Pumo Technovation

Glow IoT

Title: Glow IoT – Smart Control System

Short description: Glow IoT is a smart lighting system controlled by ESP32 microcontroller via Blynk widgets over an ad-hoc network.

Key Features (bullet points): Remote ON/OFF control, brightness and color customization, ad-hoc Wi-Fi control, real-time monitoring.

Include a simple neat diagram showing: ESP32 → LEDs/Devices → Blynk App (smartphone).

Keep text minimal, professional, and readable. Use soft, clean colors and minimalistic icons.
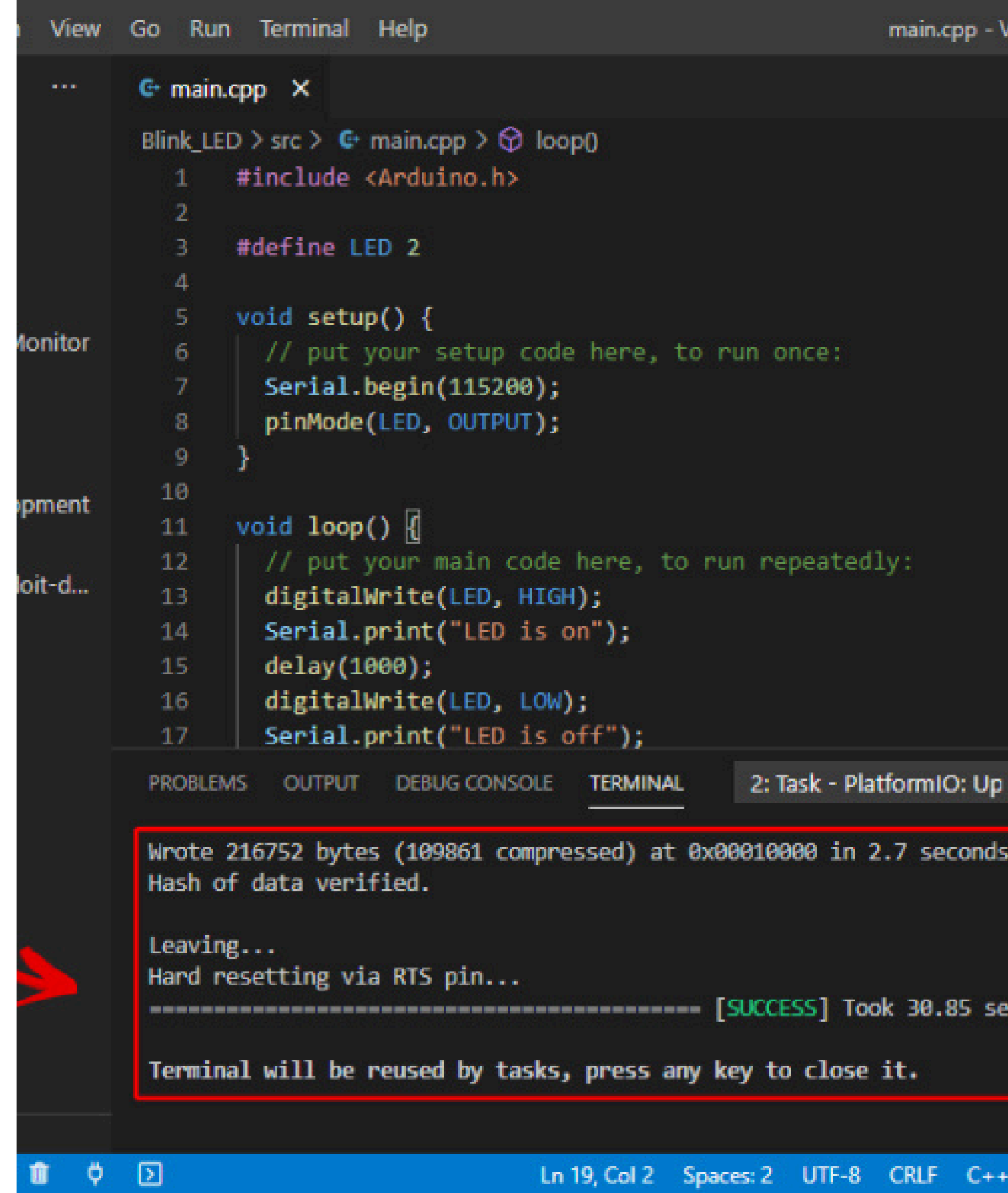
PRESENTED BY

Priyadharshini R

Coding & Implementation

The Glow IoT control system utilizes the ESP32 microcontroller, which integrates seamlessly with Blynk for remote management, allowing users to customize and monitor their smart lighting system efficiently.

Program

```
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
char auth[] = "YourBlynkAuthToken";
char ssid[] = "ESP32_AP";
char pass[] = "12345678";
void setup(){ Serial.begin(115200); Blynk.begin(auth, ssid, pass);
pinMode(2, OUTPUT);}
BLYNK_WRITE(V1){ digitalWrite(2, param.asInt()); }
void loop(){ Blynk.run(); }
```

Working Stage

- ESP32 Microcontroller → LED Device
- Show an arrow from ESP32 to LED to indicate control.
- Ad-hoc Wi-Fi Network
- Label it as "ESP32 creates direct Wi-Fi network for control."
- Blynk App on Smartphone → Remote Control
- Show Button Widget (V1) for ON/OFF
- Slider Widget (V2) for brightness control
- Live Action Flow:
- Press Blynk Button → LED turns ON/OFF immediately
- Slide Slider Widget → LED brightness changes in real-time
- Optional Notes:
- Highlight real-time response
- Show power efficiency and IoT automation
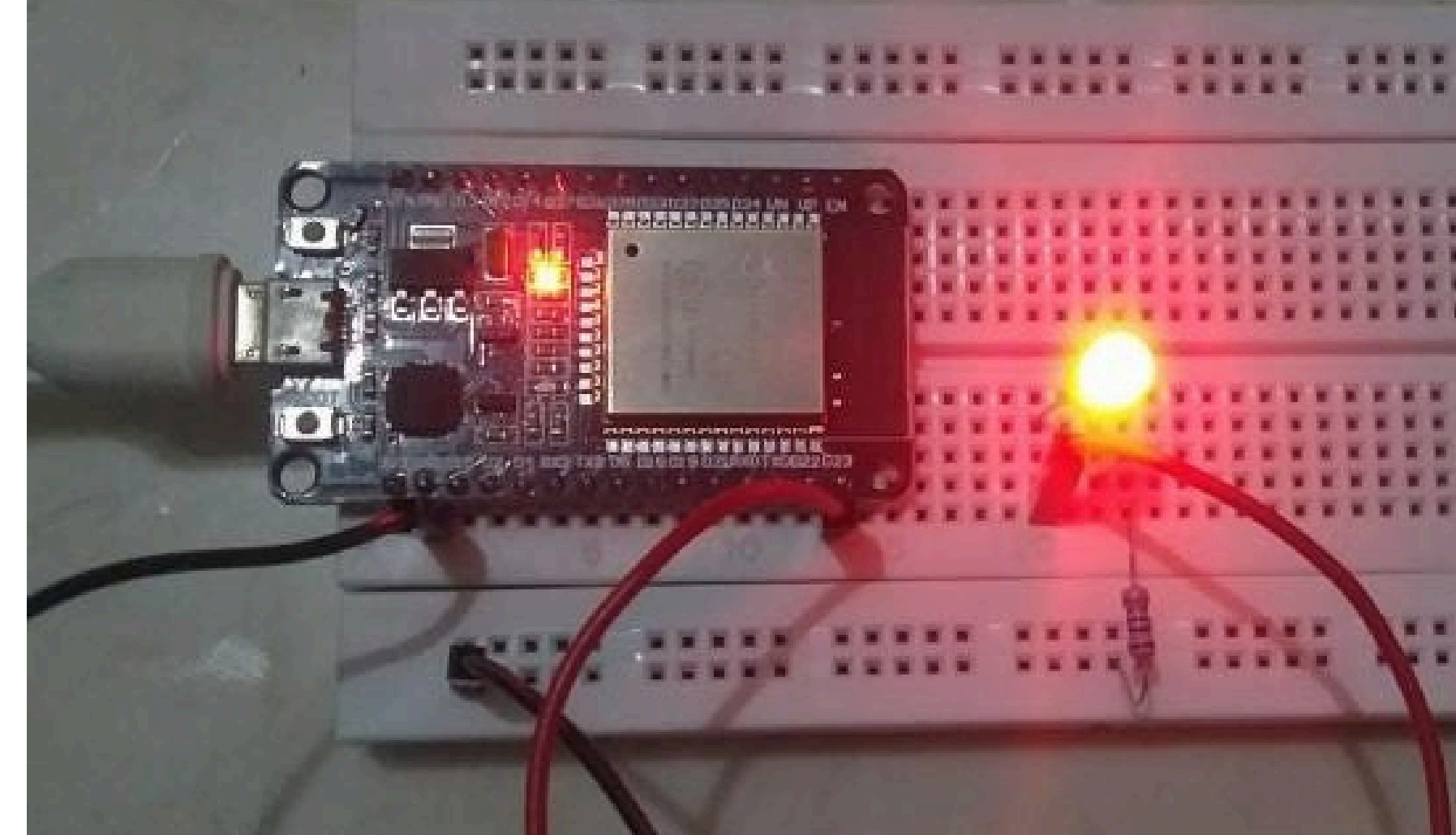
Internship Completion

Gained hands-on experience with ESP32 microcontroller and IoT projects.

Learned to program and deploy code in VSCode for embedded systems.

Implemented Blynk widgets for remote device control via ad-hoc networks.

Developed problem-solving and debugging skills in hardware-software integration.
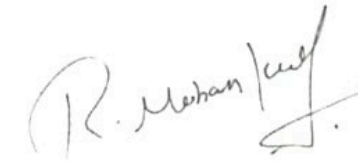
Improved teamwork, communication, and project presentation skills.