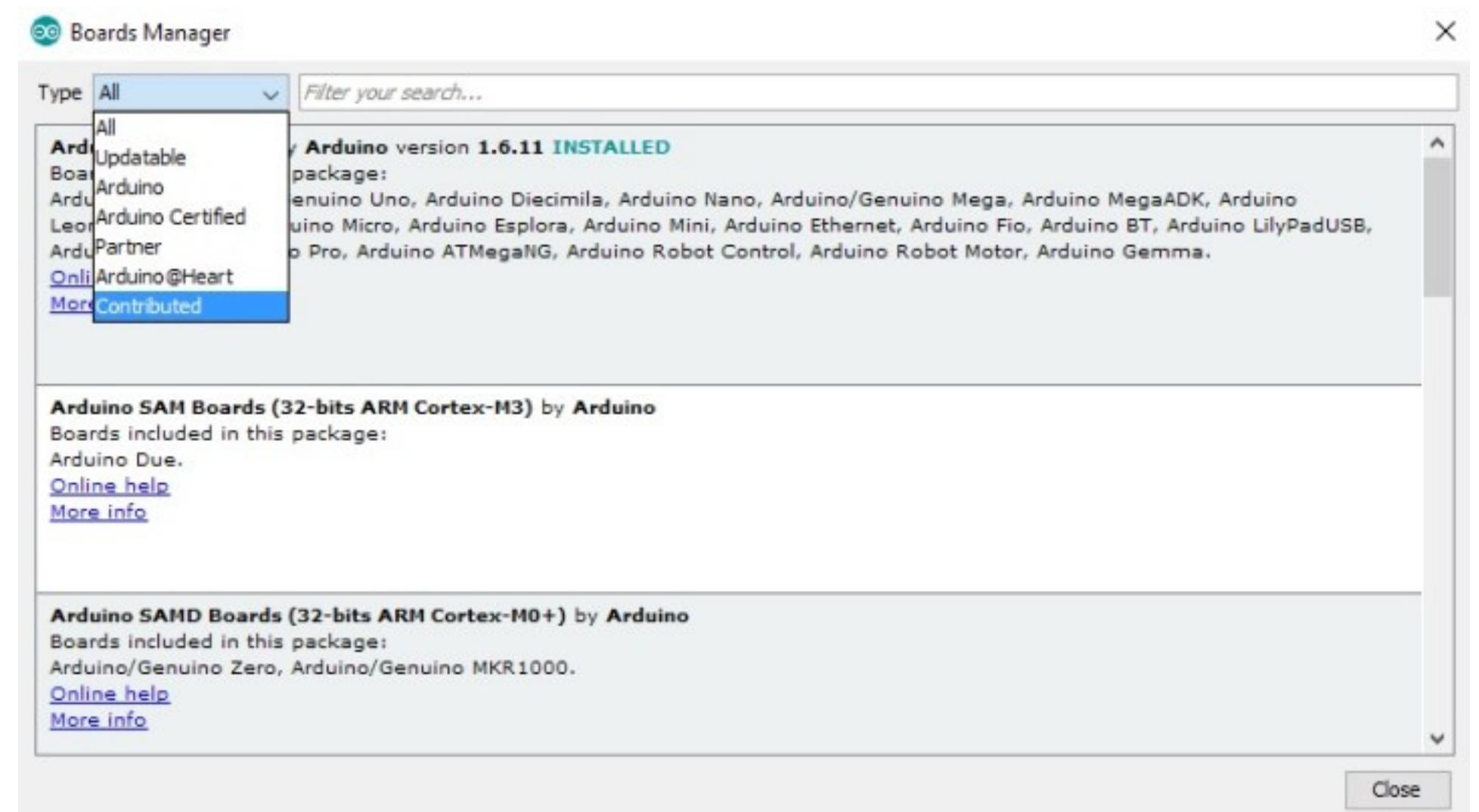


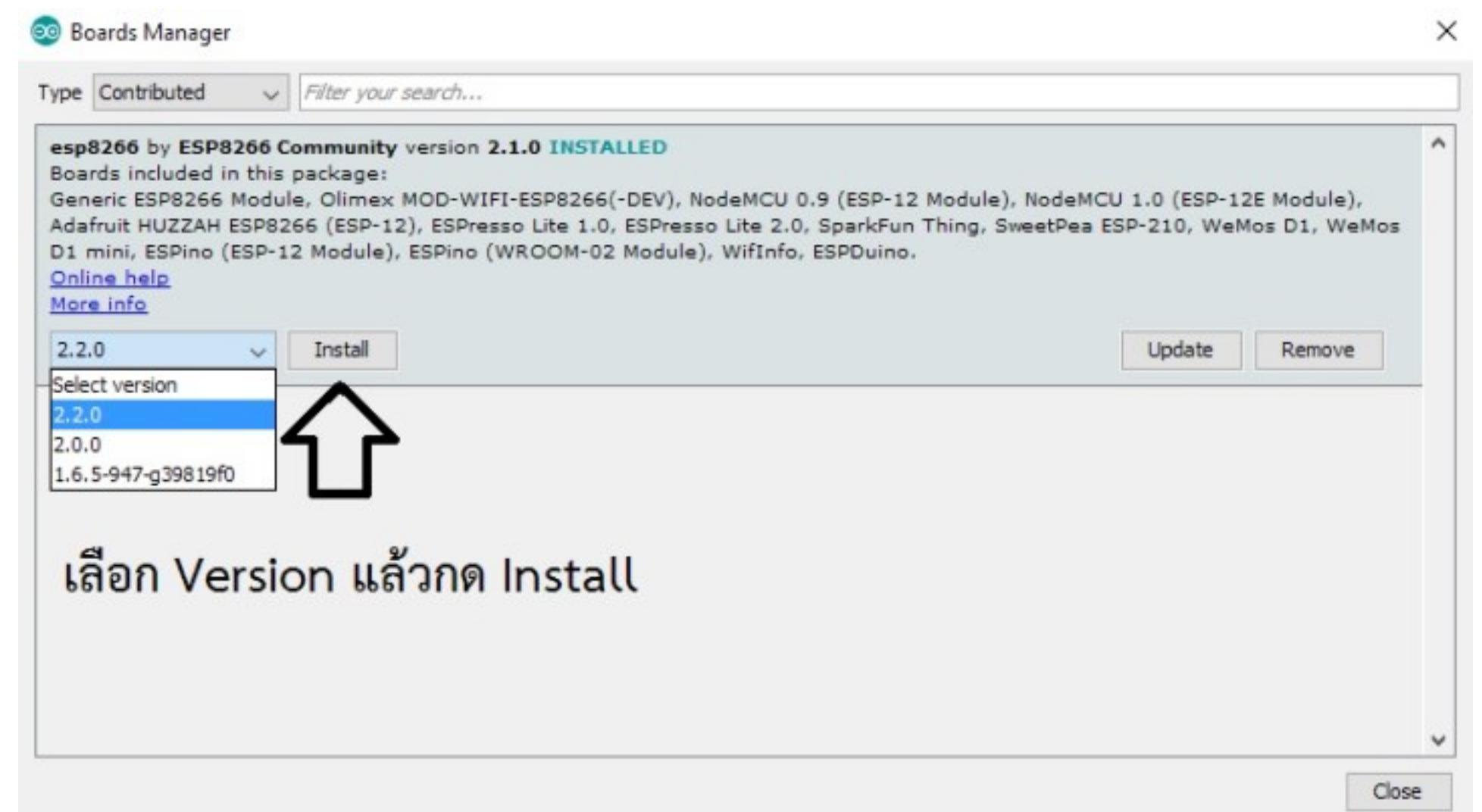


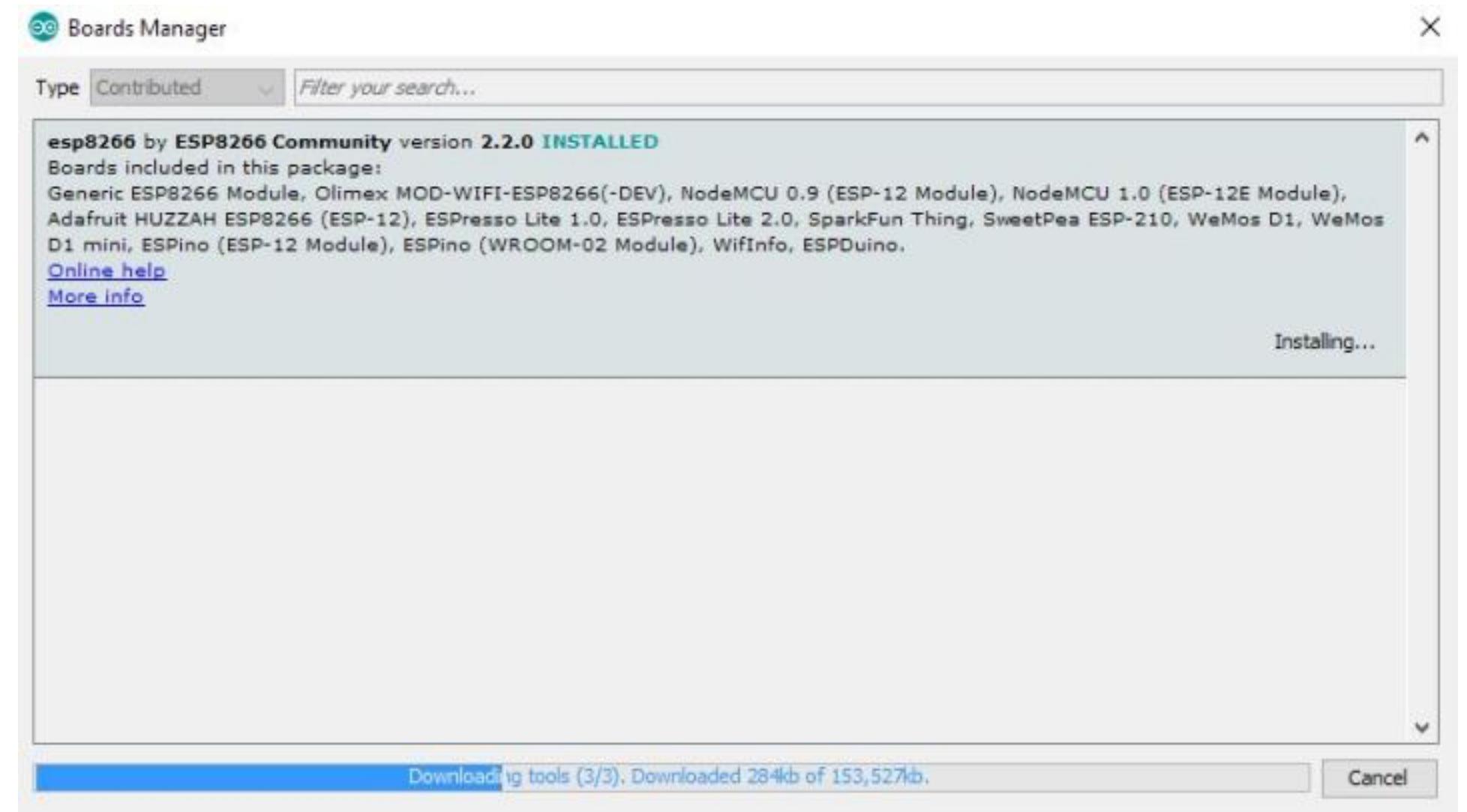


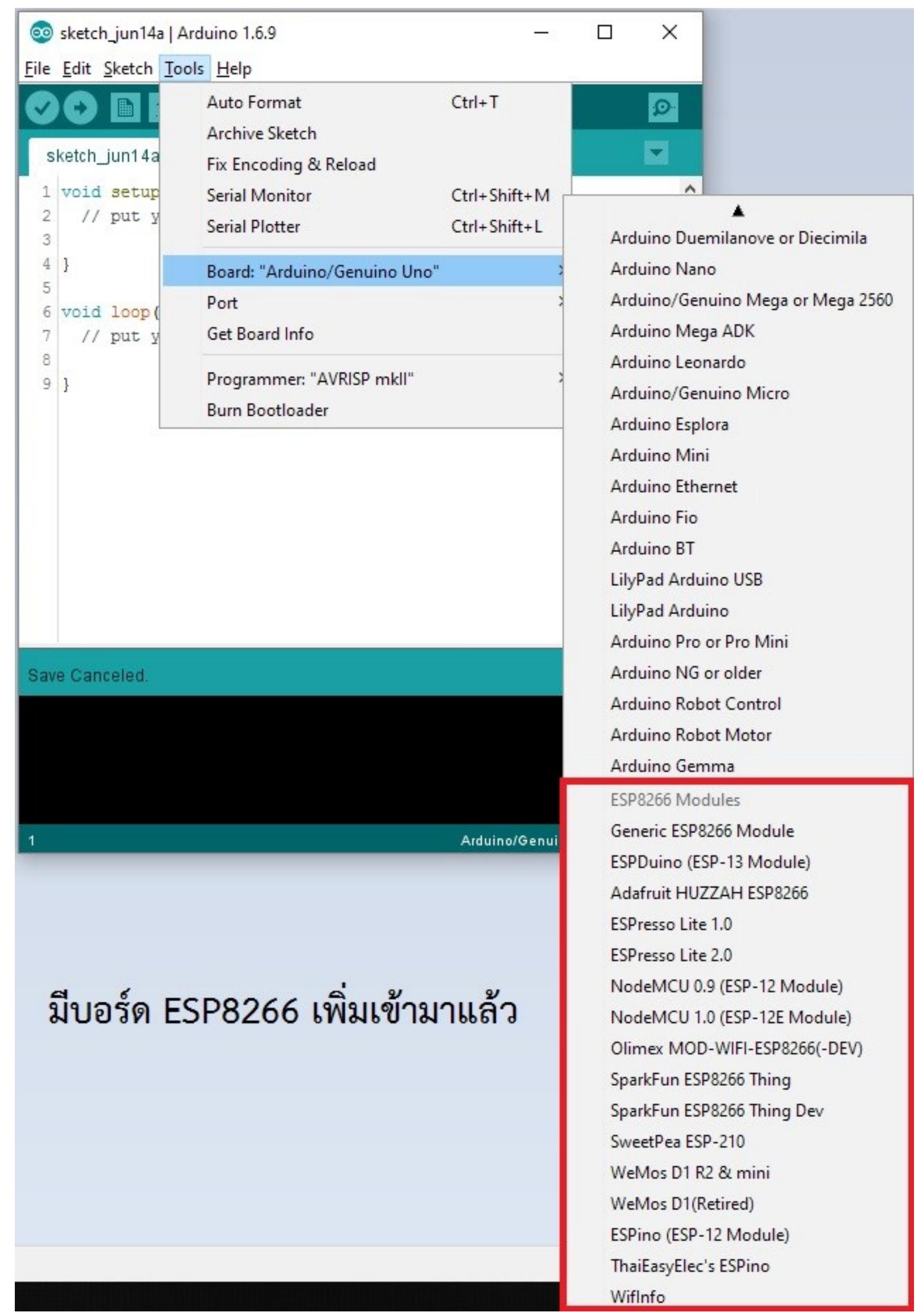
<https://github.com/esp8266/Arduino>



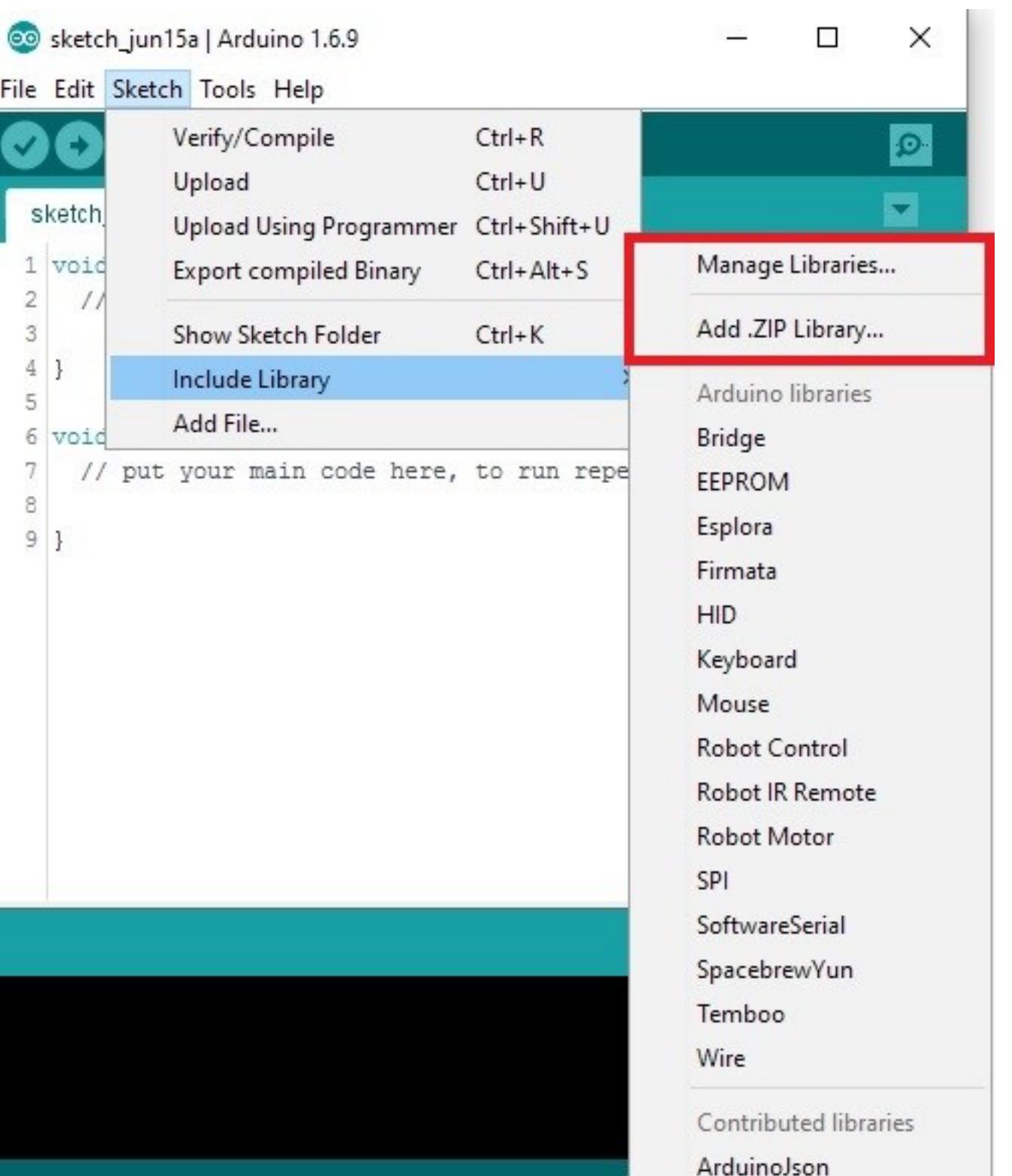


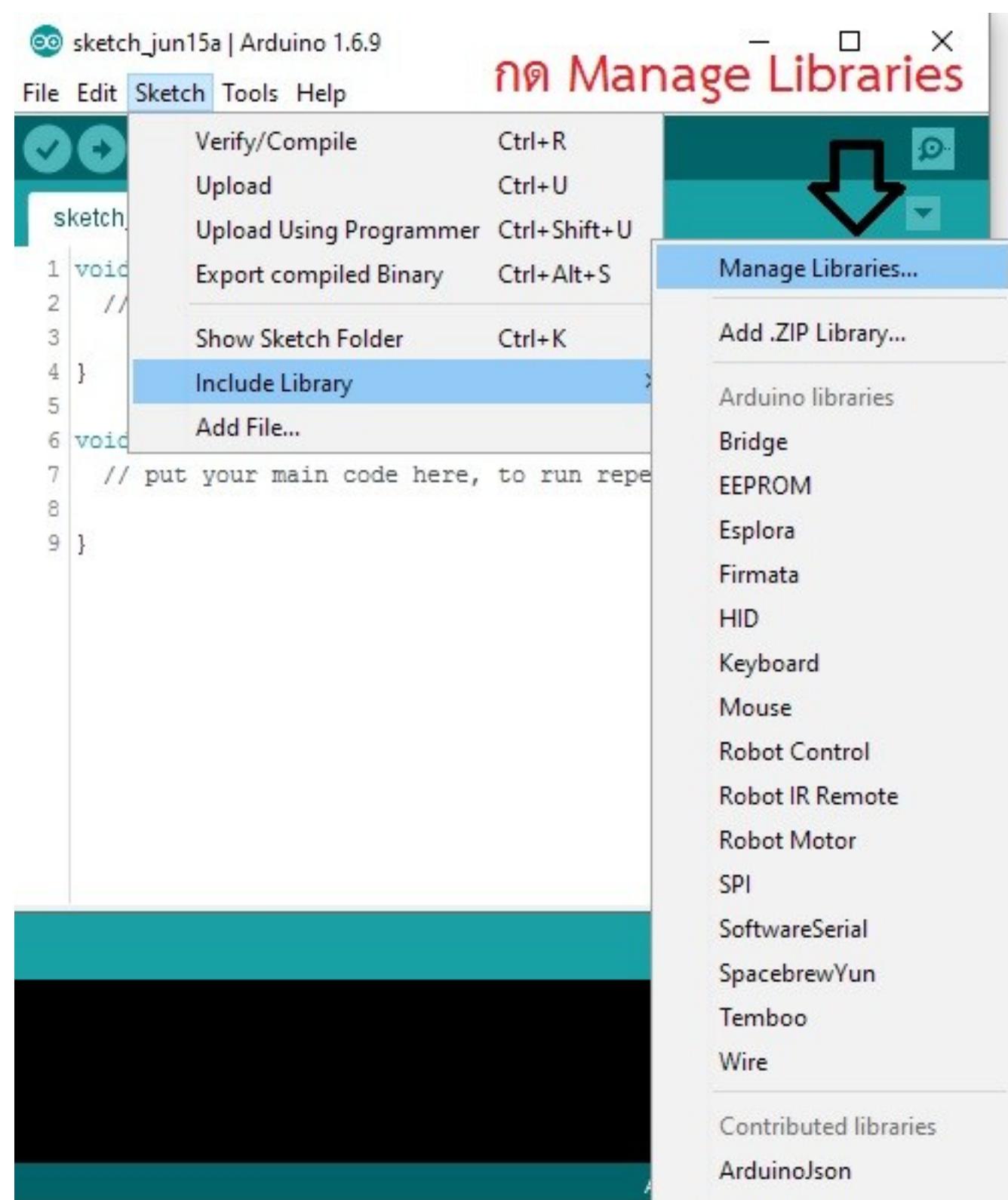


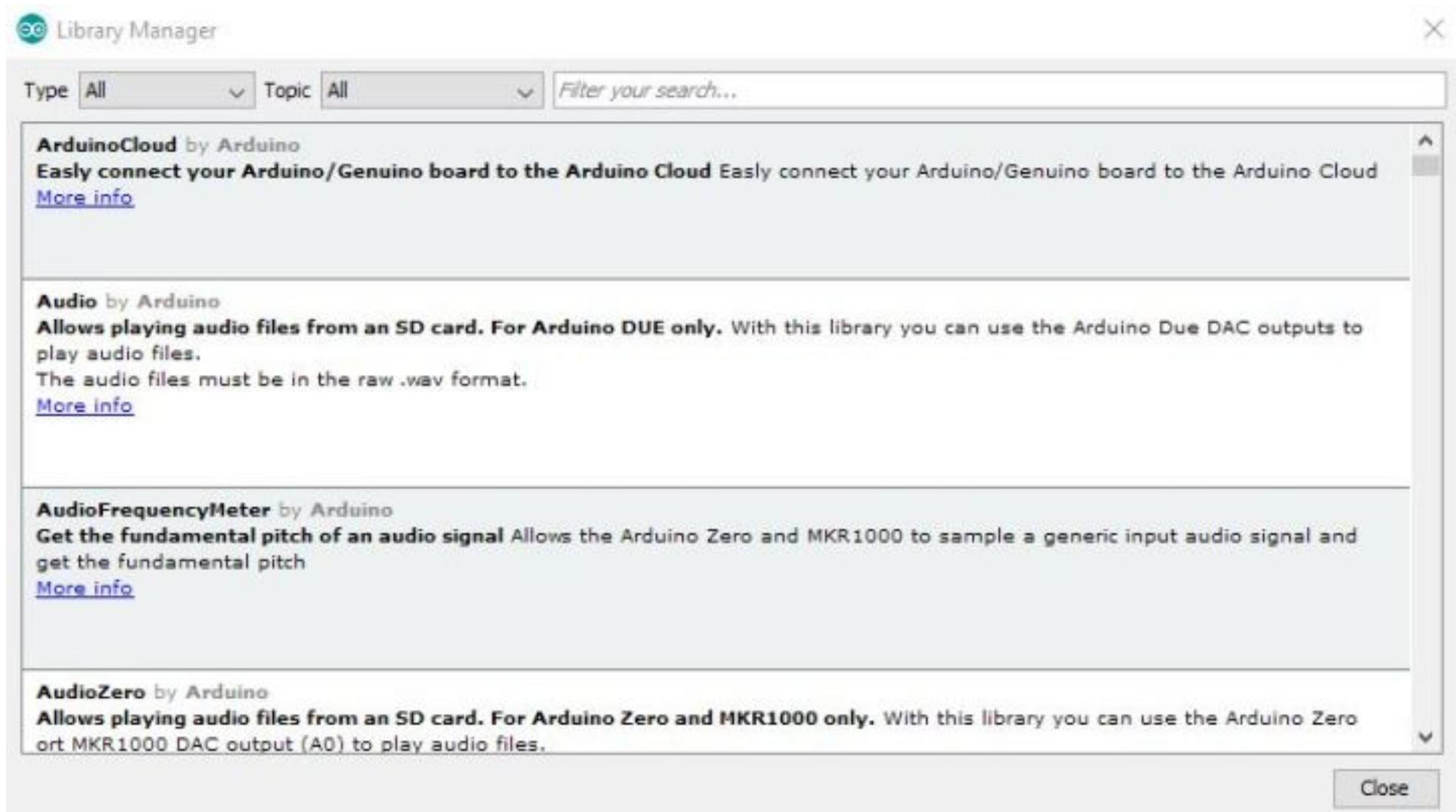


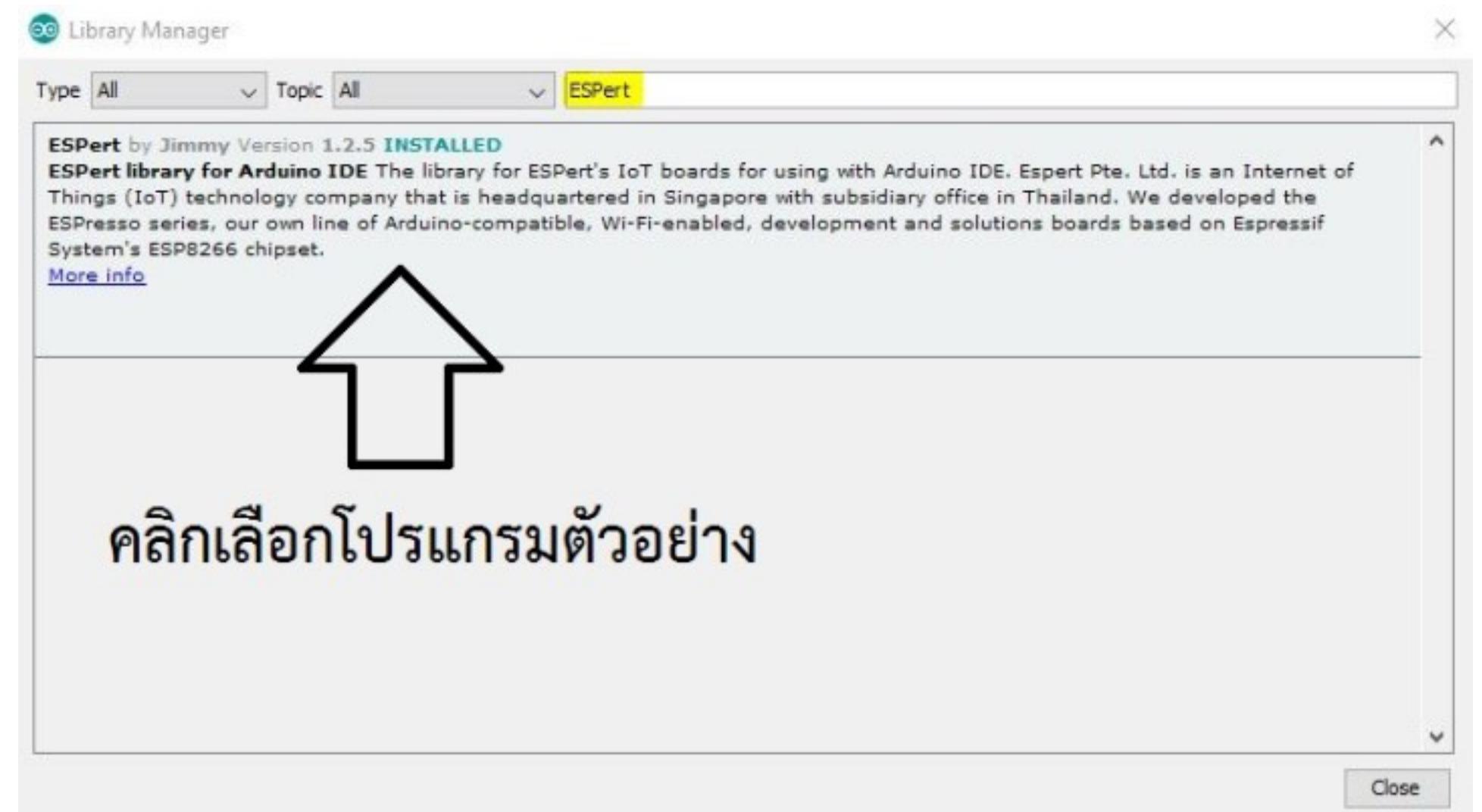


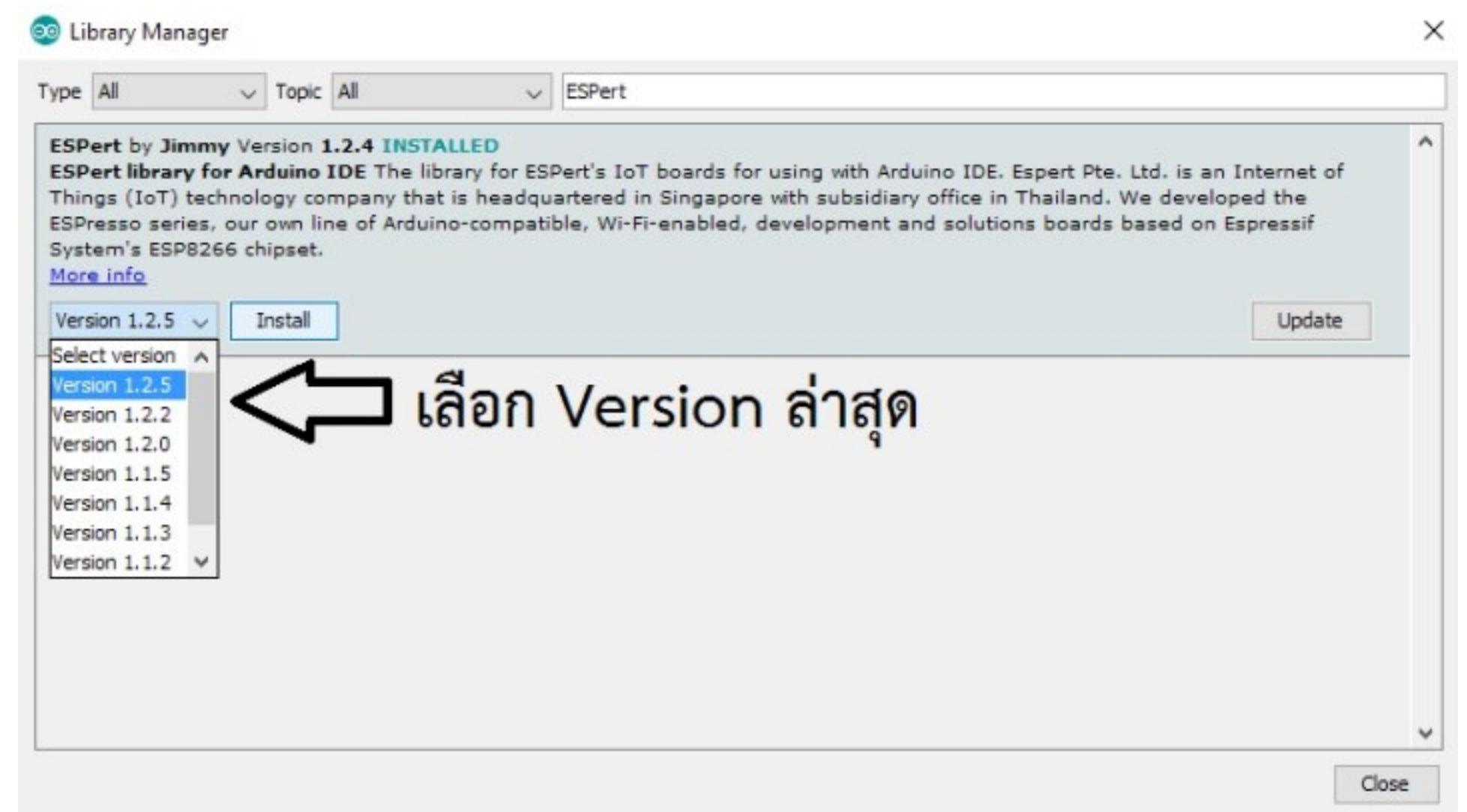
มีบอร์ด ESP8266 เพิ่มเข้ามาแล้ว

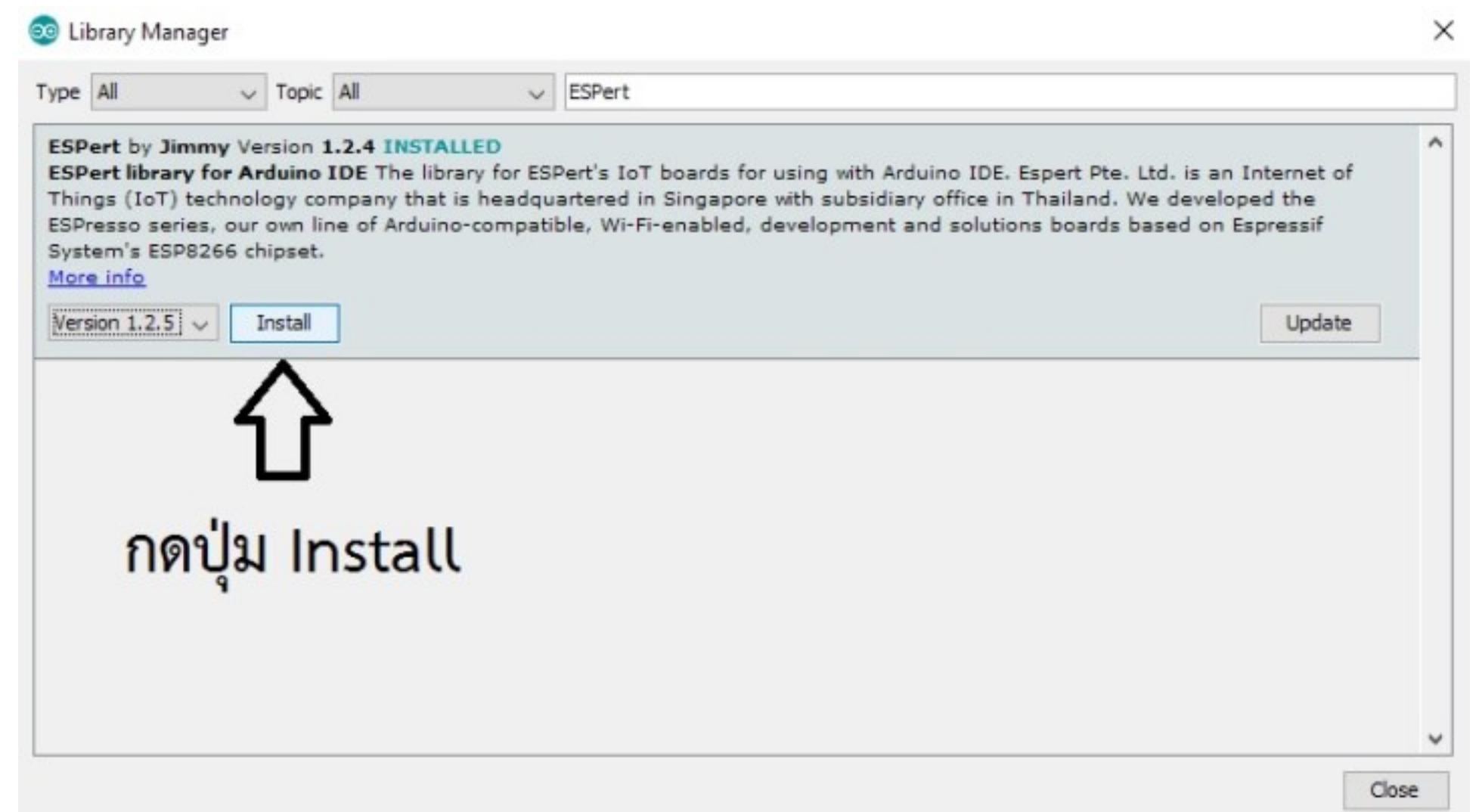












Library Manager

X

Type All

Topic All

ESPert

ESPert by Jimmy Version 1.2.4 INSTALLED

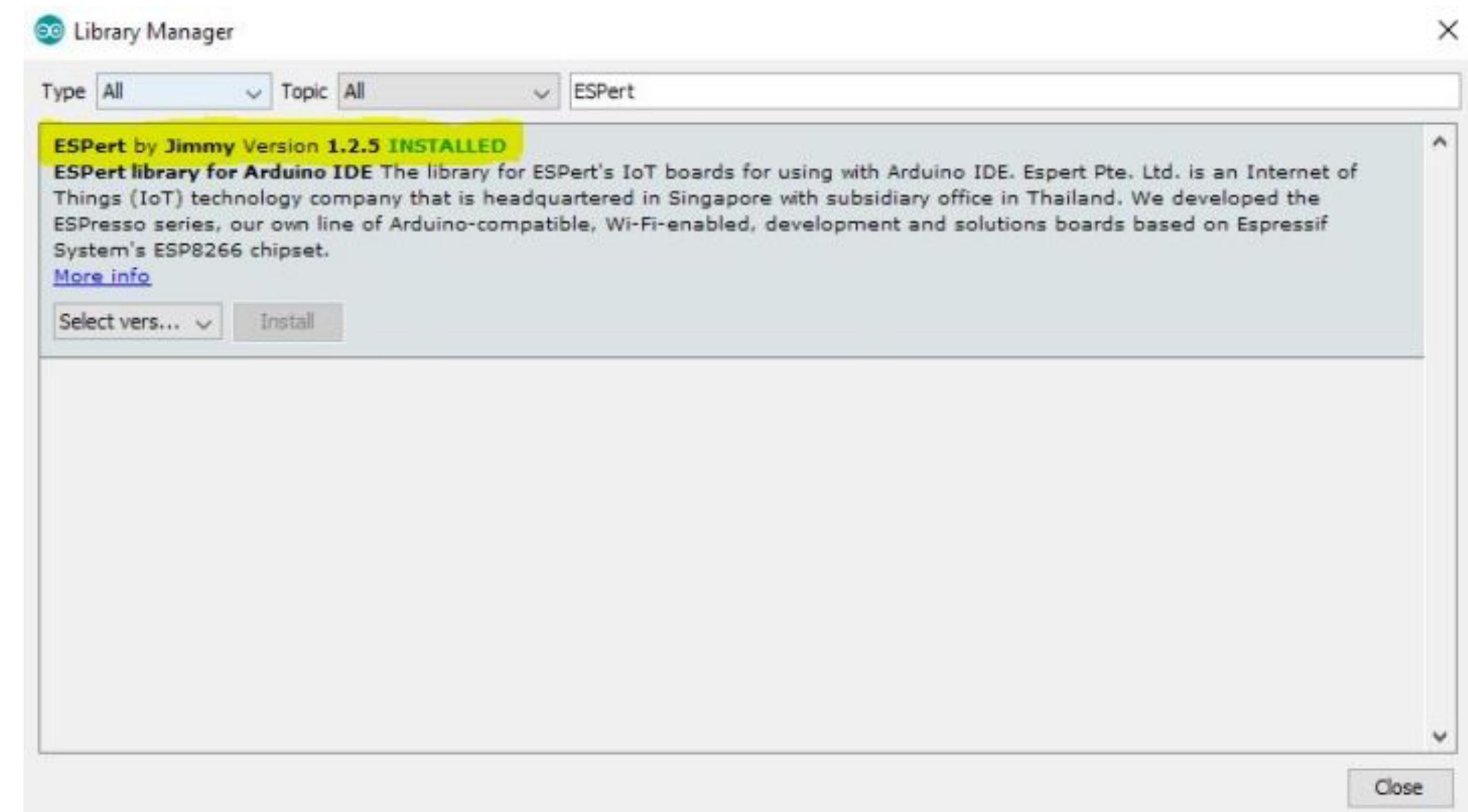
ESPert library for Arduino IDE The library for ESPert's IoT boards for using with Arduino IDE. Espert Pte. Ltd. is an Internet of Things (IoT) technology company that is headquartered in Singapore with subsidiary office in Thailand. We developed the ESPresso series, our own line of Arduino-compatible, Wi-Fi-enabled, development and solutions boards based on Espressif System's ESP8266 chipset.

[More info](#)

Installing...

Downloading library: ESPert Downloaded 292kb of 2,291kb.

Cancel



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 initializes pins and serial communication. It then enters a loop where it waits for connection, prints dots while waiting, and prints connection details once connected.
- Serial Monitor:** Shows the message "Done compiling." followed by the library path: "Using library ESP8266WiFi at version 1.0 in folder /Users/luke/Downloads/packages/ESP8266/". Below that, it displays memory usage: "Sketch uses 225,873 bytes (21%) of program storage space. Maximum is 1,044,464 bytes. Global variables use 31,800 bytes (38%) of dynamic memory, leaving 50,120 bytes for local variables. Memory use is 225,873 bytes." The monitor also shows the port information: "ESPRESSO Lite 2.0, 80 MHz, 921600, 4M (3M SPIFFS), nodemcu, Disabled, None on /dev/cu.usbserial-D00N13E".

STA WebServer

esp8266_webserver | Arduino 1.6.9

```
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 }
```

Done Saving.

JIANG MAI
ESPRESSO LITE 2.0, 80 MHz, 921600, 4M (3M SPIFFS), nodemcu, Disabled, None on /dev/cu.usbserial-D50UNI36



AP WebServer



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

- Title Bar:** esp8266_webserver | Arduino 1.6.9
- Toolbar:** Standard Arduino toolbar with icons for file operations (New, Open, Save, Print, Find, Copy, Paste, Find Next, Find Previous).
- Code Editor:** The main area displays the C++ code for an ESP8266 Webserver. The code includes setup and loop functions for connecting to WiFi, starting a MDNS responder, and handling HTTP requests.

```
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.

Basic HTTP Get

The screenshot shows the Arduino IDE interface with the title bar "esp8266_basic_httpget | Arduino 1.6.9". The main window displays the following C++ code:

```
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 }
```

The status bar at the bottom shows the following messages:

- 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

cmmc.io/docs