

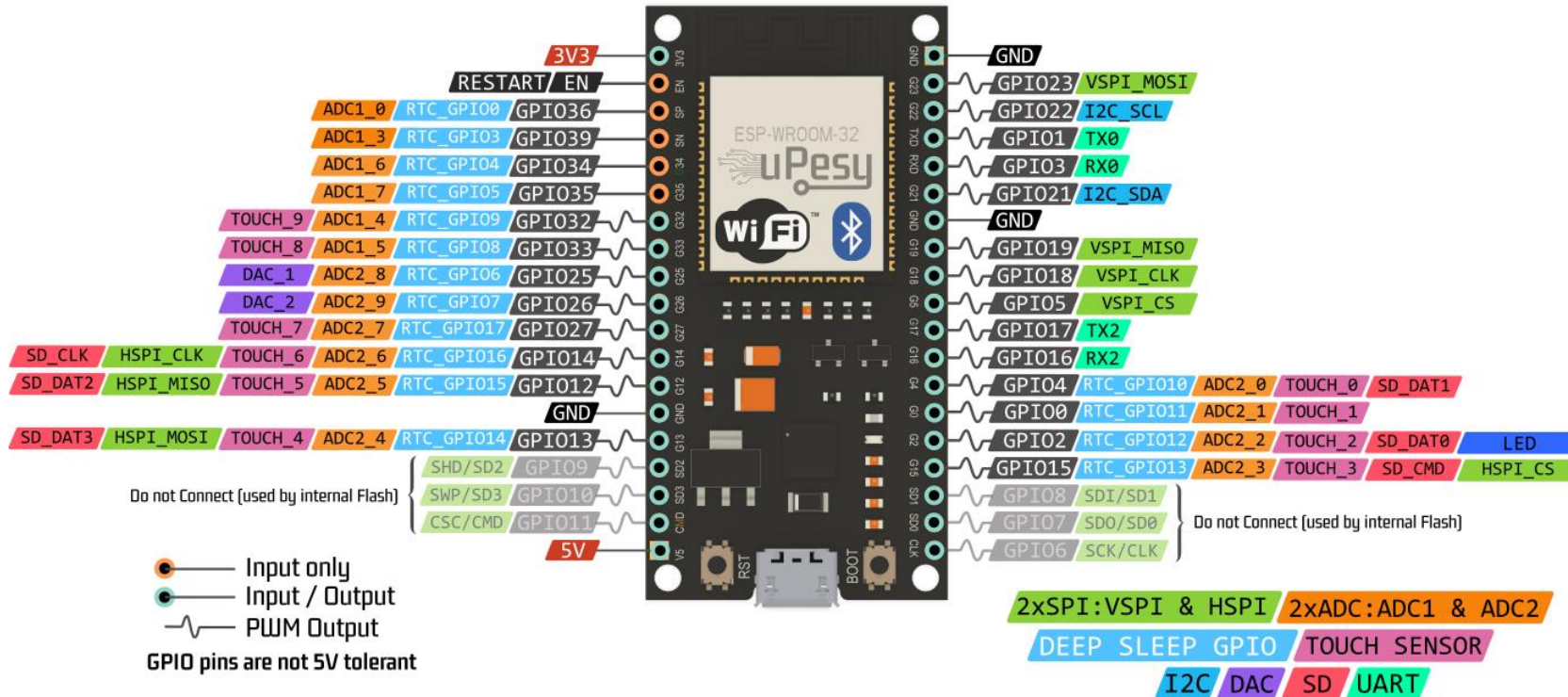


# ESP 32

Done By: Basayl Suliman Ali



# ESP32 Wroom DevKit Full Pinout



- ESP 32 board is considered an introduction to IOT world , for containing a Wi-Fi 2.4g and Bluetooth 4.2 and for a good price .

## Materials Required:

- ESP32 Module
- Arduino IDE
- Programming cable (micro USB cable)

## Hardware Information of ESP32:

**Micro-USB jack:** The micro USB jack is used to connect the ESP32 to our computer through a USB cable. It is used to program the ESP module as well as can be used for serial debugging as it supports serial communication

**EN Button:** The EN button is the reset button of the ESP module. Pressing this button will reset the code running on the ESP module

**Boot Button:** This button is used to upload the Program from Arduino to the ESP module. It has to be pressed after clicking on the upload icon on the Arduino IDE. When the Boot button is pressed along with the EN button, ESP enters into firmware uploading mode. Do not play with this mode unless you know what you are doing.

**Red LED:** The Red LED on the board is used to indicate the power supply. It glows red when the board is powered.

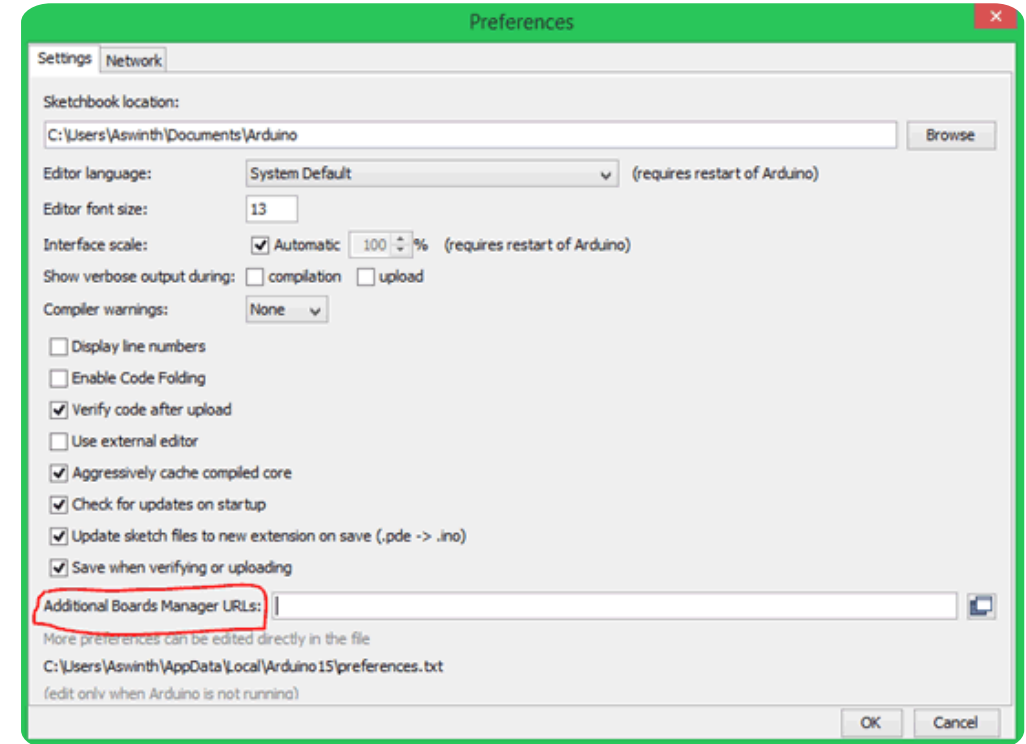
**Blue LED:** The Blue LED on the board is connected to the GPIO pin. It can be turned on or off through programming. In some Chinese cloned boards like mine, this led might also be in red colour.

**I/O pins:** This is where major development has taken place. Unlike ESP8266, on ESP32 we can access all the I/O pin of the module through the break-out pins. These pins are capable of Digital Read/Write, Analog Read/Write, PWM, IIC, SPI, DAC and much more. We will get more into that later. But if you are interested you can learn through the pin description at [ESP32 Datasheet](#).

**ESP-WROOM-32:** This is the heart of the ESP32 module. It is a 32-bit microprocessor developed by Espressif systems. If you are more of a technical person you can read through the [ESP-WROOM-32 Datasheet](#). I have also listed few important parameters below.

• **STEP 1:** Now, let's get started. The first step would be to **download and install the Arduino IDE**. This can be done easily by following the link <https://www.arduino.cc/en/Main/Software> and downloading the IDE for free. If you already have one make sure it is of the latest version.

• **STEP 2:** Once installed, open the Arduino IDE and go to *Files -> Preferences* to open the preferences window and locate the *"Additional Boards Manager URLs:"* as shown



**STEP 3:** This text box might be empty or might also contain some other URL if you have used it previously for ESP8266. If it is empty simply paste the below URL into the text box

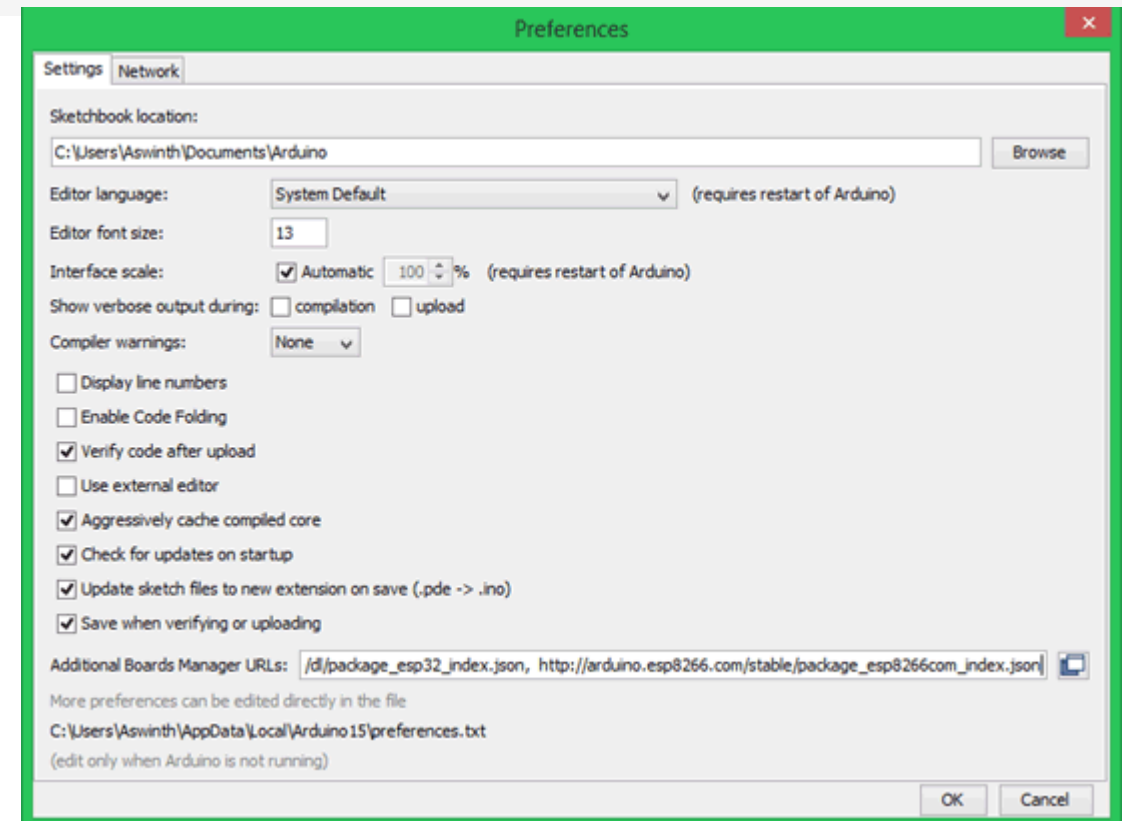
[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

If the text box already contains some other URL just add this URL to it, separate both with a comma (,). Mine already had the ESP8266 URL I just added this URL to and added a comma, like this

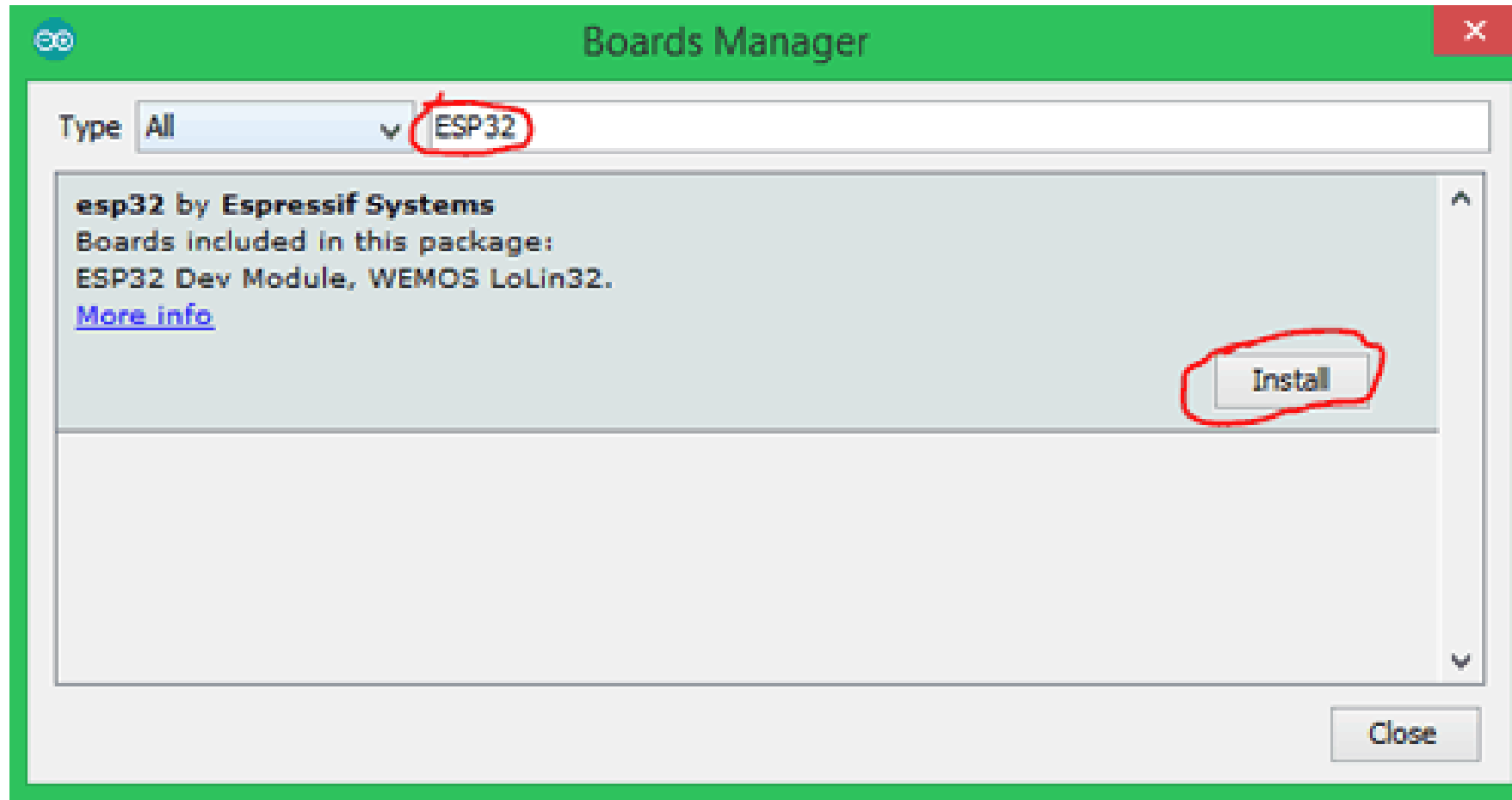
[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json),

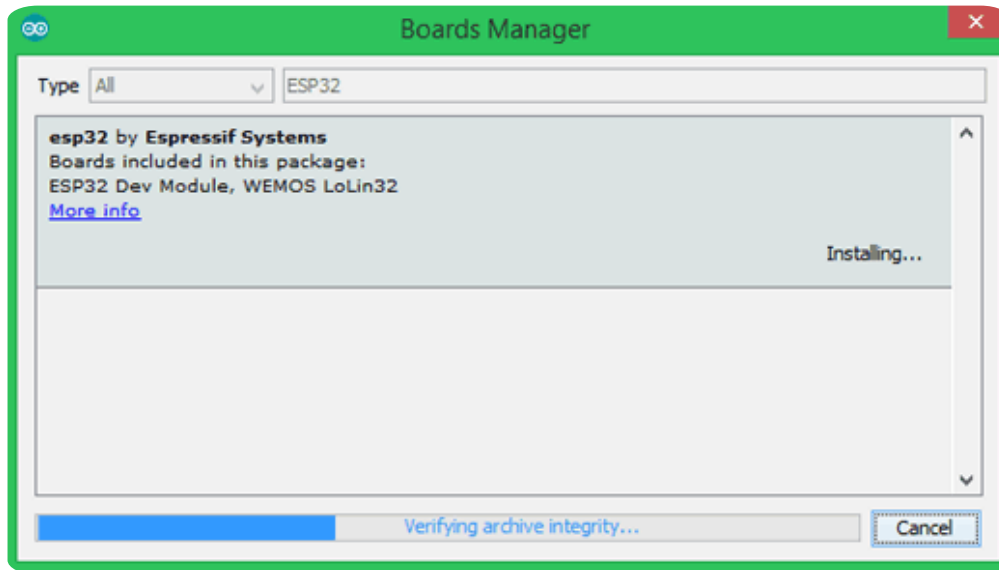
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

Once done, my preferences windows looked like this below. Just click on OK and the window will Disappear.



**STEP 4:** Now go to *Tools -> Boards -> Board Managers* to open the Board manager window and search for ESP32. If the URL was pasted correctly your window should find the below screen with *Install* button, just click on the Install button and your board should get installed.

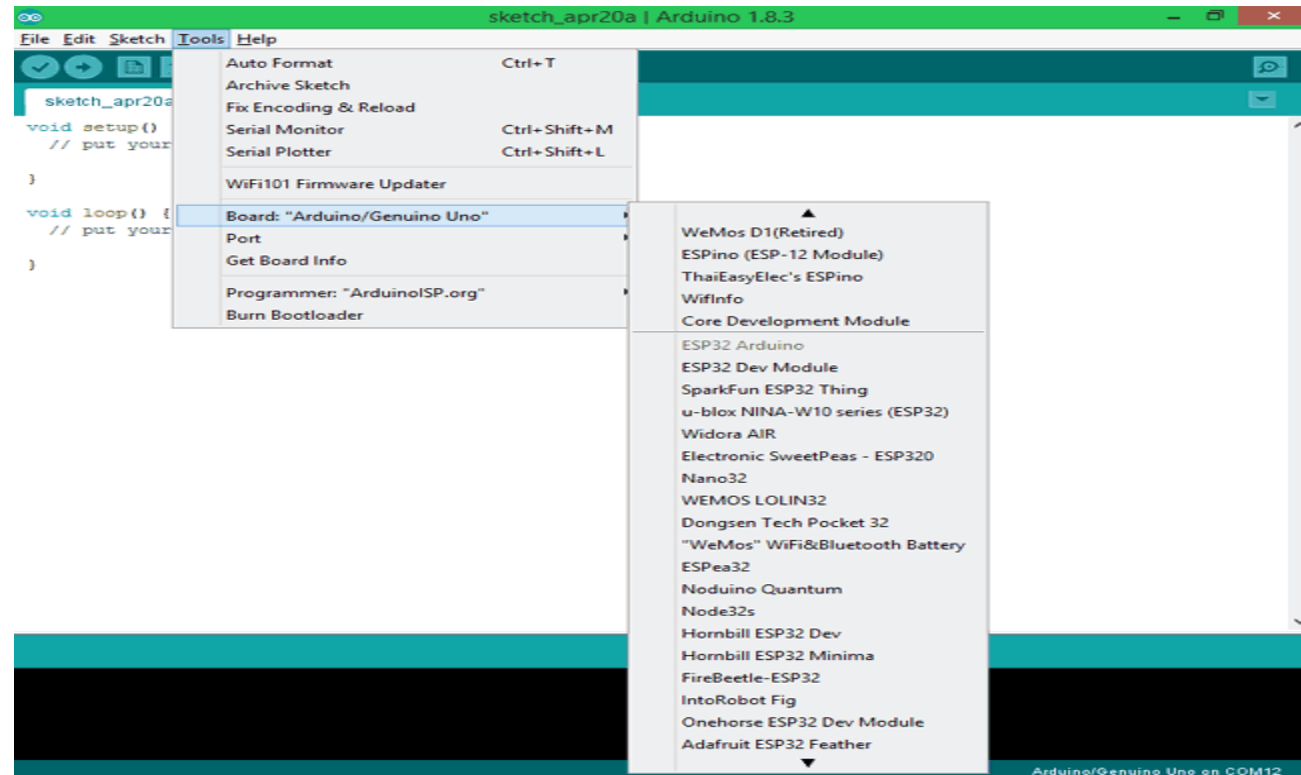




- **STEP 5:** Make sure you have an active internet connection and wait while the installation gets complete. It may take few minutes based on the speed of your internet connection.

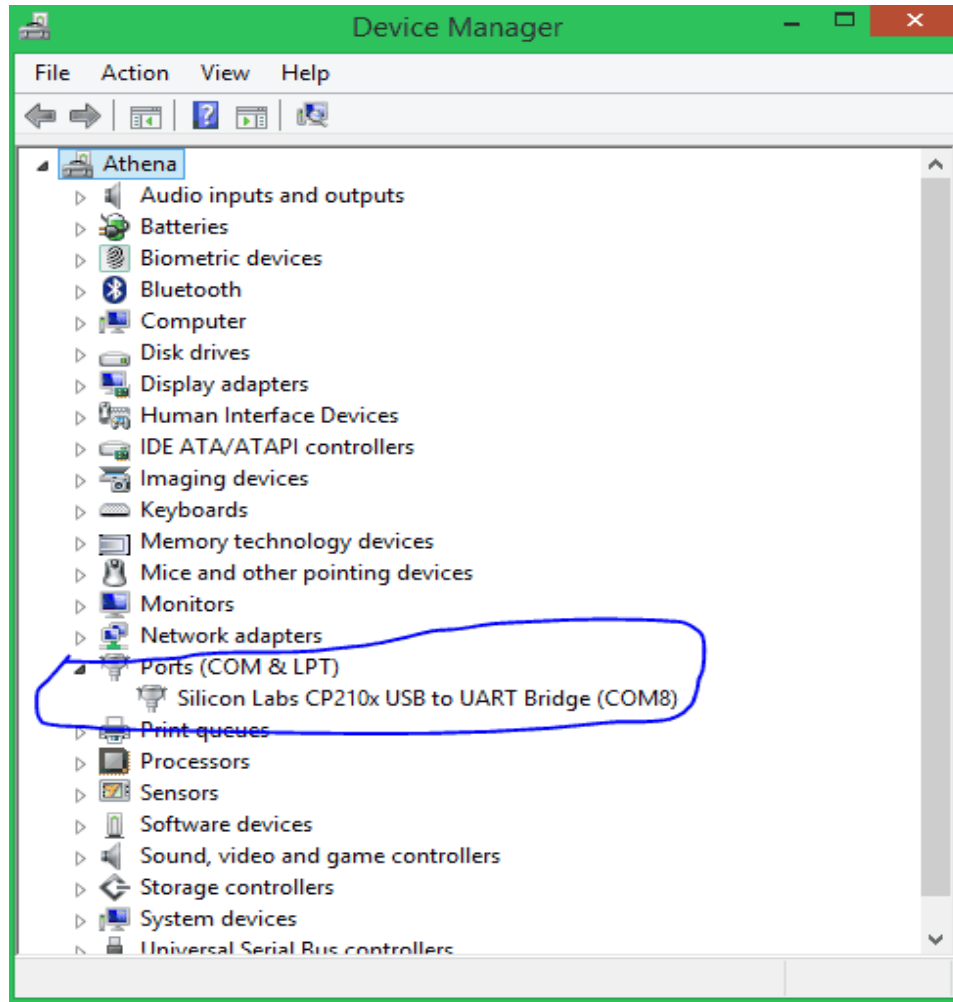
# Programming ESP32 with Arduino IDE:

- **STEP 1:** Connect your ESP32 board to your computer through the micro-USB cable. Make sure the red LED goes high on the module to ensure power supply.
- **STEP 2:** Start the Arduino IDE and navigate to *Tools -> Boards* and select *ESP32Dev* board as shown below

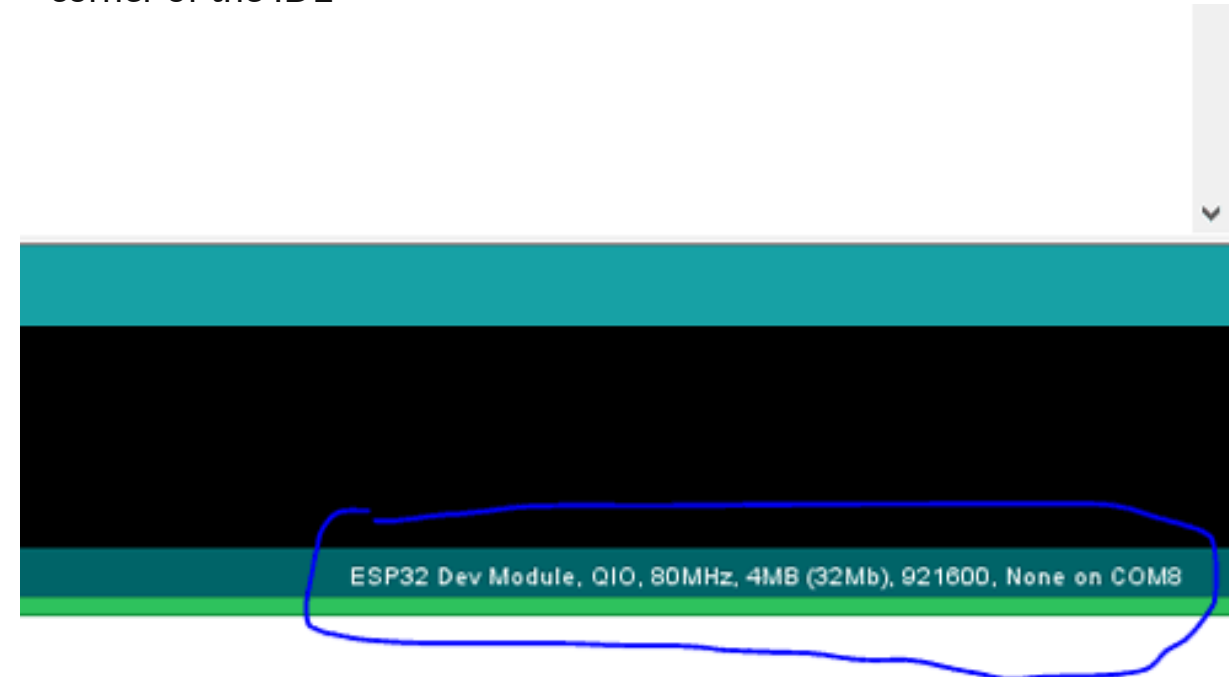




- **STEP 3:** Open device manager and check to which com port your ESP32 is connected to. Mine is connected to COM 8 as shown below



- **STEP 4:** Go back to Arduino IDE and under *Tools -> Port* select the Port to which your ESP is connected to. Once selected you should see something like this on the bottom left corner of the IDE

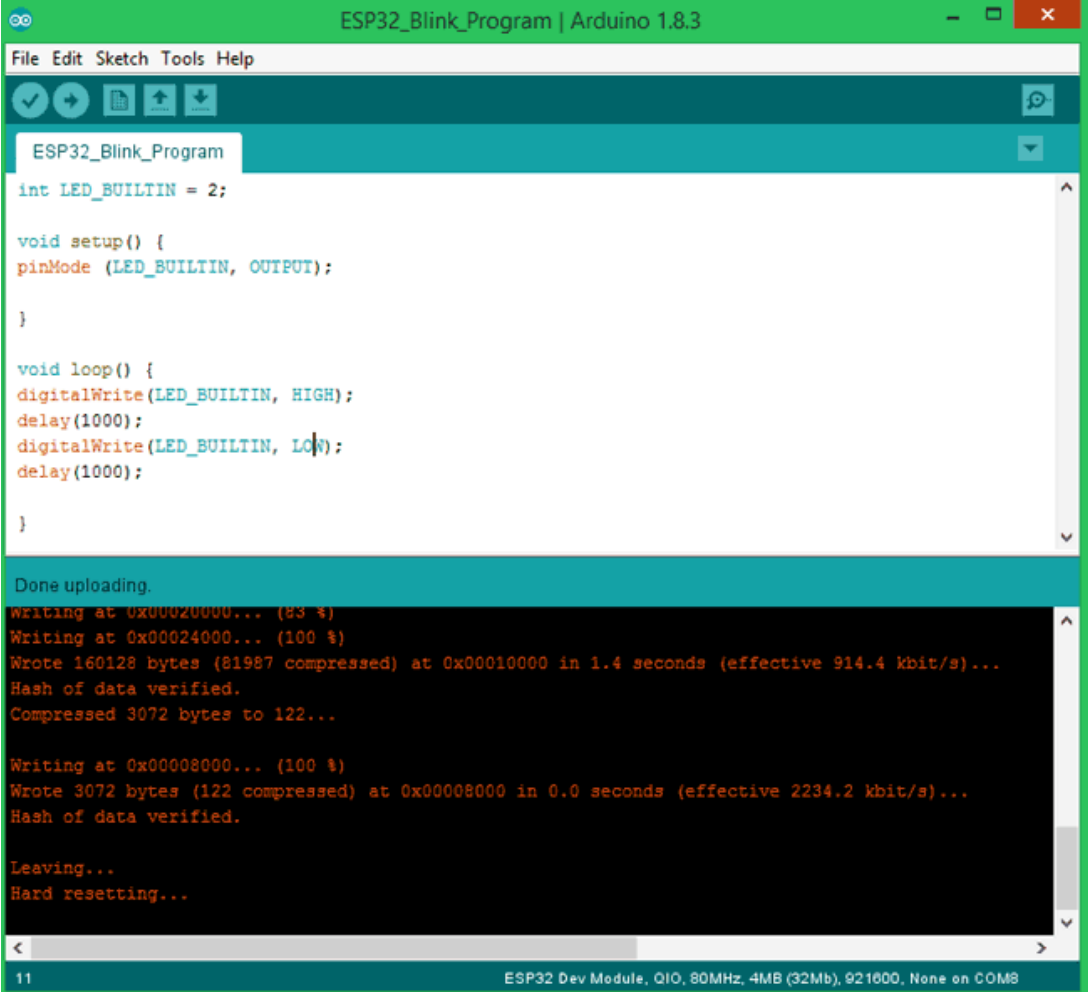


**STEP 5:** Let's upload the Blink Program, to check if we are able to program our ESP32 module. This program should blink the LED at an interval of 1 second.

```
int LED_BUILTIN = 2;
void setup() {
  pinMode (LED_BUILTIN, OUTPUT);
}
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

The program is very similar to the Arduino blink code ,But one change is that, here in ESP32 the LED on board is connected to pin number 2, while for Arduino it will be connected to pin number 13.

**STEP 6:** To upload the code, just click on upload and you should see the Arduino console displaying the following if everything works as expected



The screenshot shows the Arduino IDE interface with the title bar "ESP32\_Blink\_Program | Arduino 1.8.3". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checking, uploading, and other functions. The main editor window displays the following C++ code:

```
ESP32_Blink_Program
int LED_BUILTIN = 2;

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

Below the code editor is a console window with a teal header that says "Done uploading.". The console output shows the following messages:

```
Writing at 0x00020000... (83 %)
Writing at 0x00024000... (100 %)
Wrote 160128 bytes (81987 compressed) at 0x00010000 in 1.4 seconds (effective 914.4 kbit/s)...
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 2234.2 kbit/s)...
Hash of data verified.

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

At the bottom of the IDE, a status bar displays "11" on the left and "ESP32 Dev Module, Q10, 80MHz, 4MB (32Mb), 921600, None on COM8" on the right.