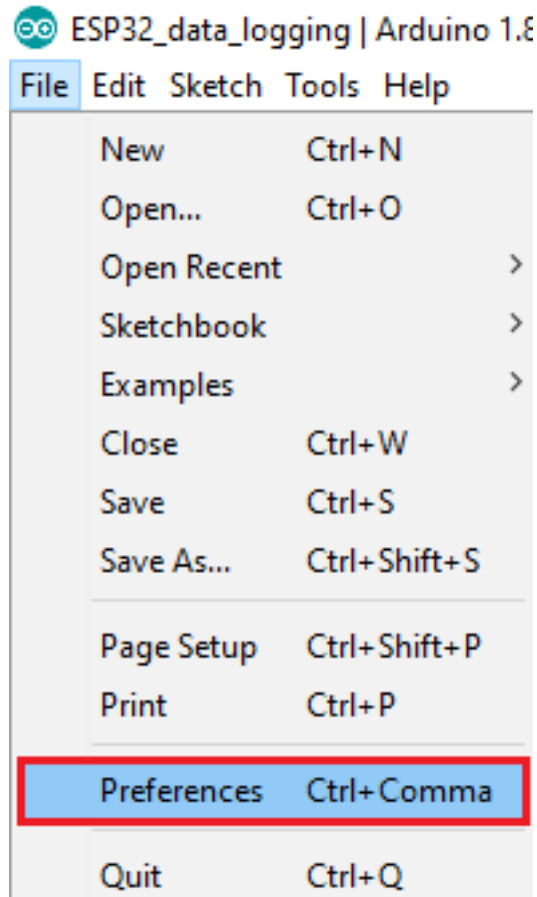


Installing ESP32 Add-on in Arduino IDE

To install the ESP32 board in your Arduino IDE, follow these next instructions:

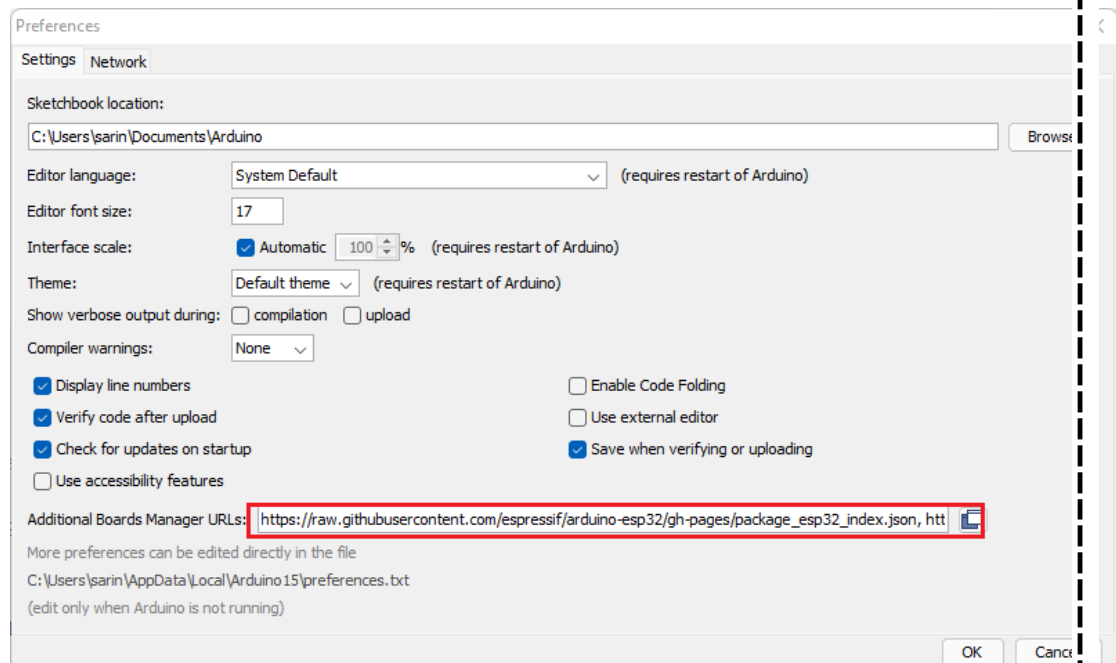
1. In your Arduino IDE, go to **File> Preferences**



2. Enter the following into the “Additional Board Manager URLs” field:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

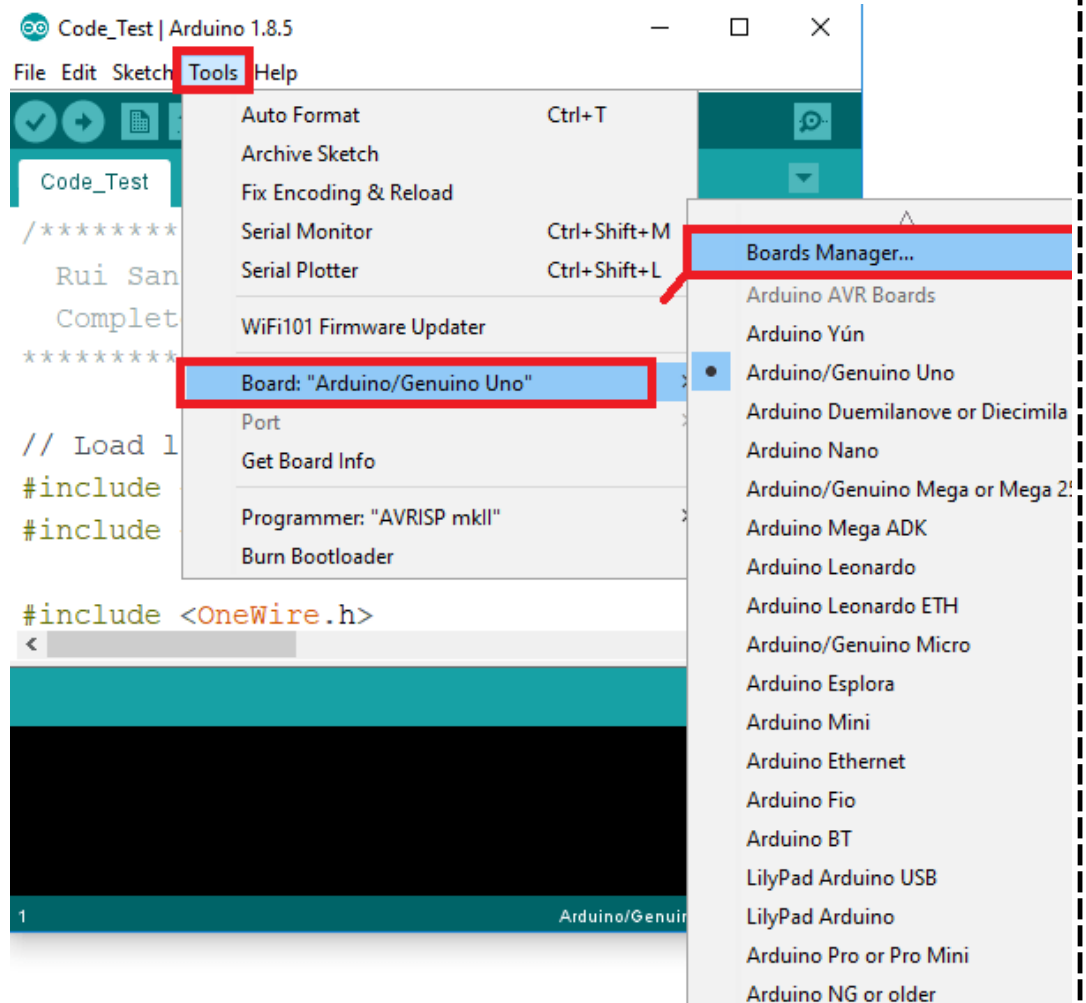
Then, click the “OK” button:



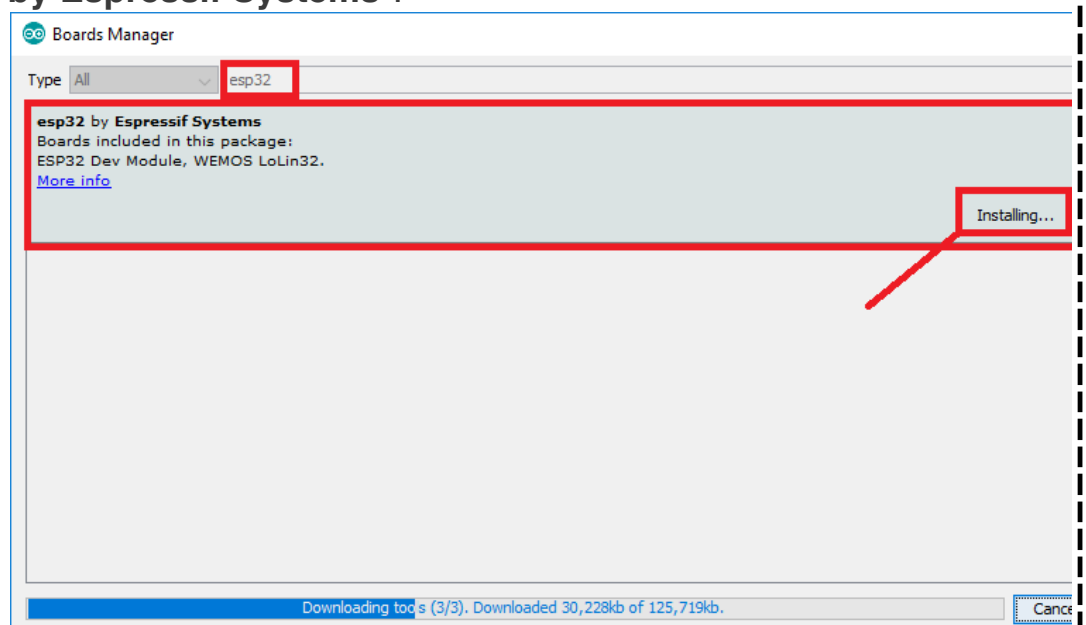
Note: if you already have the ESP8266 boards URL, you can separate the URLs with a comma as follows:

`https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_index.json`

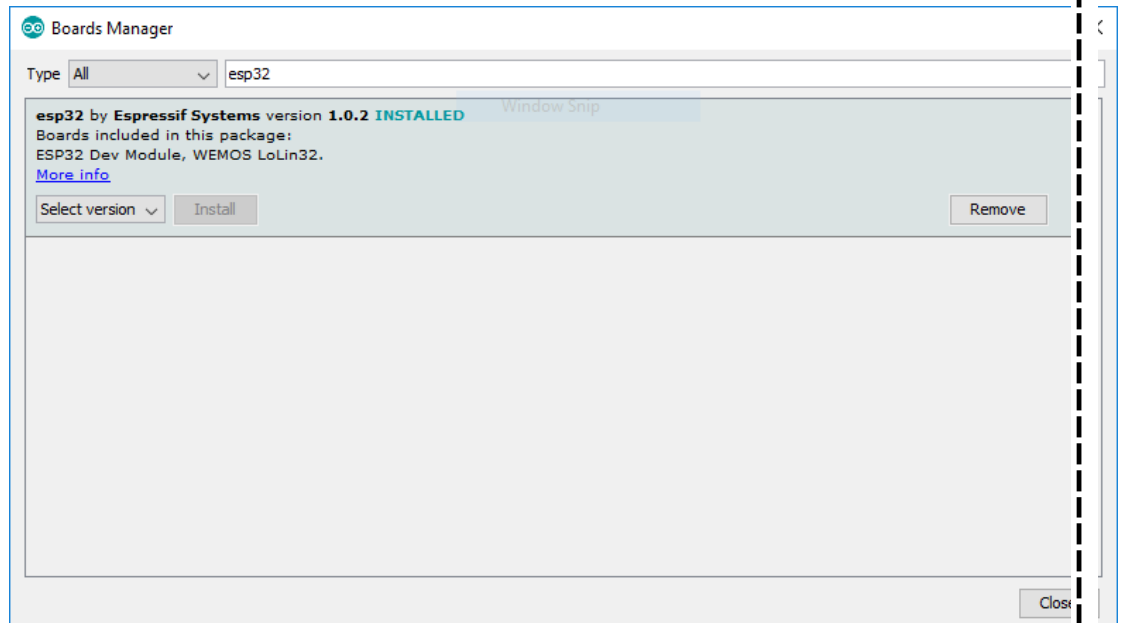
3. Open the Boards Manager. Go to **Tools > Board > Boards Manager...**



4. Search for **ESP32** and press install button for the “**ESP32 by Espressif Systems**”:



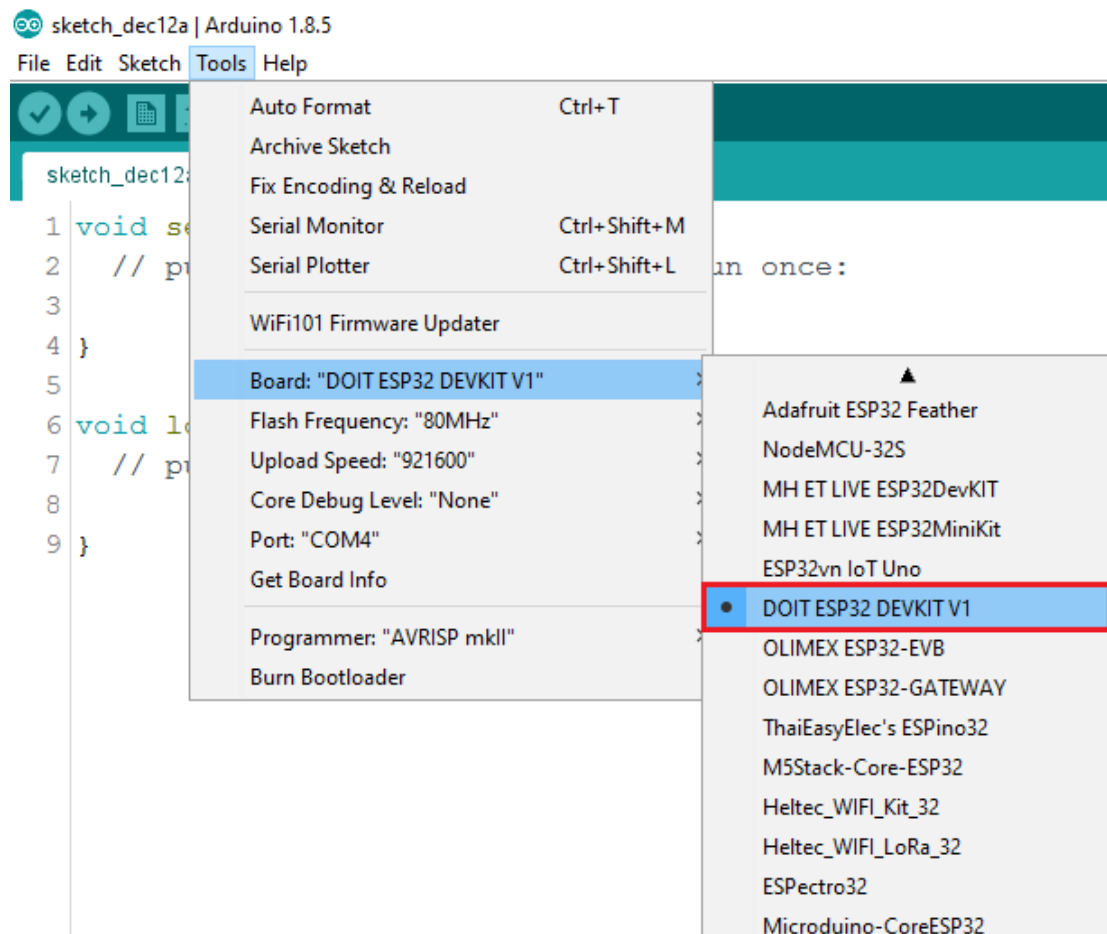
5. That's it. It should be installed after a few seconds.



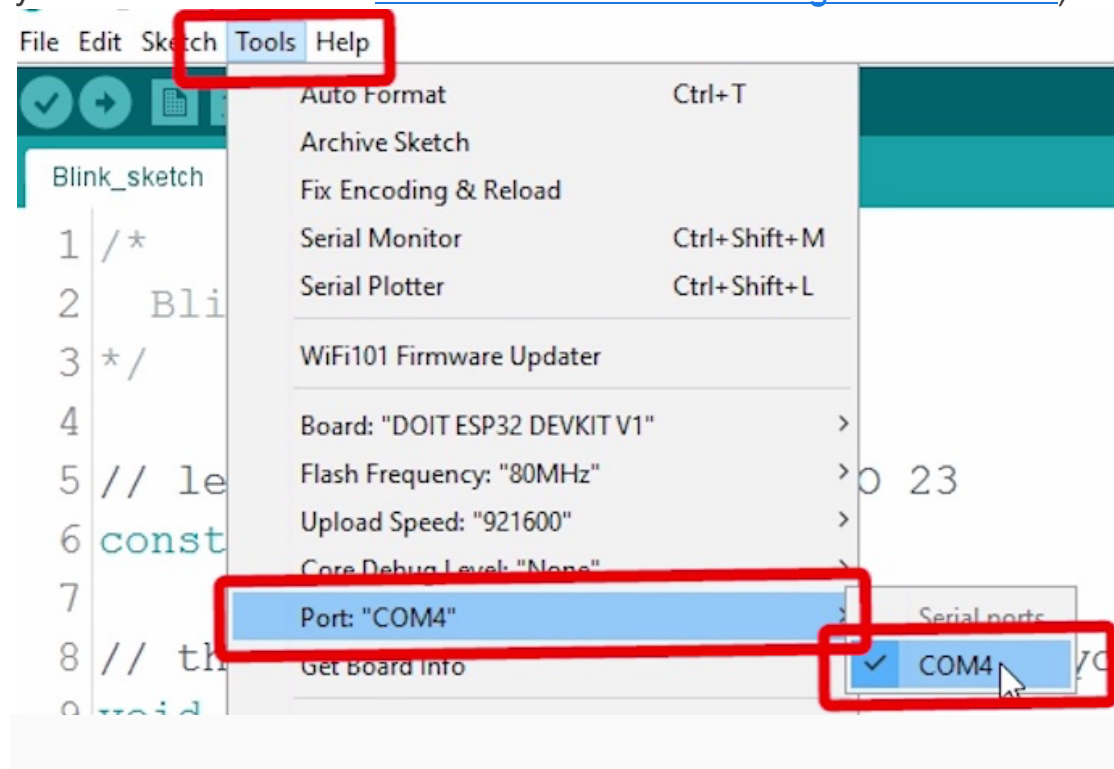
Testing the Installation

Plug the ESP32 board to your computer. With your Arduino IDE open, follow these steps:

1. Select your Board in **Tools > Board** menu (in my case it's the **DOIT ESP32 DEVKIT V1**)

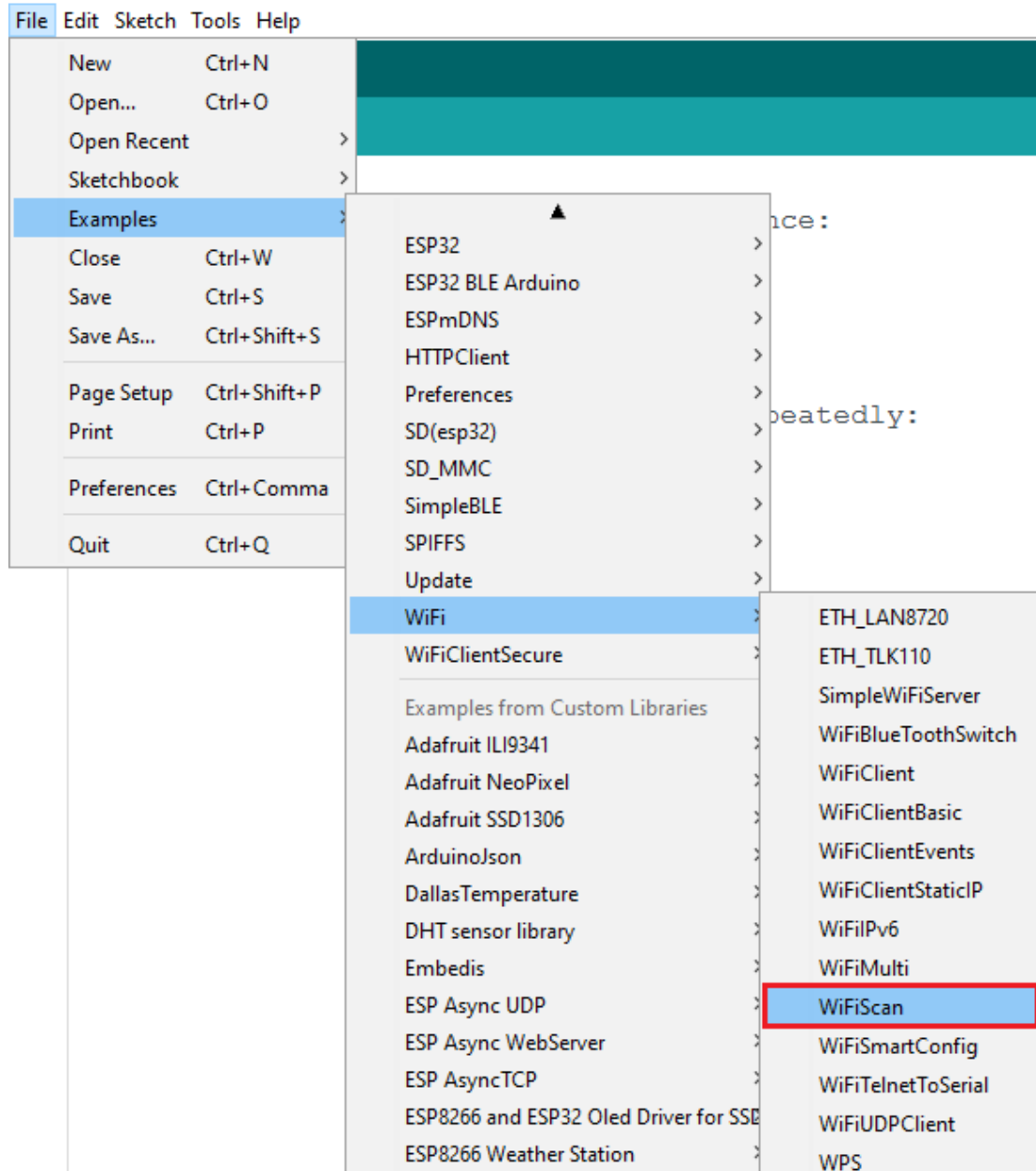


2. Select the Port (if you don't see the COM Port in your Arduino IDE, you need to install the [CP210x USB to UART Bridge VCP Drivers](#)):

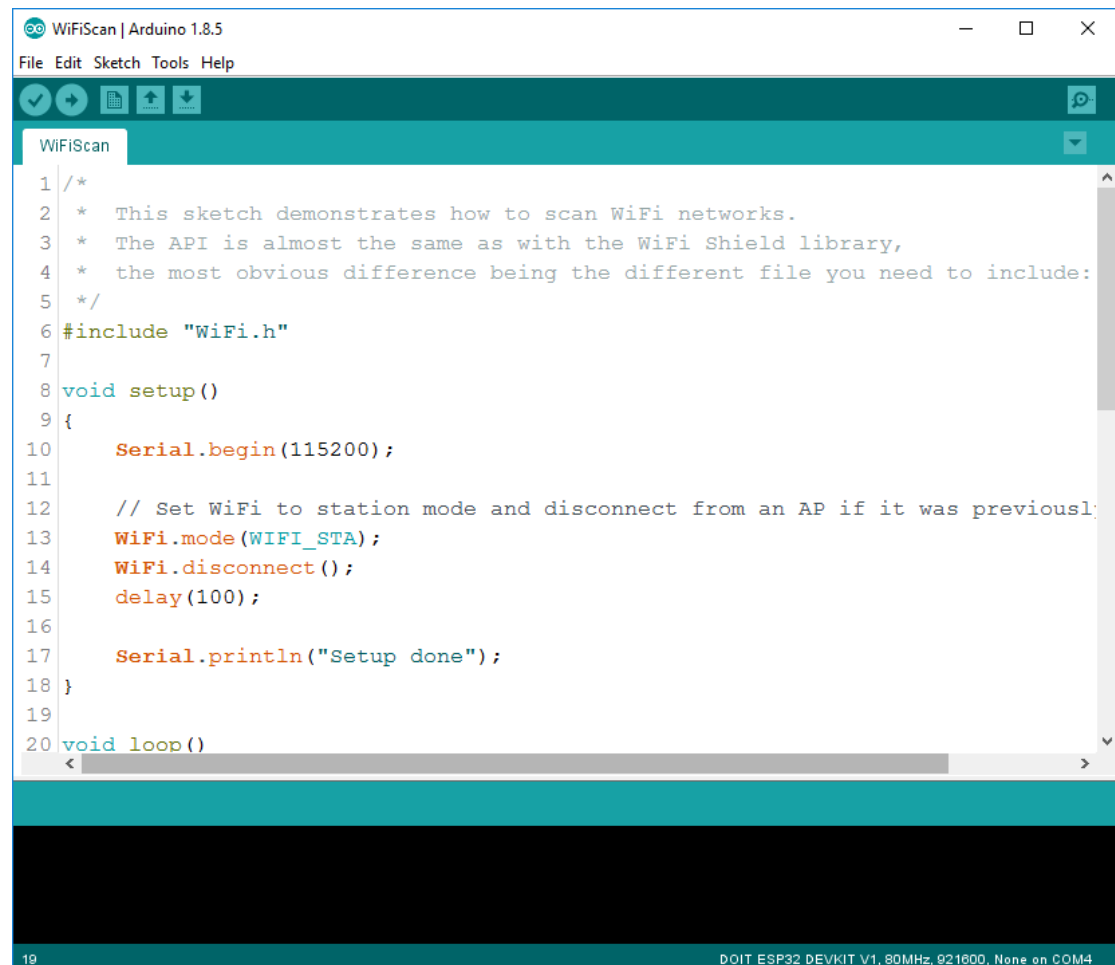


3. Open the following example under **File > Examples > WiFi (ESP32) > WiFiScan**

sketch_dec12a | Arduino 1.8.5



4. A new sketch opens in your Arduino IDE:



The screenshot shows the Arduino IDE interface with a sketch named "WiFiScan". The code is as follows:

```
1 /*
2  * This sketch demonstrates how to scan WiFi networks.
3  * The API is almost the same as with the WiFi Shield library,
4  * the most obvious difference being the different file you need to include:
5  */
6 #include "WiFi.h"
7
8 void setup()
9 {
10     Serial.begin(115200);
11
12     // Set WiFi to station mode and disconnect from an AP if it was previously
13     WiFi.mode(WIFI_STA);
14     WiFi.disconnect();
15     delay(100);
16
17     Serial.println("Setup done");
18 }
19
20 void loop()
```

The status bar at the bottom indicates the board is "DOIT ESP32 DEVKIT V1, 80MHz, 921600, None on COM4".

5. Press the **Upload** button in the Arduino IDE. Wait a few seconds while the code compiles and uploads to your board.



6. If everything went as expected, you should see a “**Done uploading.**” message.

```
Done uploading.
Writing at 0x0004c000... (84 %)
Writing at 0x00050000... (89 %)
Writing at 0x00054000... (94 %)
Writing at 0x00058000... (100 %)
Wrote 481440 bytes (299651 compressed) at 0x00010000 in 4.7 seconds
Hash of data verified.
Compressed 3072 bytes to 122...

Writing at 0x00008000... (100 %)
Wrote 3072 bytes (122 compressed) at 0x00008000 in 0.0 seconds (effective 115200 bps)
Hash of data verified.

Leaving...
Hard resetting...
```

DOIT ESP32 DEVKIT V1, 80MHz, 921600, None on COM4

7. Open the Arduino IDE Serial Monitor at a baud rate of 115200:



8. Press the ESP32 on-board **Enable** button and you should see the networks available near your ESP32:

```
COM4
|
scan done
2 networks found
1: MEO-620B4B (-49) *
2: MEO-WiFi (-50)

scan start
scan done
2 networks found
1: MEO-620B4B (-48) *
2: MEO-WiFi (-49)
```

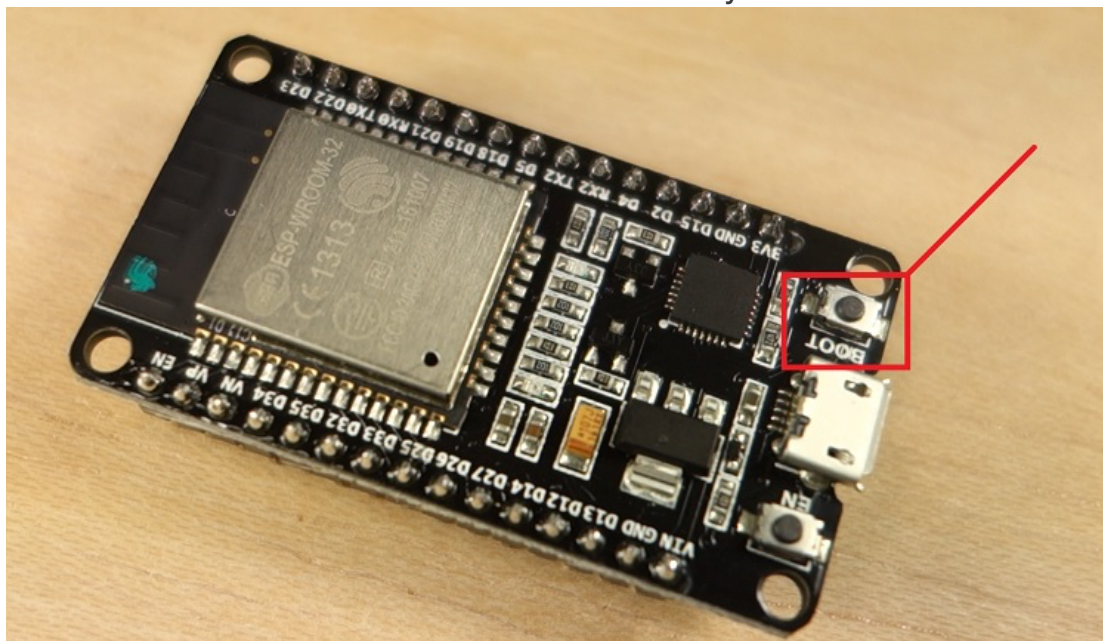
☒ Autoscroll Both NL & CR 115200 baud Clear output

Troubleshooting

If you try to upload a new sketch to your ESP32 and you get this error message “A fatal error occurred: Failed to connect to ESP32: Timed out... Connecting...”. It means that your ESP32 is not in flashing/uploading mode.

Having the right board name and COM port selected, follow these steps:

- Hold-down the “**BOOT**” button in your ESP32 board



- Press the “**Upload**” button in the Arduino IDE to upload your sketch:



- After you see the “**Connecting....**” message in your Arduino IDE, release the finger from the “**BOOT**” button:

```
Uploading...
Archiving built core (caching) in: C:\Users\ROISAN-1\AppData\Local\Temp\arduino_cache_959883\core\core_esp8266_esp32_esp32doit-devkit-v1_Flash
Sketch uses 501366 bytes (39%) of program storage space. Maximum is 1310720 bytes.
Global variables use 37320 bytes (12%) of dynamic memory, leaving 257592 bytes for local variables. Maximum is 254912 bytes.
esptool.py v2.1
Connecting.....
Chip is ESP32D0WDQ6 (revision (unknown 0xa))
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 8192 bytes to 47...

Writing at 0x0000e000... (100 %)
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.0 seconds (effective 8192.1 kbit/s)...
Hash of data verified.
Compressed 12304 bytes to 8126...

Writing at 0x00001000... (100 %)
```

- After that, you should see the “**Done uploading**” message

That’s it. Your ESP32 should have the new sketch running. Press the “**ENABLE**” button to restart the ESP32 and run the new uploaded sketch.

You’ll also have to repeat that button sequence every time you want to upload a new sketch. But if you want to solve this issue once for all without the need to press the **BOOT** button, follow the suggestions in the next guide:

- [\[SOLVED\] Failed to connect to ESP32: Timed out waiting for packet header](#)

If you experience any problems or issues with your ESP32, take a look at our in-depth [ESP32 Troubleshooting Guide](#).