Arduino + ESP32 Instruction

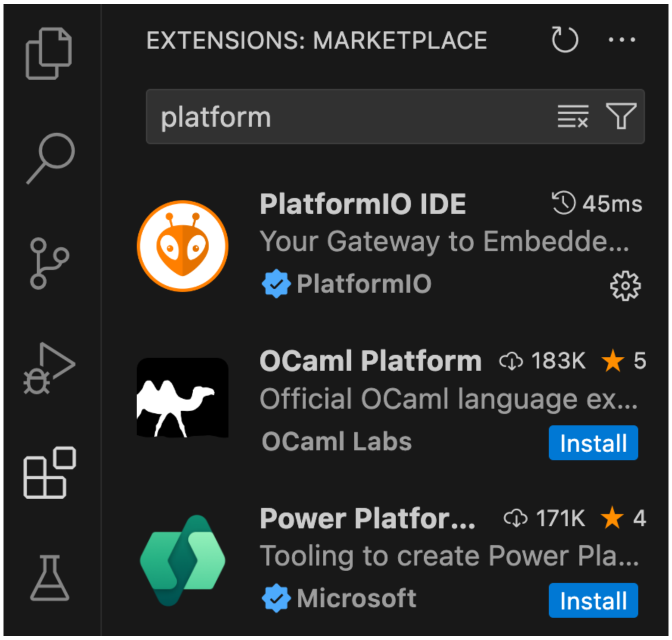
Install Visual Studio Code and PlatformIO plugin

The Arduino IDE is the recommended default IDE for programming Arduino/ESP32 board, but its function can be limited. Instead, in this course we will use Visual Studio Code and the PlatformIO as the main programming environment. 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 MacOS as an exampler.

Installation steps

1. Install VS Code. There is nothing much to say here, just find the one that match your OS.



EXTENSIONS: MARKETPLACE

platform

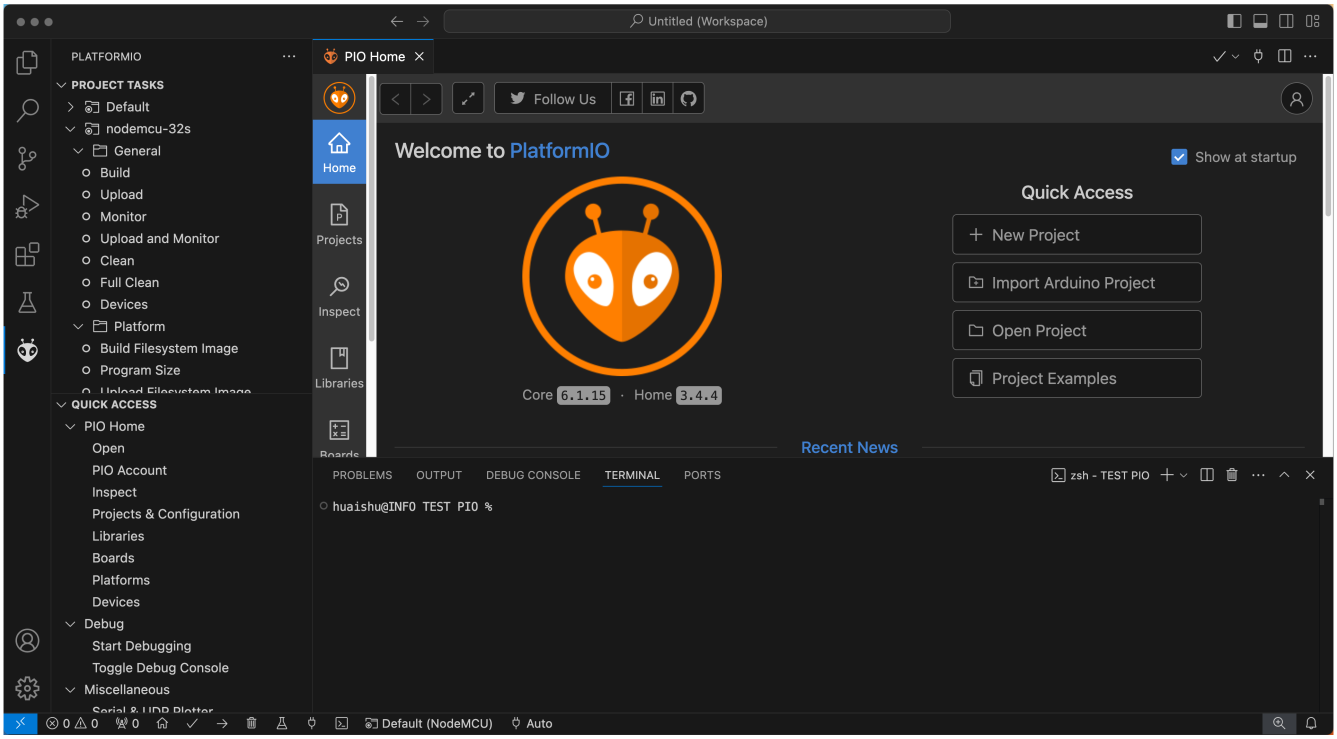
PlatformIO IDE 45ms
Your Gateway to Embedde...
PlatformIO

OCaml Platform 183K ★ 5
Official OCaml language ex...
OCaml Labs

Power Platfor... 171K ★ 4
Tooling to create Power Pla...
Microsoft

Figure 1: Install PIO plugin

2. Install PlatformIO (PIO) extension. PIO is a versatile embedded software programming platform that supports a whole lot of different hardware board, framework, library, etc. To install PIO, you open VS Code first, on the left, click the “Extension” button, and search PlatformIO (See Figure 1). Click the Install button in blue. It will take a while to install all the dependencies and ask you to restart the plugin. You should now see an “alien” icon in the left menu bar and the PIO Home tab (Figure 2).



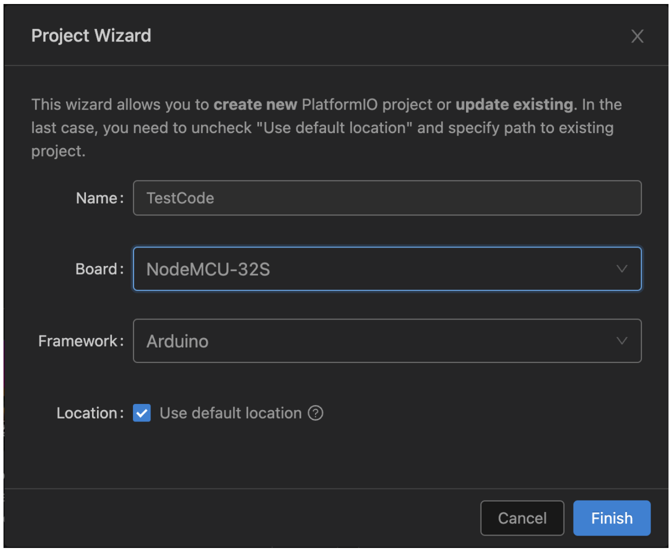
```
huaishu@INFO TEST PIO %
```

Figure 2: PIO Home tab

Verify and Upload your Code

Let’s test if the installation is successful.

1. Create a new project. Click the “New Project” button, give it a project name. On the Board window, type “NodeMCU”, it should allow you to choose something like “NodeMCU-32S”. Choose the default “Arduino” framework, as showed in Figure 3. It may take a while for the first time we run the Arduino framework, but then you should see the basic folder structure. That looks similar to many coding framework, e.g., you have a lib folder, a src folder, etc. Under the src folder, you will find the “main.cpp” (Figure 4).



Project Wizard

This wizard allows you to **create new** PlatformIO project or **update existing**. In the last case, you need to uncheck "Use default location" and specify path to existing project.

Name: TestCode

Board: NodeMCU-32S

Framework: Arduino

Location: ☑ Use default location

Figure 3: Create a new project.

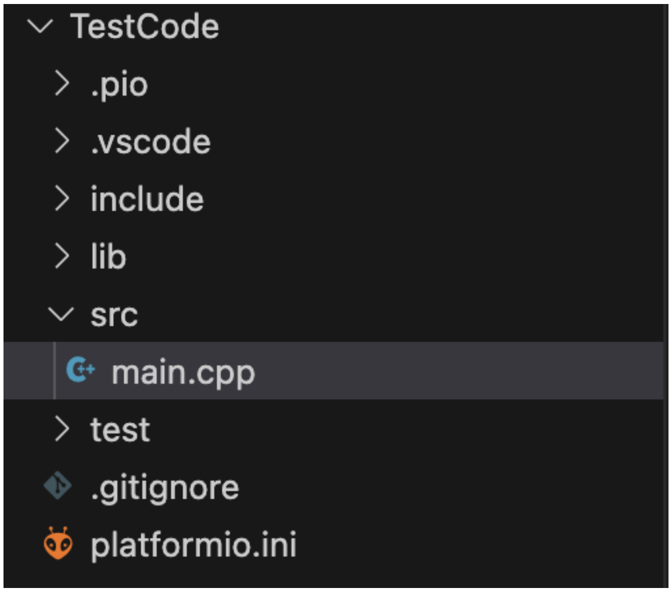


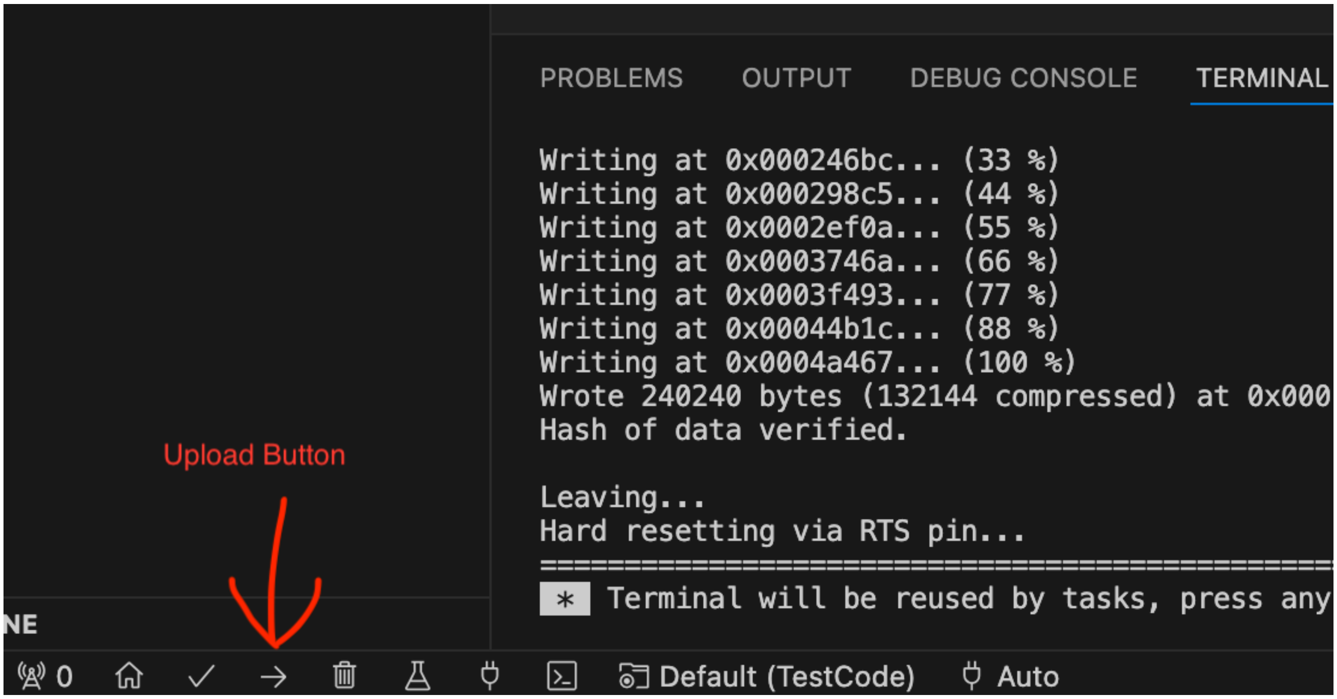
Figure 4: Main cpp file to work with.

2. Edit the “main.cpp” file. Copy and Paste the following code sketch.

```
#include <Arduino.h>

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. Plugin your ESP32 board to the USB port. The PIO should be able to detect the new board. Now click the “Upload” button at the bottom of the VS Code. It will compile your code, and begin writing the compiled file to the ESP32 hardware (Figure 5).



```
Writing at 0x000246bc... (33 %)
Writing at 0x000298c5... (44 %)
Writing at 0x0002ef0a... (55 %)
Writing at 0x0003746a... (66 %)
Writing at 0x0003f493... (77 %)
Writing at 0x00044b1c... (88 %)
Writing at 0x0004a467... (100 %)
Wrote 240240 bytes (132144 compressed) at 0x000
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
==================================================
 *  Terminal will be reused by tasks, press any
```

Figure 5: Upload the code.

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