

Preparing your Arduino IDE

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:"

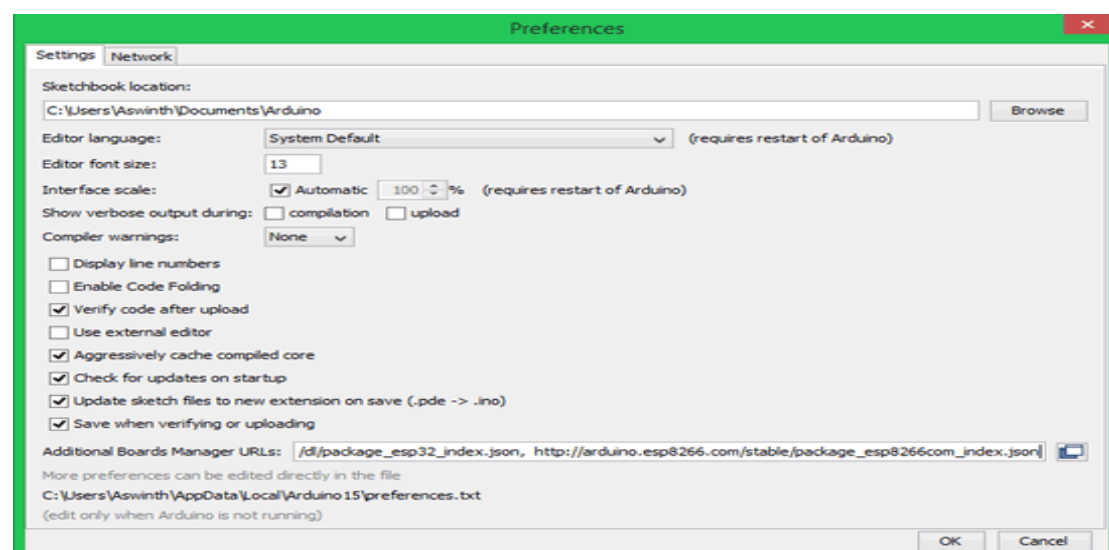
STEP 3: This text box might be empty or might also contain some other URL if you have used it previously for ESP8266. If it is empty simply paste the below URL into the text box

https://dl.espressif.com/dl/package_esp32_index.json

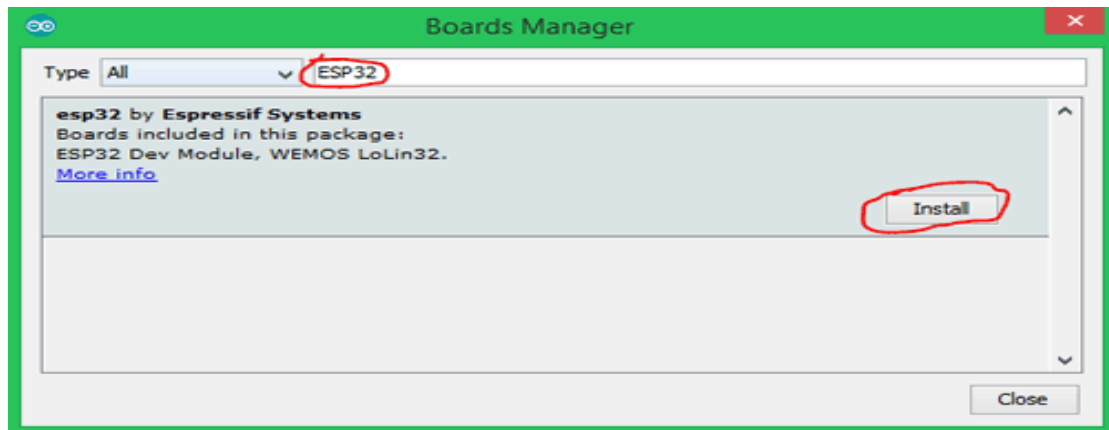
If the text box already contains some other URL just add this URL to it, separate both with a comma (,). Mine already had the ESP8266 URL I just added this URL to and added a comma, like this

```
https://dl.espressif.com/dl/package_esp32_index.json,  
http://arduino.esp8266.com/stable/package_esp8266com_index.js  
on
```

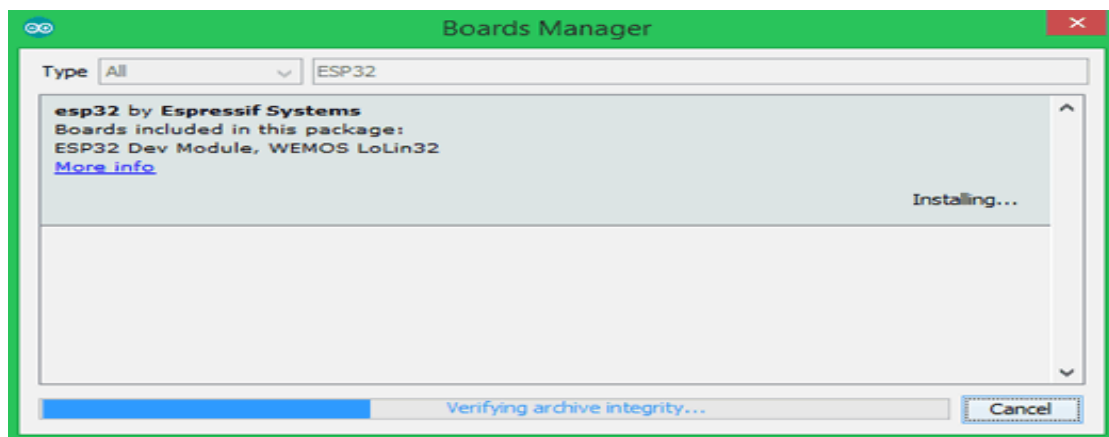
Once done, my preferences windows looked like this below. Just click on OK and the window will disappear.



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.

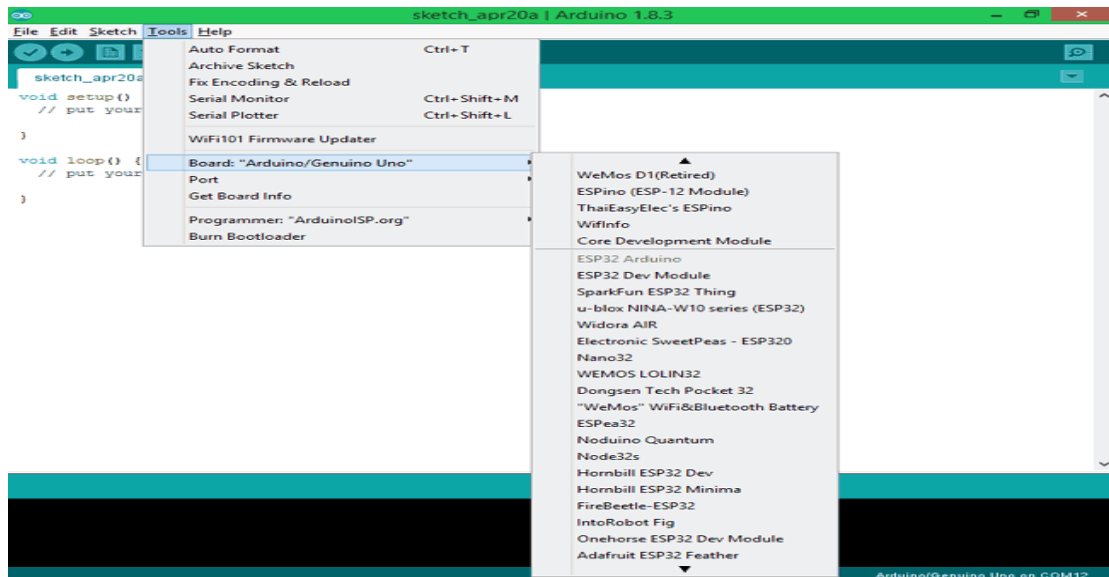


That is it now our **Arduino IDE** is prepared to work with **ESP32**. Let's go ahead and check if it is working.

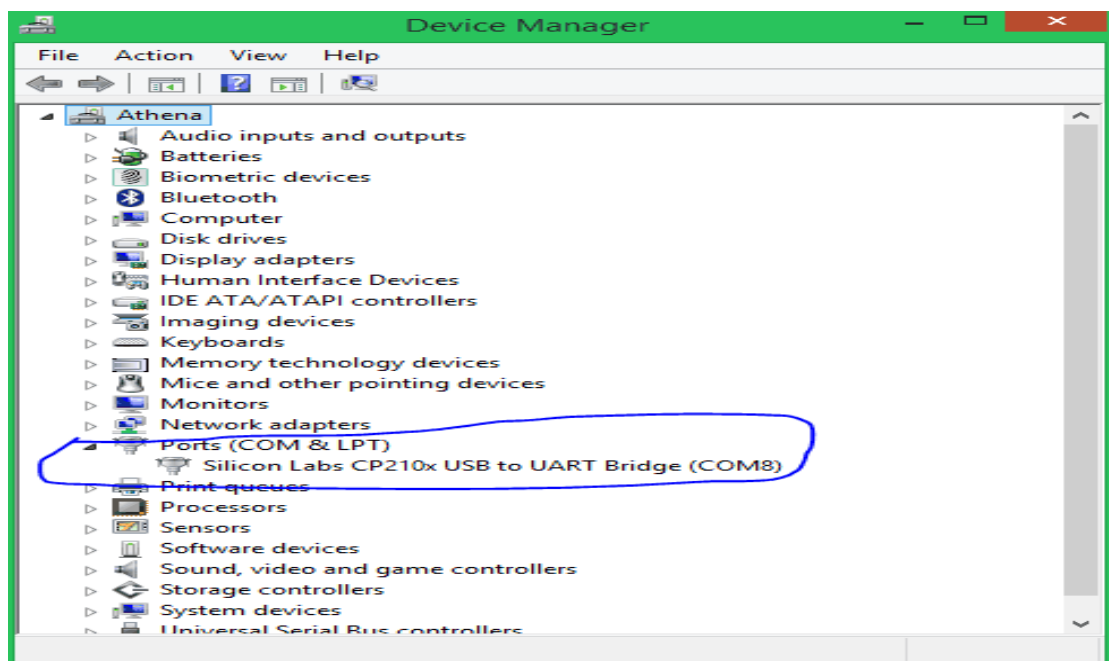
Programming ESP32 with Arduino IDE:

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.

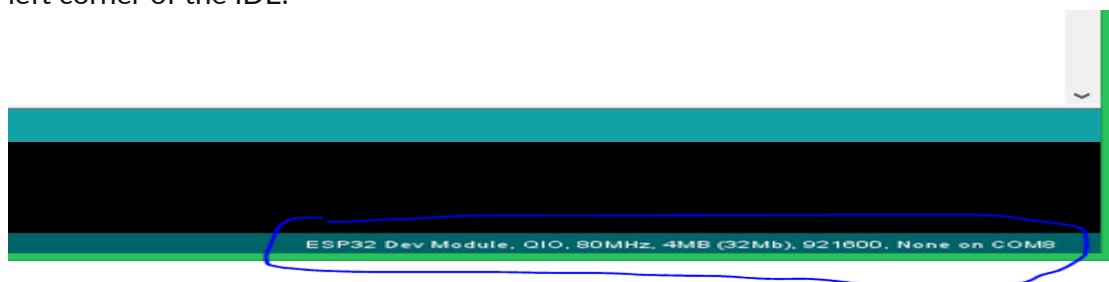
STEP 2: Start the Arduino IDE and navigate to *Tools -> Boards* and select *ESP32Dev* board as shown below



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.

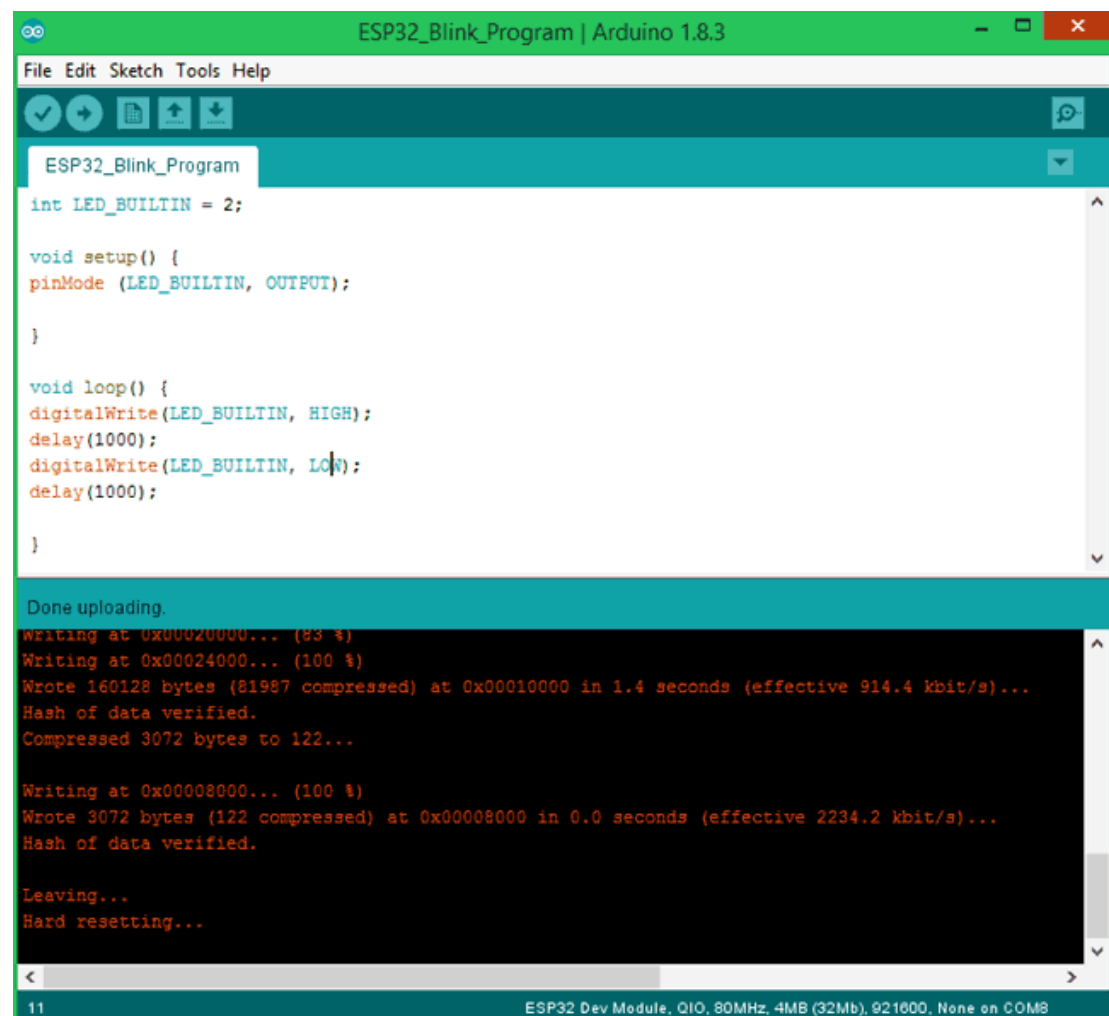
```
int LED_BUILTIN = 2;

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 hence I am not explain them in detail. 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". The toolbar contains icons for opening, saving, and uploading files. The sketch editor displays the following code:

```
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, the console shows the upload progress:

```
Done uploading.
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...
```

The status bar at the bottom indicates "11" and "ESP32 Dev Module, Q10, 80MHz, 4MB (32Mb), 921600, None on COM8".

Note: For some modules, you might have to hold the Boot button during uploading to avoid error.

That is it we have successfully uploaded our first code to our ESP32 board. **My module with its LED blinking is shown below**

This is how **Programming ESP32 using Arduino IDE** can be implemented. You can go ahead and try the other example programs which are available at *File -> Example -> ESP32* to work with other functionalities of the ESP32. If you have had any problem in getting this work, feel free to post the query on the comment sections below. You can also use the Forum for getting technical help.

```
int LED_BUILTIN = 2;

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

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