



Arduino + ESP32 Instruction

Install Arduino IDE

The **Arduino IDE** is the primary tool we will use in the class for programming the **ESP32** based microcontroller, though there are a lot of other options, such as **Visual Studio Code**. We will also use the **Arduino programming language**, which is essentially a wrapper of C/C++. You can also use **MicroPython**, a subset of Python that designed for to run on a microcontroller. However, we will not cover the syntax of MicroPython, and it is up to you to figure this out.

Depending on your operating system, the installation may vary. Contact TA if you encounter any issues and we can figure it out together. Below we will use Windows system as an exemplar, with side notes for Mac users.

Installation steps

1. Install Arduino IDE. We will use the **Arduino IDE 2.2.1** version: here is the link for **Windows** or **MacOS**.
 2. Install ESP32 support. Go to File - Preference, as in Figure 1. In the box Additional boards manger URLs, paste **https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json**
- Click OK.

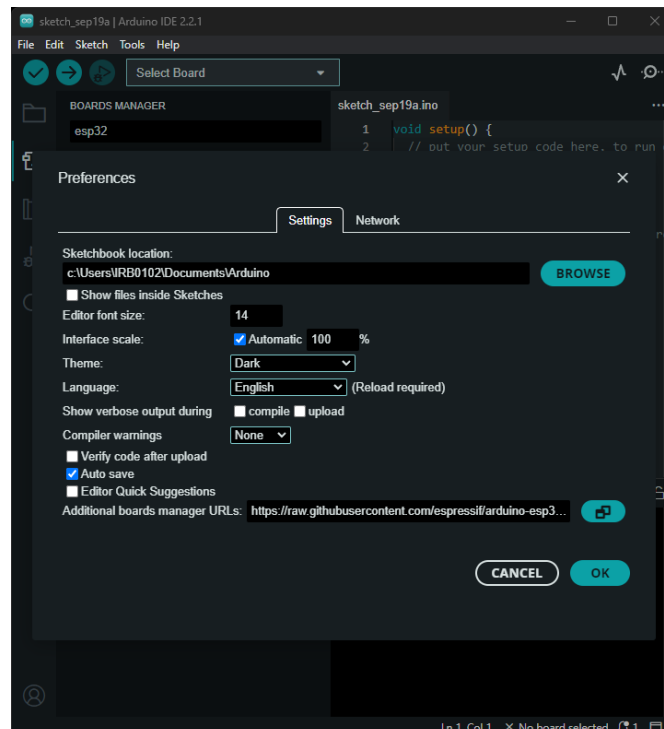


Figure 1

3. Click Tools - Board: - Boards Manger... (Figure 3).

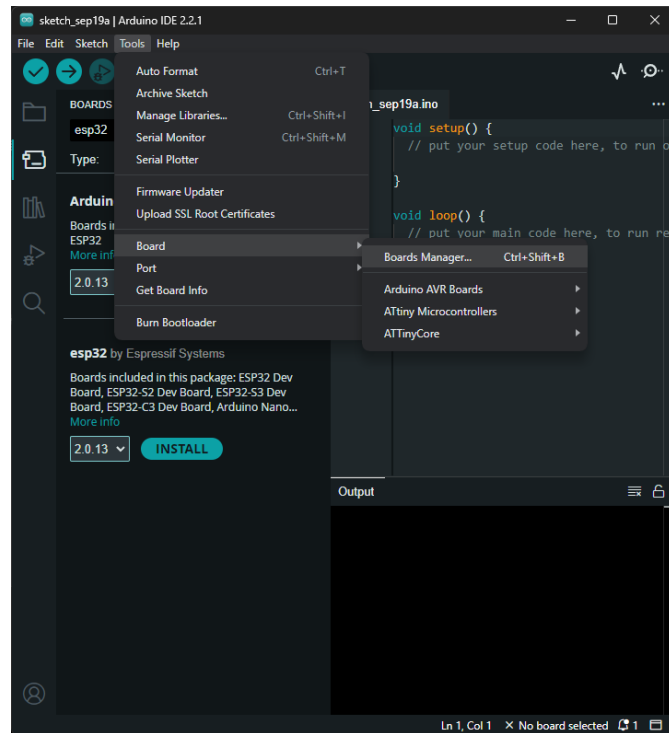


Figure 2

4. In the Boards Manager, type ESP32 and click Install.

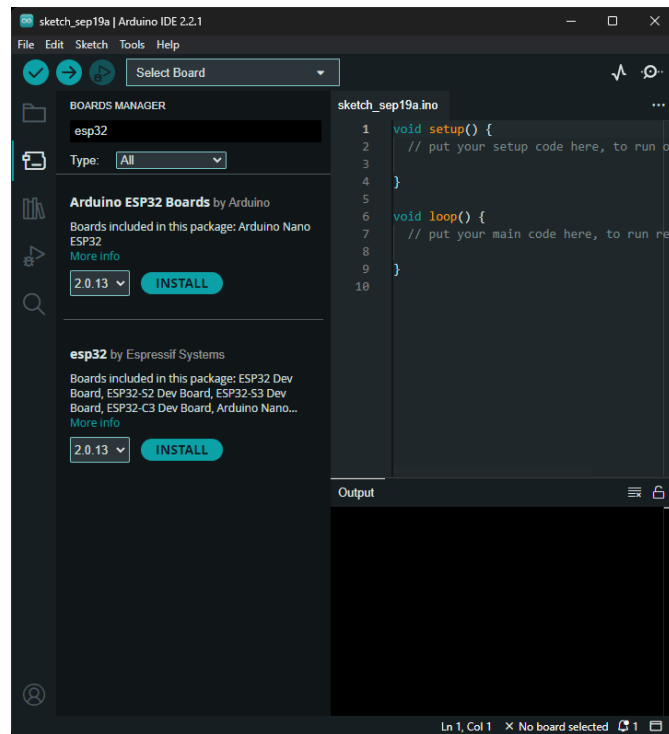


Figure 3

Verify and Upload your Code

Let's test if the installation is successful.

1. Plug the ESP32 board to your computer. In the IDE, Select Tools - Board - NodeMCU-32S (Figure4)

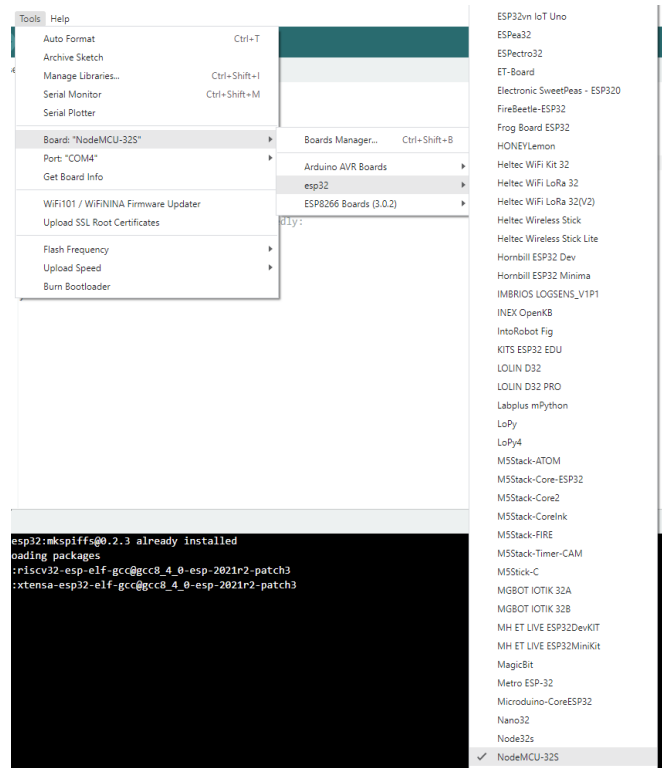


Figure 4

2. Copy and Paste the following code sketch to the canvas.

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```

3. Select the Port that talks to your ESP32. In my case, it's COM4 (Figure 5). On Mac, it is something similar to '/dev/tty.usbserial-A6004byf 9600'. Yours may be different. If you don't see the correct port in your Arduino IDE, you may need to install the UART driver: [CP210x USB to UART Bridge VCP Drivers](#).

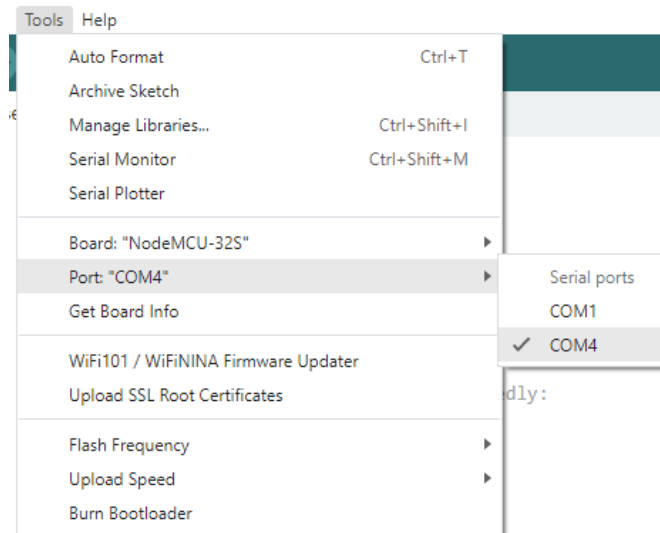


Figure 5

4. Upload the code to ESP32. Press the Upload button in the Arduino IDE. For our ESP board, you need to hold-down the “BOOT” button in your board until you see the code is uploading, like what is showed in Figure 7. The ”BOOT” button is to the right of the micro USB cable of your board. See Figure 6.

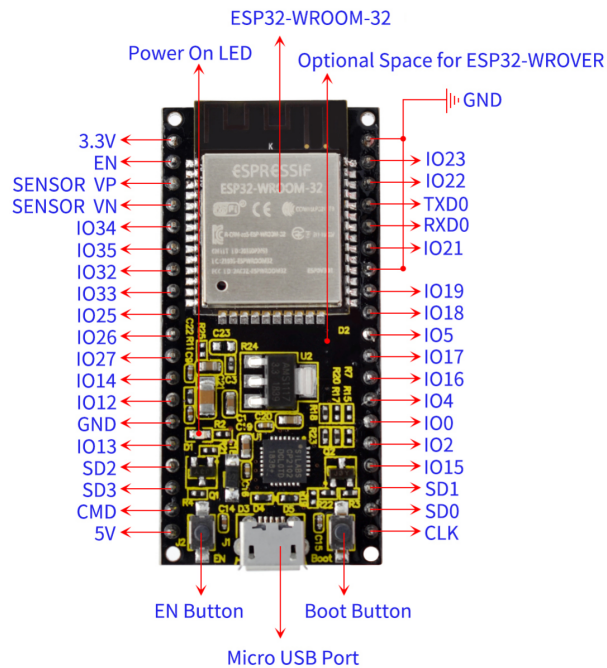


Figure 6

```

Sketch uses 219888 bytes (16%) of program storage space. Maximum is 1310720 bytes.
Global variables use 16888 bytes (4%) of dynamic memory, leaving 311592 bytes for local variables. Maximum is 327680 bytes.
esptool.py v4.2.1
Serial port COM6
Connecting.....
Chip is ESP32-D0WQ2 (revision 3)
Features: WiFi, BT, Dual Core, 240MHz, Vref calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 48:91:11:9f:02:f0
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Flash will be erased from 0x00001000 to 0x00007fff...
Flash will be erased from 0x00008000 to 0x0000ffff...
Flash will be erased from 0x0000c000 to 0x0000ffff...
Flash will be erased from 0x00010000 to 0x0000ffff...
Compressed 17472 bytes to 12125...
Writing at 0x00001000... (100 %)
Wrote 17472 bytes (12125 compressed) at 0x00001000 in 0.4 seconds (effective 387.1 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 128...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.1 seconds (effective 455.9 kbit/s)...
Hash of data verified.
Compressed 8192 bytes to 47...
Writing at 0x0000c000... (100 %)
Wrote 8192 bytes (47 compressed) at 0x0000c000 in 0.2 seconds (effective 425.2 kbit/s)...
Hash of data verified.
Compressed 215472 bytes to 128878...
Writing at 0x00010000... (12 %)
Writing at 0x0001d000... (25 %)
Writing at 0x0002f000... (37 %)
Writing at 0x0002f000... (50 %)

```

Figure 7

5. If everything works out, you should see a blue LED flash at 1Hz on your board.