**ESP32 Real Hardware Practice - LED**

*Andreika Luna Alghivari*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: andreikaluna@student.ub.ac.id*

**Abstract**

This practical session aims to provide a fundamental understanding of using the ESP32 microcontroller to control simple electronic components, specifically LEDs. In this experiment, two LEDs are controlled using an ESP32 development board, with programming and deployment handled through Visual Studio Code (VSCode) integrated with the PlatformIO extension. The Arduino programming framework is used to write and upload the code to the board. Additional components include male-to-male and male-to-female jumper wires, as well as a motherboard (breadboard) for circuit assembly. The implemented program allows the two LEDs to blink alternately at a specified time interval. This practice introduces essential concepts such as digital pin configuration, delay functions, and basic microcontroller programming logic. The results demonstrate that the ESP32 can reliably control two LEDs according to the programmed sequence.

*Keywords: ESP32, LED Blinking, Embedded Systems, Breadboard, Visual Studio Code, PlatformIO.*

**1. Introduction**

* 1. **Background**

The rapid advancement of embedded systems and Internet of Things (IoT) technologies has increased the need for flexible and programmable microcontrollers. ESP32 is widely used due to its low cost, high performance, and built-in Wi-Fi and Bluetooth features. Controlling basic components such as LEDs is a fundamental step in understanding how microcontrollers interact with external hardware. This practice focuses on implementing a real hardware setup using the ESP32 board to control two LEDs through digital output pins. The development environment utilizes Visual Studio Code integrated with PlatformIO, allowing for efficient programming, code uploading, and testing using the Arduino framework. This approach provides a practical foundation for further exploration in embedded system applications.

* 1. **Objective**

1. To introduce the basic concept of digital output using the ESP32 microcontroller.
2. To implement a simple LED blinking program using two LEDs connected to the ESP32.
3. To understand the process of configuring hardware and software for embedded system development.
4. To practice assembling basic electronic circuits using jumper wires and a breadboard.

**2. Methodology**

**2.1 Tools & Materials**

ESP32, 2 LED, Jumper Wires, Breadboard, USB Cable, Arduino IDE, PlatformIO and Visual Studio Code.

**2.2 Implementation Steps**

1. Install the ESP32, 2 LEDs, Jumper Wires required on the breadboard according to the rules. (Note the length of the short LED cable is negative, the long one is positive)
2. Install the Silicon Labs CP210x driver so that the computer can recognize the ESP32 hardware.
3. Change the platformio.ini code with the code on the module, pay attention to the upload\_port and monitor\_port sections. On macOs, with the command ls /dev/tty.\* on the terminal to get upload\_port and monitor\_port.
4. Enter the LED light code that has been made in the previous module in the main.cpp file, pay attention to the decration of the lights. Make sure it matches the ESP32 hardware that has been assembled. Then, upload the file.
5. After uploading the file, the lamp will automatically turn on.

**3. Results and Discussion**

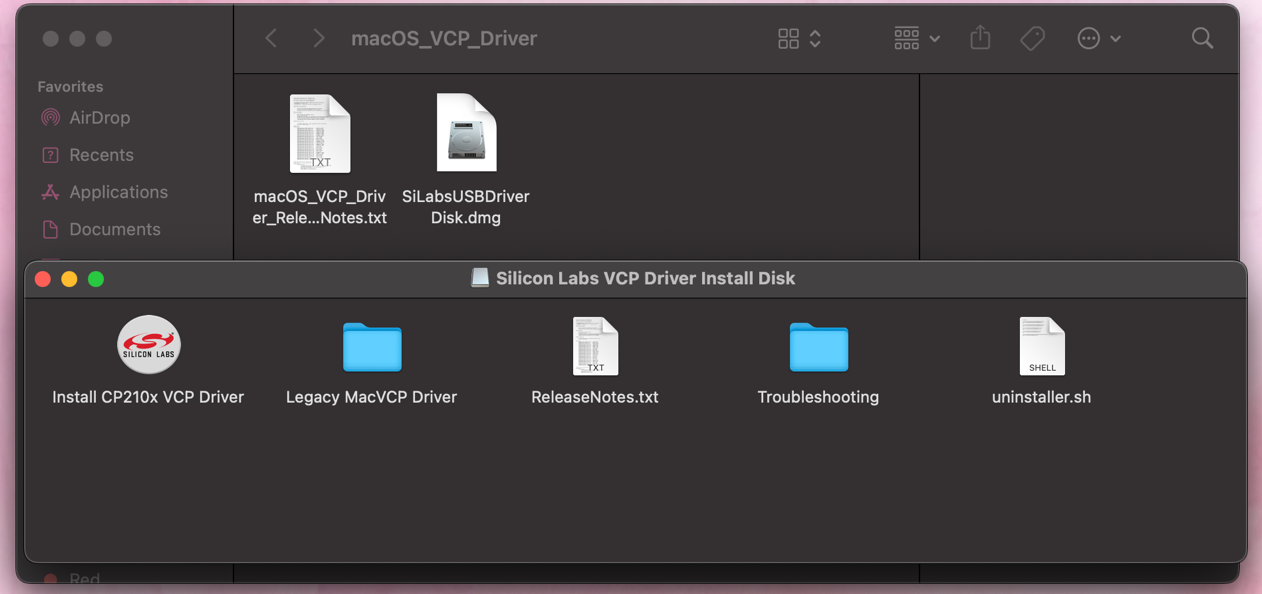
**3.1 Experimental Results**

1. Create new project in platform.io

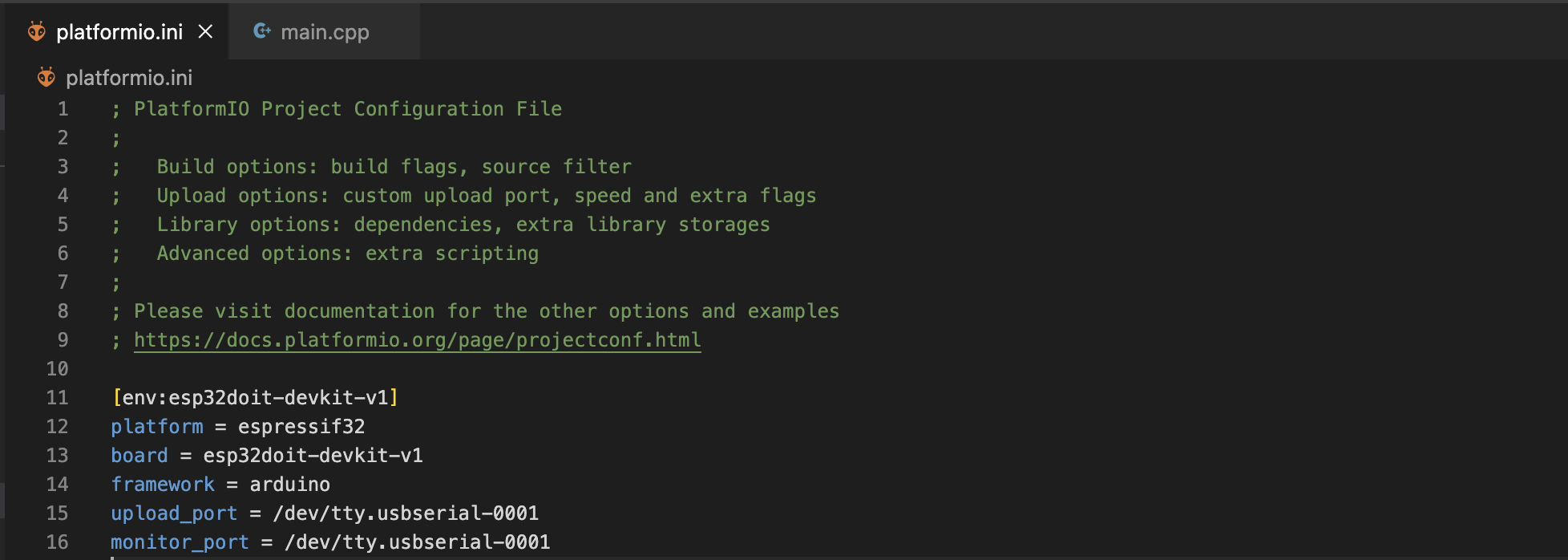
A screenshot of a computer

Description automatically generated

1. Install driver Silicon Labs CP210x



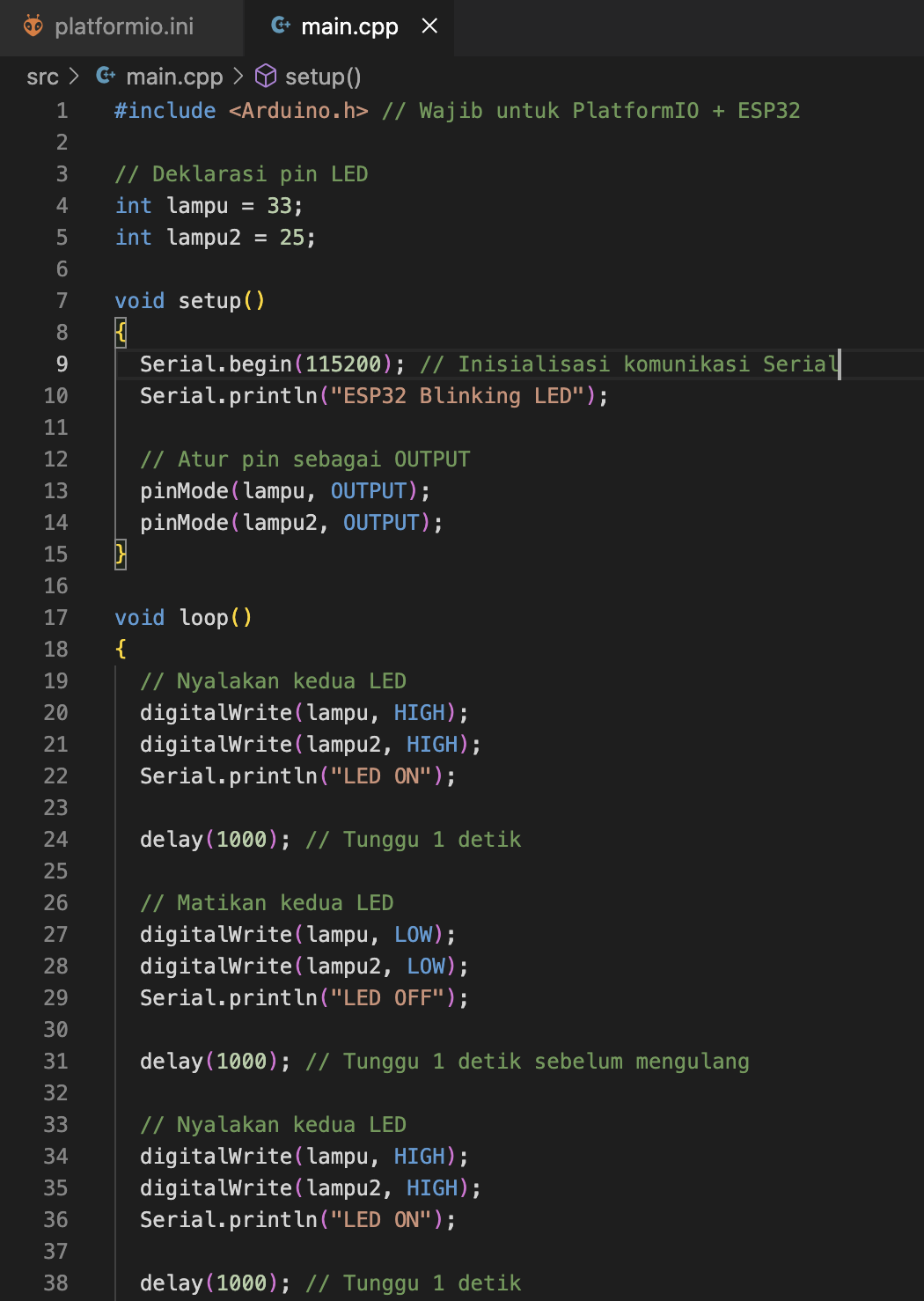
1. Platformio.ini code and terminal with command for search the port on macOs.

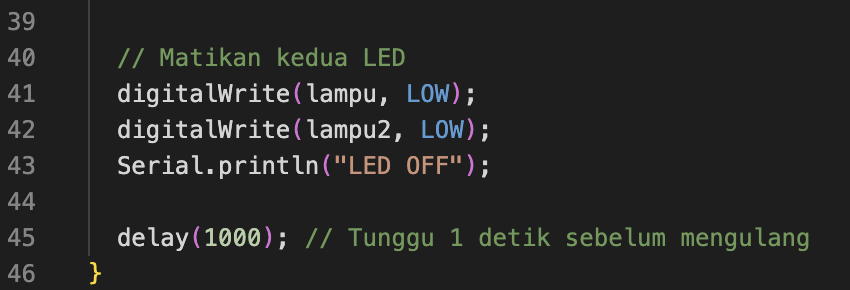


A screenshot of a computer

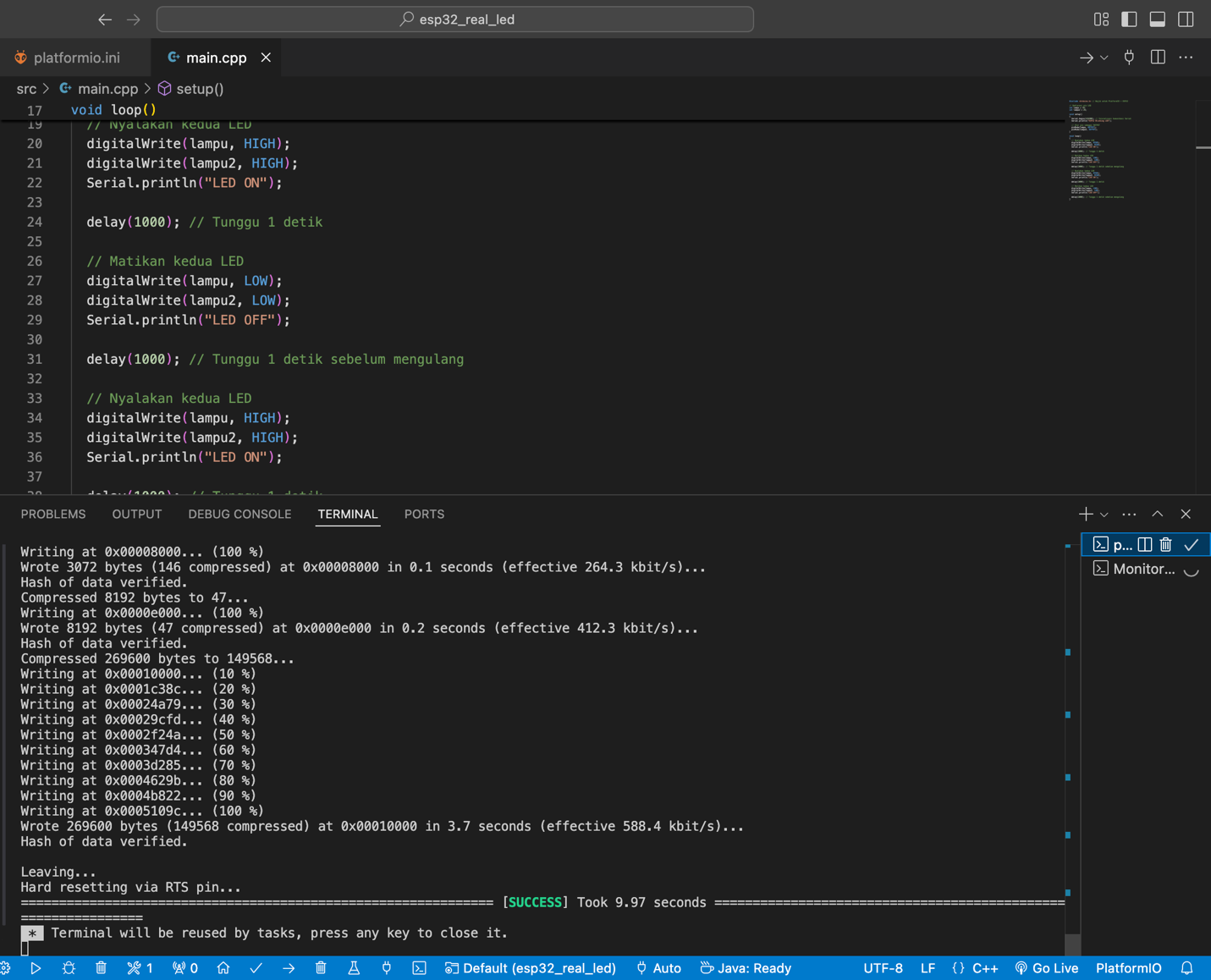
Description automatically generated

1. Code Main.cpp

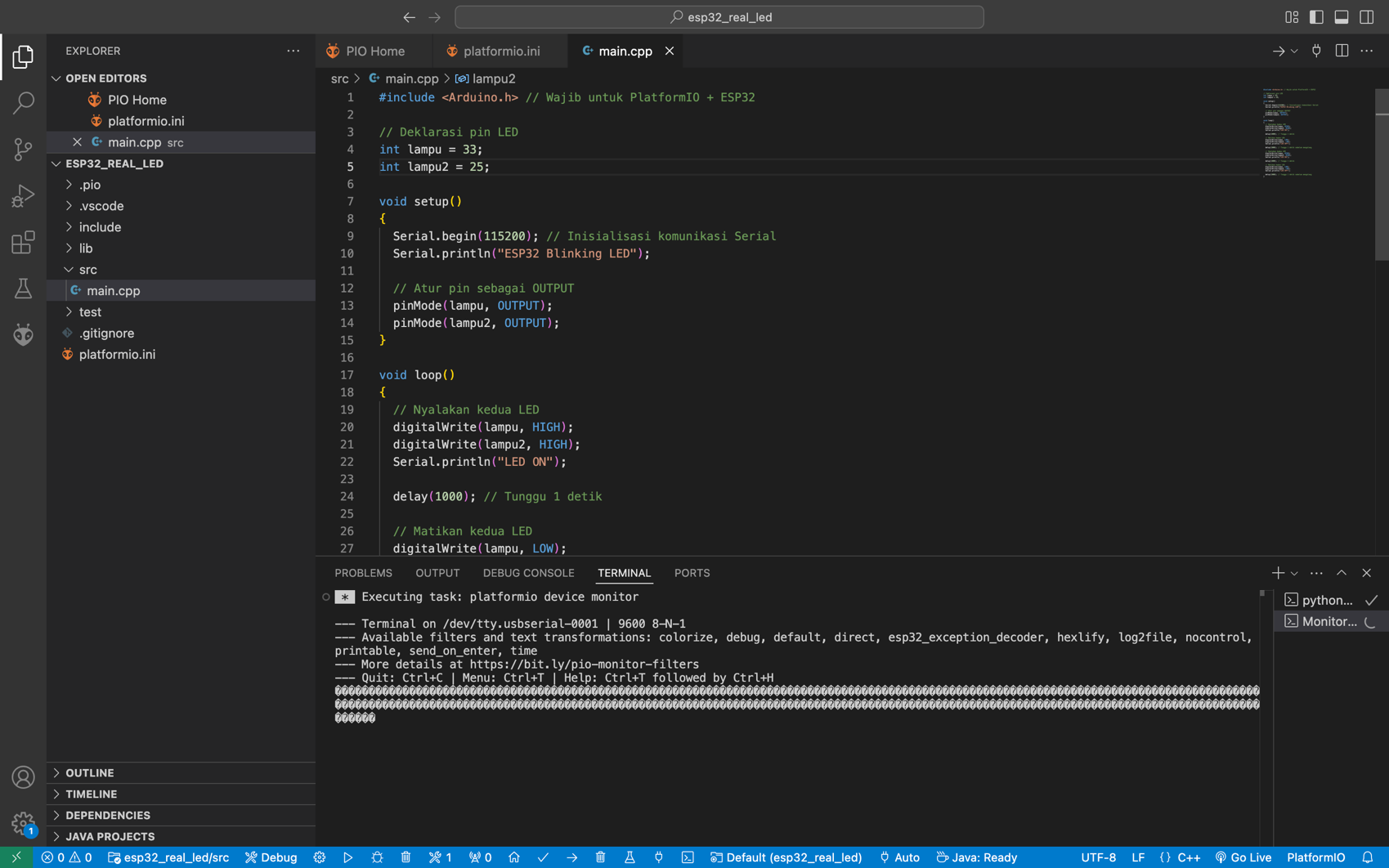




1. Upload file



1. After uploading, the terminal will change like this and the LED will light up.



1. All set!

A circuit board with wires connected to it

Description automatically generated

The real hardware implementation using the ESP32 microcontroller to control two LEDs was successfully carried out. The LEDs responded correctly to the programmed instructions by blinking alternately according to the defined time interval. This practice demonstrated that the ESP32 is capable of managing digital output operations effectively using the Arduino framework within the PlatformIO environment. It also validated the correct wiring and configuration between hardware components and software. Overall, this experiment served as a successful introduction to real-time embedded system control using ESP32.