**FlexiGlow: Versatile Portable Study Tabletop with Intelligent Features**

**Project Summary**: This embedded system project is designed to create a smart control and monitoring solution using the ESP32 microcontroller at its core. The project integrates various peripherals including a DHT22 temperature-humidity sensor, an RTC DS3231 module, a DWIN HMI touch display, WS2812B RGB LED strip, a two-channel relay module for lamp control, and a MQ-135 gas sensor for air quality monitoring. Additionally, an ESP8266 microcontroller is used as a slave module for controlling the RGB LED strip. The entire system is orchestrated via the DWIN HMI display, which acts as the main control interface, and Bluetooth is used for updating tasks or sending commands. Component-wise

Explanation: 1. **ESP32 Microcontroller –**

**Role**: Acts as the central controller for the entire project. - Why Used: Offers dual-core processing, multiple UARTs, analog input capability, Wi-Fi/Bluetooth connectivity - perfect for handling multiple peripherals simultaneously. - How Used: Interfaces with all sensors, controls relays, manages communication with the ESP8266 and the HMI, and processes analog/digital inputs.

2. **RTC DS3231 (Real-Time Clock) –**

**Role**: Provides accurate real-time data (date and time). - Why Used: Essential for time-based control or data logging tasks, especially when Wi-Fi-based NTP is not reliable or available. - How Used: Connected to the ESP32 via I2C (SCL/SDA), used for timestamping events and potentially for scheduling lighting tasks

. 3. DHT22 (Temperature and Humidity Sensor) - Role: Monitors environmental conditions. - Why Used: Precise and stable sensor for measuring ambient temperature and humidity. - How Used: Connected to GPIO 32 of the ESP32 and periodically read to display values on the HMI or for triggering events based on conditions

.

4. **MQ-135 Gas Sensor** –

**Role:** Detects CO2 and other harmful gases in the air. - Why Used: Air quality monitoring is essential in smart environments for health and safety. - How Used: Connected to analog pin 34 of the ESP32. Analog readings are interpreted to estimate CO2 levels.

5. **DWIN HMI Display –**

**Role**: Acts as the user interface for monitoring and controlling the system. - Why Used: Advanced GUI interface for direct user interaction without needing external devices like mobile apps. - How Used: - Two UART ports are used: - UART1 (pins 16 & 17) for transmission - UART2 (pins 25 & 26) for reception - The display sends commands to the ESP32 (e.g., to control relays or LED patterns). - Also acts as a gateway for Bluetooth task updates.

6. **Two-Channel Relay Module (for Lamp Control) -** Role: Controls the lighting pattern inside a tampered lamp. - Why Used: Unique lamp design allows pattern control by grounding specific internal points. - How Used: - Relay 1 connects point A to GND. - Relay 2 connects point B to GND. - Controlled using GPIO 18 and 19. - Different combinations of A and B being grounded result in three uniqu

e light patterns.

7. **ESP8266 Microcontroller** - Role: Acts as a secondary controller to manage the RGB LED strip. - Why Used: Offloads the RGB LED handling from the ESP32 and keeps timing-sensitive LED updates isolated. - How Used: - Connected to ESP32 on GPIO 33 and to D4 of the ESP8266. - The connection simulates a button press using a pulled-up line - a pulse from the ESP32 is counted by the ESP8266 to switch RGB patterns. - Each pulse triggers a pattern change on the WS2812B LED strip.

8. **WS2812B RGB LED Strip (120 LEDs) - Role**: Provides programmable RGB lighting. - Why Used: Visually rich feedback with programmable colors and animations. - How Used: - Controlled by the ESP8266. - A total of 4 different RGB lighting patterns can be toggled based on pulses from ESP32.

9. **Communication Between ESP32 and ESP8266 - Role**: Sends control signals to toggle RGB LED patterns. - Why Used: Keeps RGB control logic isolated and lightweight. - How Used: - GPIO 33 of ESP32 sends a brief LOW pulse to D4 of ESP8266. - A count is maintained on ESP8266, and each pulse changes the lighting mode.

**System Overview**: - The DWIN HMI is the command center. It allows the user to send commands via touch or Bluetooth. - ESP32 gathers data from DHT22, MQ-135, and RTC, processes user input from the HMI, and controls: - Relays to change lamp light patterns. - ESP8266 to toggle RGB LED modes. - ESP8266, upon receiving a signal, cycles the WS2812B strip through four preset lighting modes. - Sensor data and status feedback are shown on the HMI, making the system both interactive and autonomous.

**Features**: - Real-time environmental monitoring (Temperature, Humidity, CO2). - Tampered lamp pattern control via smart relay shorting. - RGB light customization with multiple modes. - Centralized user control via touch screen and Bluetooth. - Expandable and customizable via UART and GPIO.