

ESP32-WROOM-32 LED Blink Guide (38-pin) - Using PlatformIO

1. Objective

To blink three LEDs connected to an ESP32-WROOM-32 (38-pin version) in sequence using PlatformIO and monitor messages via the serial console.

2. Hardware Required

- ESP32-WROOM-32 (38-pin) Development Board
- 3 x LEDs (Red, Yellow, Green)
- 3 x 220-ohm resistors
- Breadboard
- Jumper wires

3. Circuit Diagram and Pin Connections

ESP32 Pin	Connects To	LED Color	Description
GPIO12	Anode (+) of Red LED	Red	Long leg of Red LED
GPIO14	Anode (+) of Yellow LED	Yellow	Long leg of Yellow LED
GPIO27	Anode (+) of Green LED	Green	Long leg of Green LED
GND	Through 220-ohm resistors	All LEDs	Cathode (-) legs connected to GND via resistor

Note: GPIO12 is a strapping pin; avoid pressing the BOOT button while powering up if using GPIO12.

4. Code for Sequential LED Blink

```
#include <Arduino.h>
#define led1 12
#define led2 14
#define led3 27

void setup() {
  // put your setup code here, to run once:
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  Serial.begin(115200);
}
```

```

void loop() {
    // put your main code here, to run repeatedly:
    Serial.println("Red LED Glows");
    digitalWrite(led1,HIGH);
    delay(1000);
    Serial.println("Red LED Stops");
    digitalWrite(led1,LOW);
    delay(1000);
    Serial.println("Yellow LED Glows");
    digitalWrite(led2,HIGH);
    delay(1000);
    Serial.println("Yellow LED Stops");
    digitalWrite(led2,LOW);
    delay(1000);
    Serial.println("Green LED Glows");
    digitalWrite(led3,HIGH);
    delay(1000);
    Serial.println("Green LED Stops");
    digitalWrite(led3,LOW);
    delay(1000);
}

```

5. Explanation of Code (Line by Line)

```
#include <Arduino.h>
```

Includes the Arduino core library necessary for using functions like `pinMode()`, `digitalWrite()`, and `delay()`.

```

#define led1 12
#define led2 14
#define led3 27

```

Defines symbolic names for the GPIO pins connected to Red, Yellow, and Green LEDs respectively. This makes code more readable and maintainable.

```

void setup() {
    // put your setup code here, to run once:
    pinMode(led1,OUTPUT);
    pinMode(led2,OUTPUT);
    pinMode(led3,OUTPUT);
    Serial.begin(115200);
}

```

- Sets each LED pin as an **output** so we can turn them on and off.
- Initializes **serial communication** at 115200 baud for debugging via Serial Monitor.

```
void loop() {
```

This function repeats **infinitely** after `setup()`. Each LED blinks one after the other with 1-second intervals and status messages are printed to the serial monitor.

```
Serial.println("Red LED Glows");
digitalWrite(led1, HIGH);
delay(1000);
```

- Turns on Red LED.
- Waits for 1 second.
- Prints a status message to serial monitor.

```
Serial.println("Red LED Stops");
digitalWrite(led1, LOW);
delay(1000);
```

- Turns off Red LED.
- Waits for 1 second.

Same logic applies for Yellow and Green LEDs.

6. PlatformIO Setup

1. Create a new PlatformIO project.
2. Select board: `esp32dev` or your specific ESP32 variant.
3. Ensure `platformio.ini` includes:

```
[env:upesy_wroom]
platform = espressif32
board = upesy_wroom
framework = arduino
```

4. Upload the code, open Serial Monitor, and observe LED sequence and messages.

7. Conclusion

This project demonstrates how to control multiple LEDs in sequence using the ESP32 and how to use the serial monitor for simple runtime feedback. This sets a foundation for building traffic light systems, status indicators, or learning multitasking with ESP32.

SERIAL MONITOR

Red LED Glows

Red LED Stops

Yellow LED Glows

Yellow LED Stops

Green LED Glows

Green LED Stops

