**ARDUINO IDE**

**SOFTWARE**

**ARDUINO:**

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board. Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

**The key features are −**

* Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
* You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
* Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
* Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
* Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

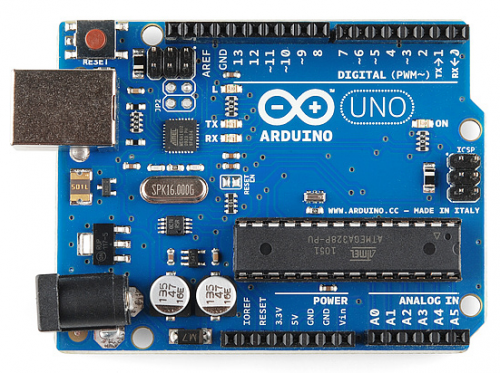
**PREREQUISITES:**

Before you start proceeding with this tutorial, we assume that you are already familiar with the basics of C. If you are not well aware of these concepts, then we will suggest you go through our short tutorials on C. A basic understanding of microcontrollers and electronics is also expected.

**INSTALLATION AND CONFIGURATION:**

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

**1**. First you must have your Arduino board and a USB cable. In case you use Arduino UNO or esp32, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.



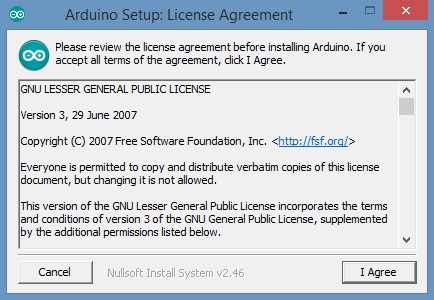


**2.** Install Arduino IDE -windows - IDE Version provided in Software Section of DVD or Visit <http://www.arduino.cc/en/main/software> to download the latest Arduino IDE version for your computer’s operating system. There are versions for Windows, Mac, and Linux systems. In the download page, click on “Windows Installer” option for the easiest installation.

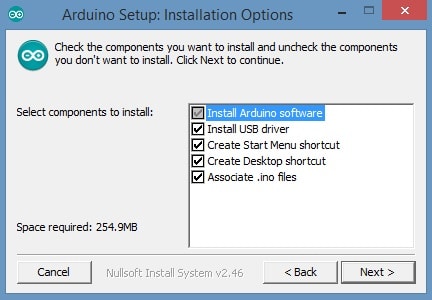
3. Save the .exe file to your hard drive

4. Open the .exe file

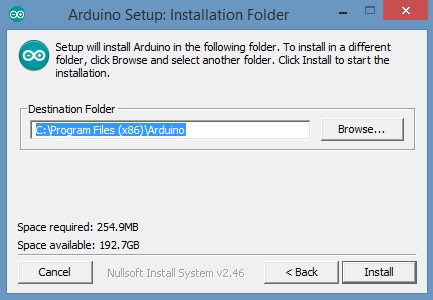
5. Click the button to agree to the licensing agreement:



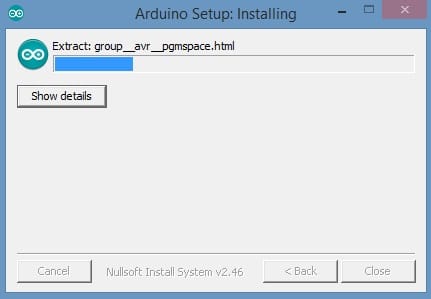
5. Decide which components to install, and then click “Next”:



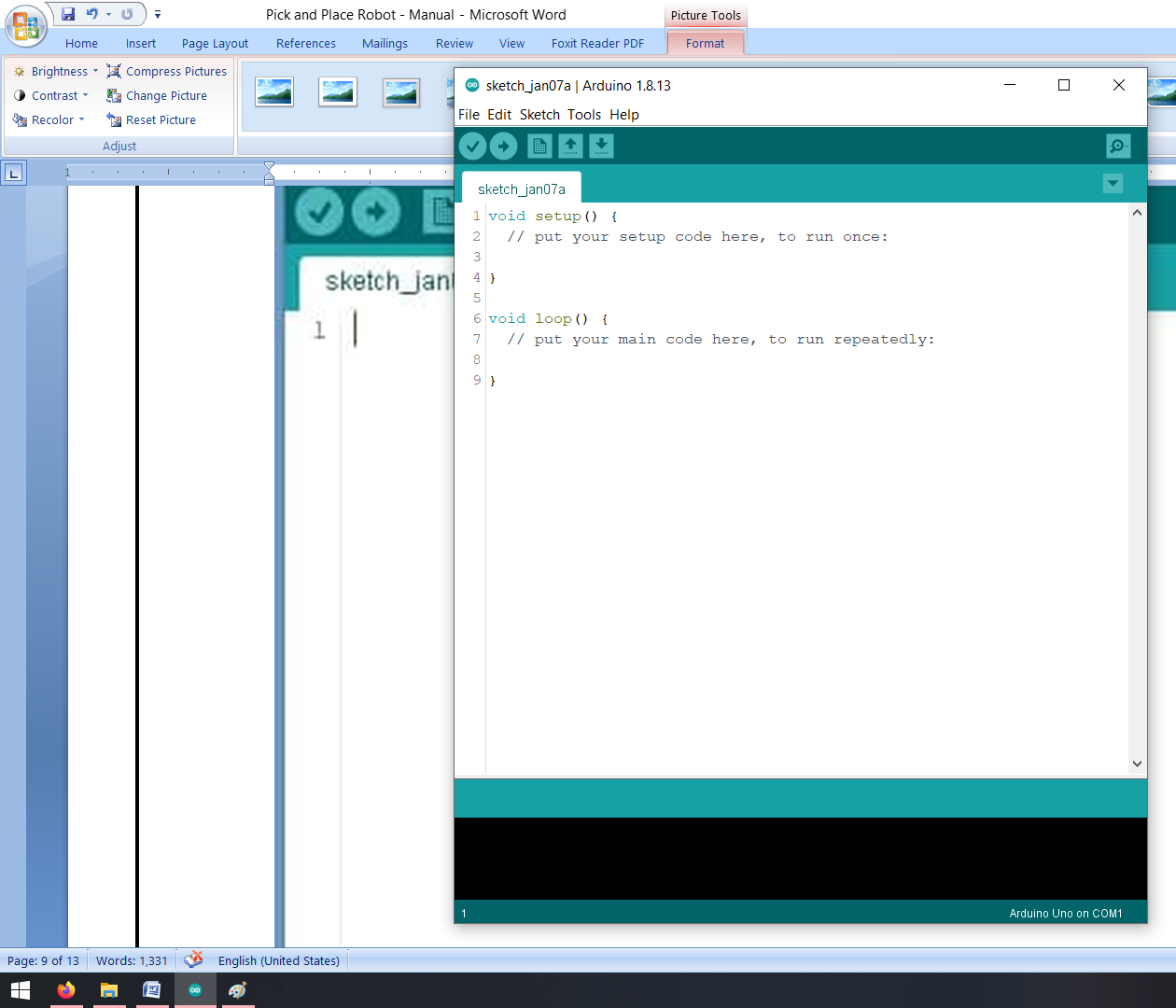
6. Select which folder to install the program to, then click “Install”:



7. Wait for the program to finish installing, and then click “Close”:



8. Find the Arduino shortcut on your Desktop and click on it. The IDE will open up and you’ll see the code editor:

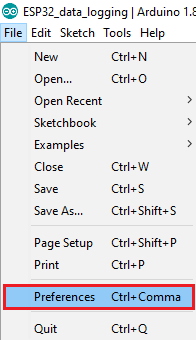


# Installing the ESP32 Board in Arduino IDE

There’s an add-on for the Arduino IDE that allows you to program the ESP32 using the Arduino IDE and its programming language. In this tutorial we’ll show you how to install the ESP32 board in Arduino IDE whether you’re using Windows, Mac OS X or Linux.

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

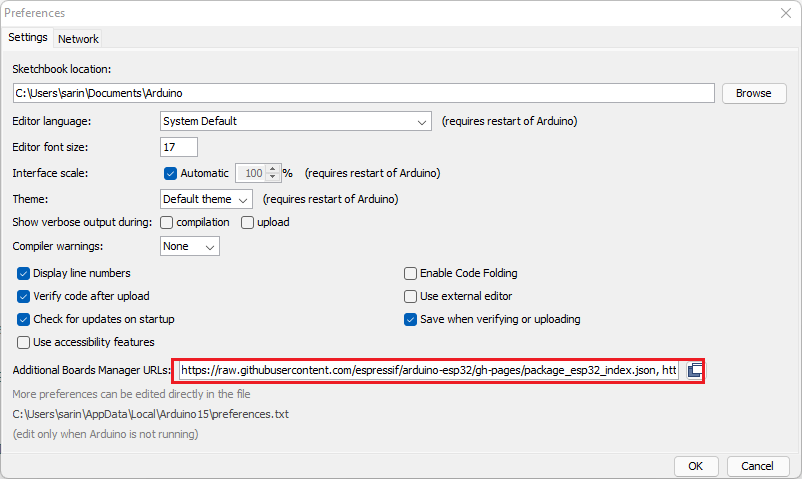
1. In your Arduino IDE, go to **File**> **Preferences**



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

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\_esp32\_index.json

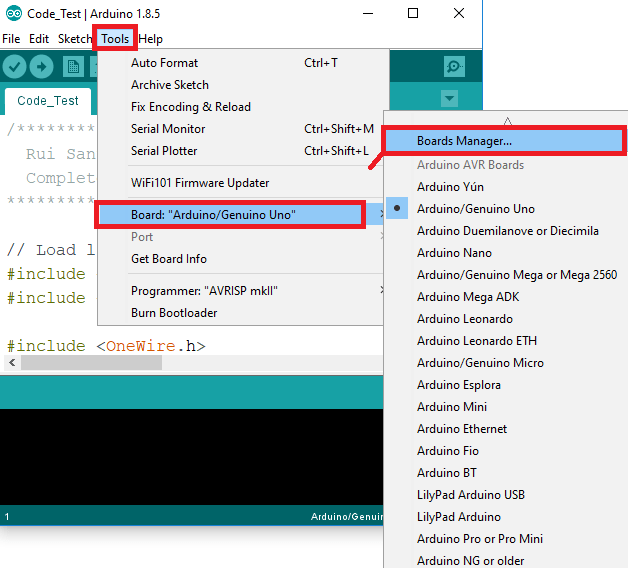
Then, click the “OK” button:



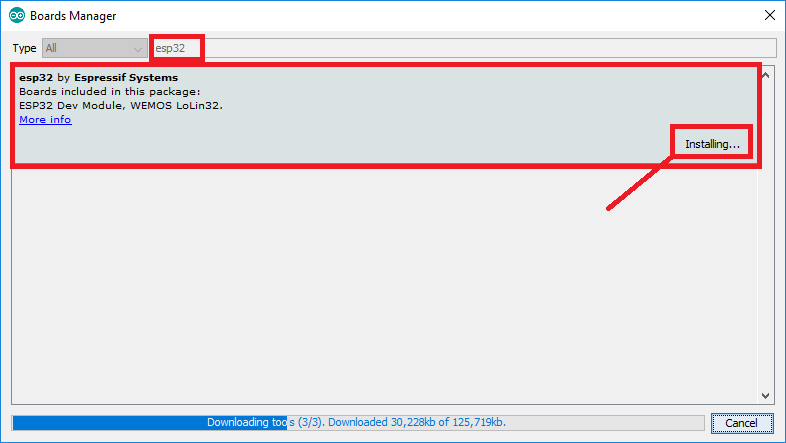
**Note:** if you already have the ESP8266 boards URL, you can separate the URLs with a comma as follows:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\_esp32\_index.json, http://arduino.esp8266.com/stable/package\_esp8266com\_index.json

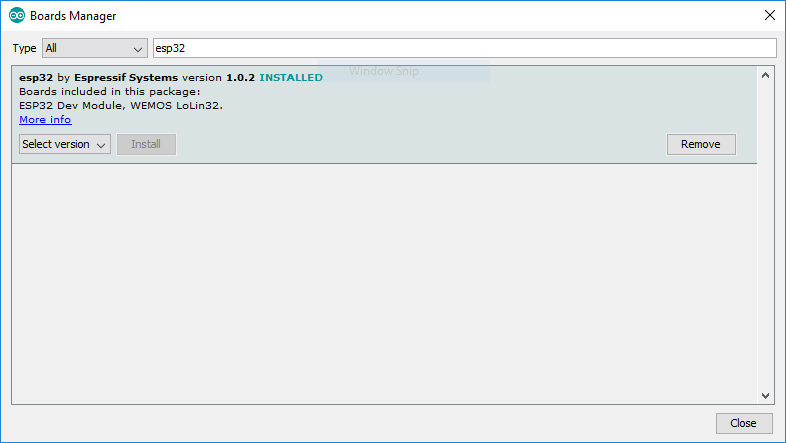
1. Open the Boards Manager. Go to **Tools** > **Board** > **Boards Manager…**



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



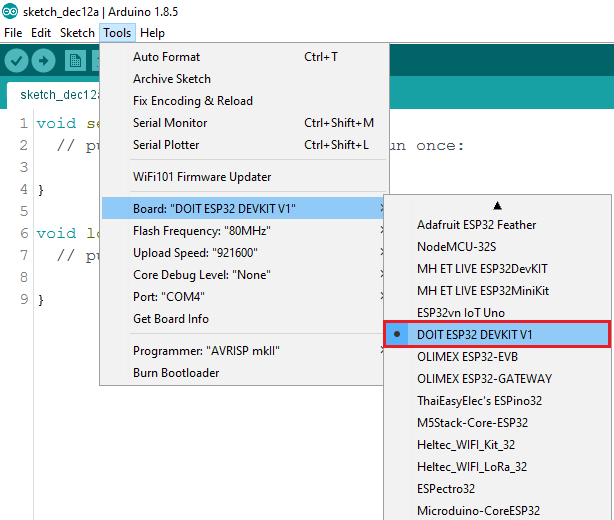
1. That’s it. It should be installed after a few seconds.



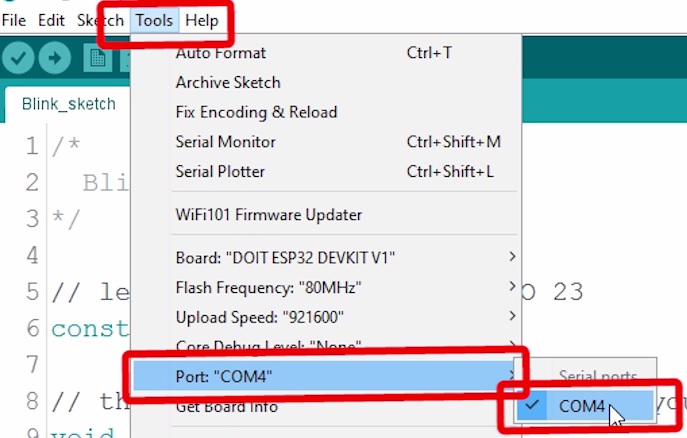
## Testing the Installation

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

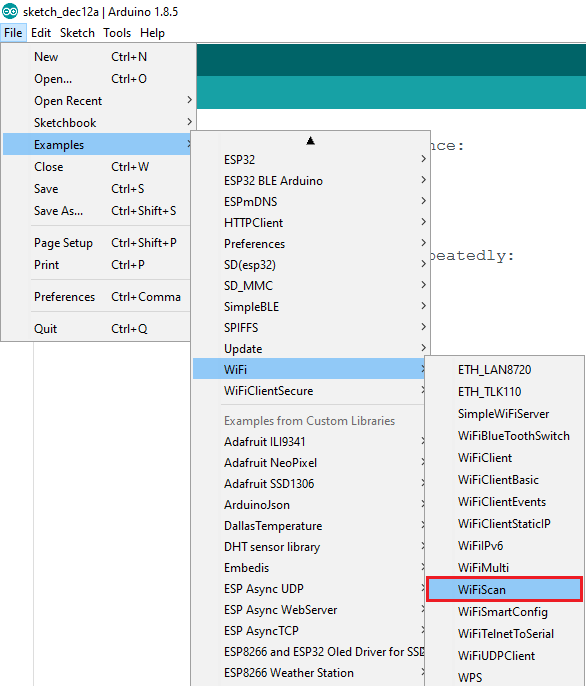
1. Select your Board in **Tools** > **Board** menu (in my case it’s the **DOIT ESP32 DEVKIT V1**)



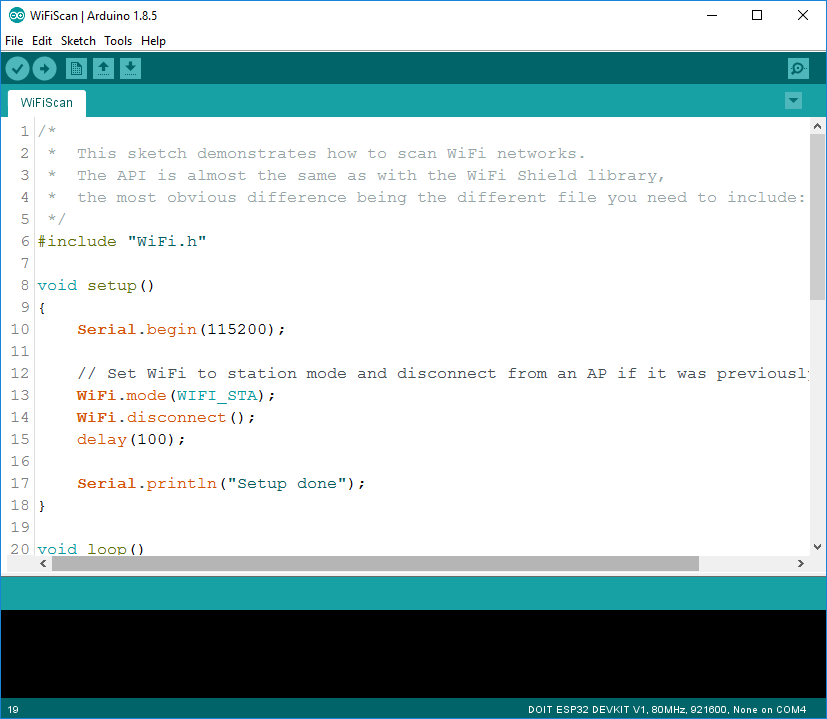
2. Select the Port (if you don’t see the COM Port in your Arduino IDE, you need to install the [CP210x USB to UART Bridge VCP Drivers](https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers)):



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



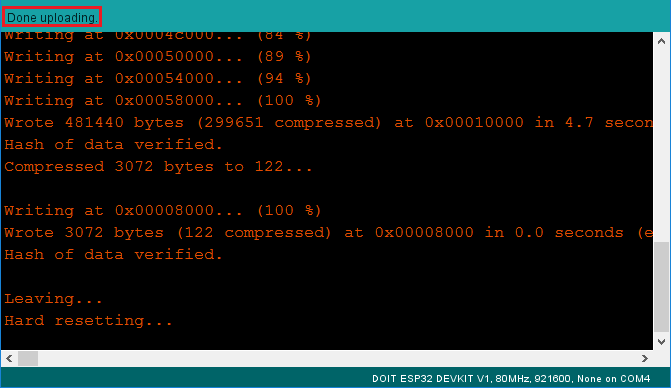
4. A new sketch opens in your Arduino IDE:



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

Arduino IDE upload WiFiScan sketch to ESP32

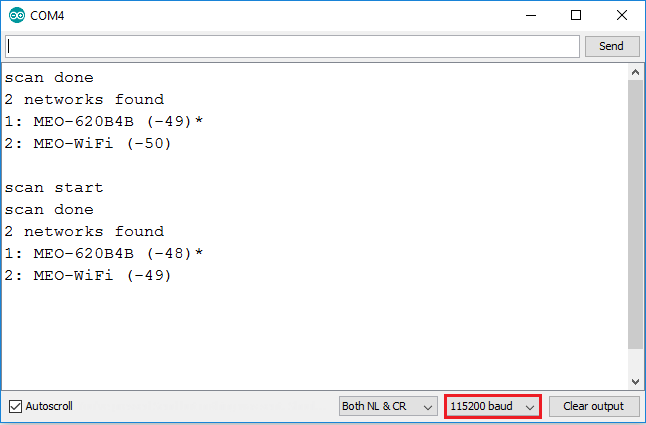
6. If everything went as expected, you should see a “**Done uploading.**” message.



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

Open Arduino IDE Serial Monitor at baud rate 115200

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

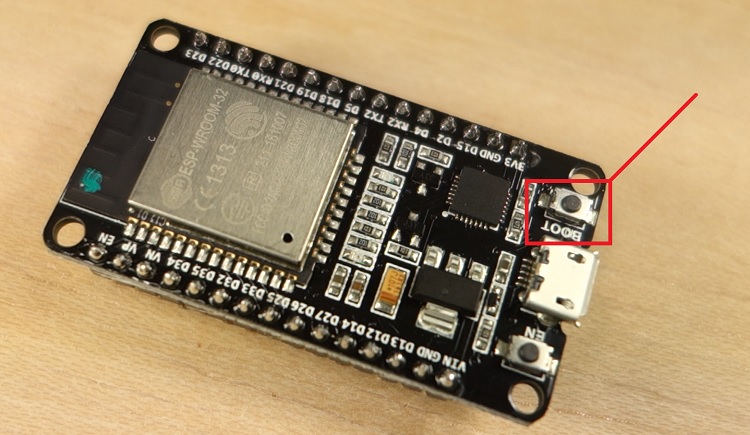


## Troubleshooting

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 por selected, follow these steps:

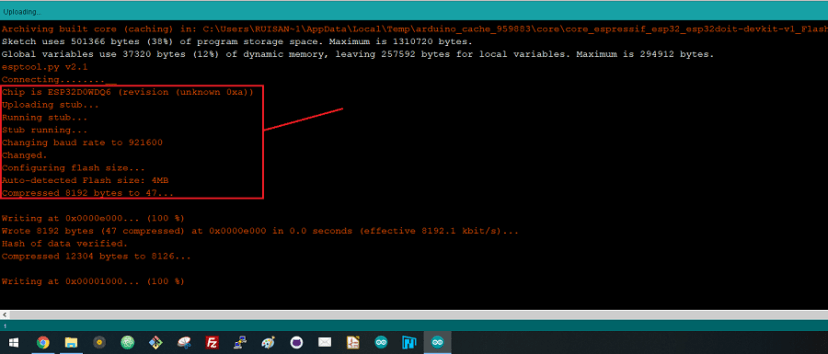
* Hold-down the “**BOOT**” button in your ESP32 board



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

Arduino IDE uploading new sketch to ESP32

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



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

**Downloading Thing Speak library:**

You need to download Thing Speak library for projects using Node-MCU and generic ESP8266.

<https://github.com/mathworks/thingspeak-arduino>

DHT- Sensor Library:

- DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library

- Adafruit Unified Sensor Lib: <https://github.com/adafruit/Adafruit_Sensor>

