Setting Up PlatformIO

Your printable reference for Episode 2 — VS Code, PlatformIO, and your first project.

EP 02

Why PlatformIO Instead of Arduino IDE?

Feature	Arduino IDE	PlatformIO
IntelliSense / autocomplete	No	Yes — full C++ autocomplete as you type
Library version pinning	No	Yes — locked in platformio.ini, never breaks
Multi-board environments	No	Yes — test on DevKit and S3 from one project
Professional toolchain	No	Yes — same setup used in industry

Installation — Two Steps

Do these once. You will never need to do them again.

Step 1 — Install VS Code

Go to: code.visualstudio.com

Download the installer for your OS (Windows / Mac / Linux).

Run the installer — accept all defaults.

VS Code is free and used by millions of developers worldwide.

Step 2 — Install the PlatformIO Extension

In VS Code: click the Extensions icon in the left sidebar (four squares).

Keyboard shortcut: Ctrl+Shift+X on Windows, Cmd+Shift+X on Mac.

Search for: PlatformIO IDE

Click Install on the result published by PlatformIO.

Wait 1–2 minutes — it downloads and configures the toolchain.

When done: the PlatformIO alien-head icon appears in the left sidebar.

PlatformIO Home — Three Sections to Know

Projects

All your builds in one place. Every ESP32 project, every board variant, listed here.

Libraries

10,000+ libraries searchable. Find FastLED, GxEPD2, Adafruit — copies the install line into your platformio.ini for you.

Boards

1,000+ supported boards. Use this to find the exact board identifier string for your platformio.ini.

Your First Project — File by File

PlatformIO generates this structure automatically. Here is what each file does.

PROJECT FOLDER STRUCTURE

```
my_project/
  ■■■ platformio.ini
  ■■■ src/
  ■   ■■■ main.cpp
  ■■■ include/
  ■■■ lib/
  ■■■ .pio/  ← auto-generated, do not edit
```

platformio.ini

The control file for your entire project.

Board, framework, libraries, and build flags all live here.

One project can have multiple board environments.

src/main.cpp

Your code lives here. Always.

Must include: #include <Arduino.h>

Contains setup() and loop() — same as Arduino.

include/

Your own header files (.h) go here.

Local libraries specific to this project.

The platformio.ini File — Anatomy

platformio.ini

```
[env:esp32dev]
platform = espressif32
board = esp32dev
framework = arduino
; Optional: libraries go here:
lib_deps = fastled/FastLED @ ^3.7.0
monitor_speed = 115200
```

[env:esp32dev]

The environment name. You choose this. Used when switching between boards.

platform

The chip family. espressif32 covers all ESP32 variants.

board

The exact board ID. Search for yours at platformio.org/boards

lib_deps

Libraries. Paste the line from the PlatformIO Registry — version is pinned automatically.

monitor_speed

Serial baud rate. Must match Serial.begin() in your code. 115200 is the standard.

Your First Project — The Code

The main.cpp skeleton and what to watch out for.

The main.cpp Skeleton

Every PlatformIO project starts here. This is your blank canvas.

src/main.cpp

```
#include <Arduino.h>

// Put your pin definitions and variables here
const int LED_PIN = 2;

void setup() {
  Serial.begin(115200);  // Start serial monitor
  pinMode(LED_PIN, OUTPUT);  // Set pin as output
}

void loop() {
  digitalWrite(LED_PIN, HIGH);  // LED on
  delay(1000);
  digitalWrite(LED_PIN, LOW);   // LED off
  delay(1000);
}
```

Important: #include <Arduino.h> is required in PlatformIO. Without it you will get confusing errors — this is the most common beginner mistake.

In the Arduino IDE this include is added silently for you. In PlatformIO you write it yourself. It must be the first line.

Build, Upload & Monitor — The Three Buttons

✓ Build	↑ Upload	■ Monitor
Ctrl+Alt+B	Ctrl+Alt+U	Ctrl+Alt+S
Compiles your code and checks for errors. Do this before uploading.	Builds and sends code to your board. Board must be connected via USB.	Opens the serial monitor. Use this to see Serial.println() output from your board.

COM Port not detected?

Install the CP2102 or CH340 driver for your board. PlatformIO will auto-detect the port once the driver is in place.