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 grammar 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 exampler, with side notes for Mac users.

Installation steps

1. Install Arduino IDE. We will use the Arduino IDE 2.0.0 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.

3. Click Tools - Board: - Boards Manger... (Figure 3).
4. In the Boards Manager, type ESP32 and click Install.

![Figure 2](image)

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 (Figure 4)
2. Copy and Paste the following code sketch to the canvas.

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