

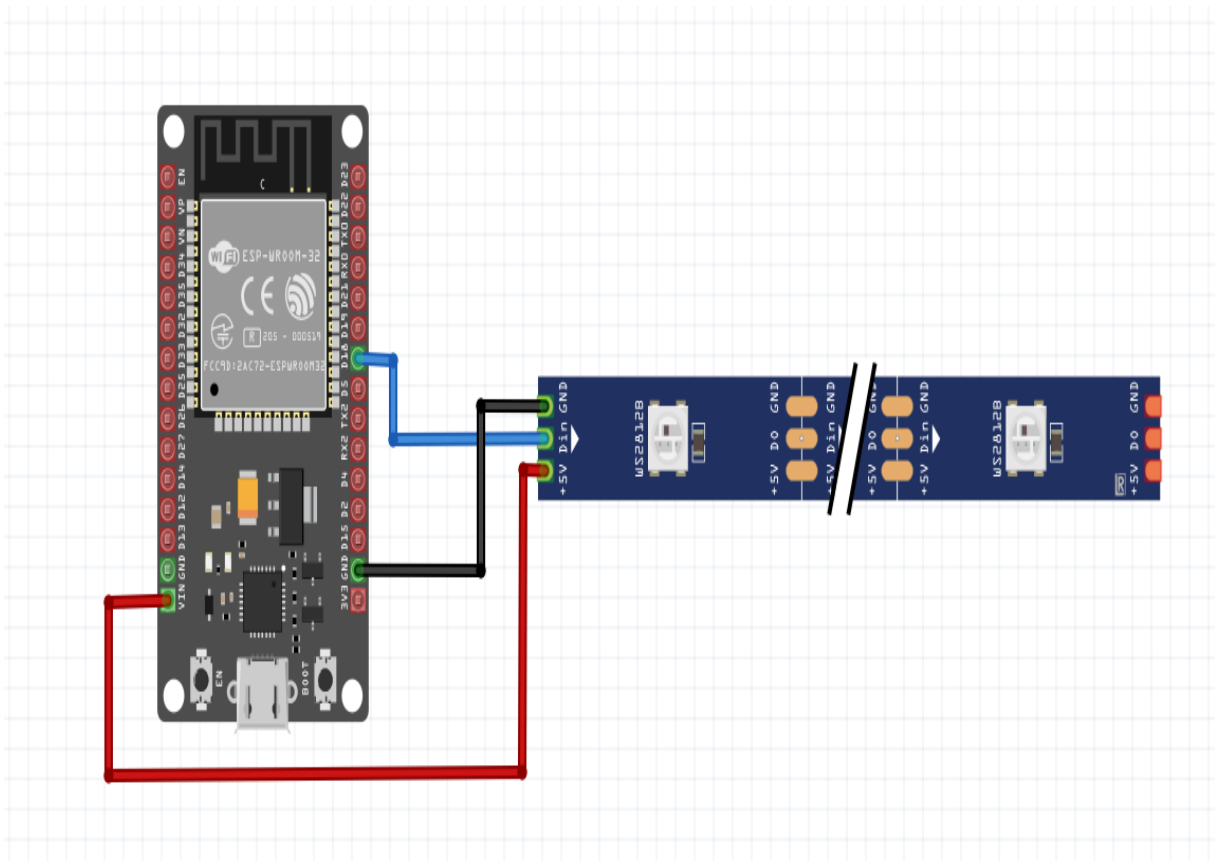
SMART LAMP USING ESP32

Objective: The objective of this project is to create a Bluetooth-controlled smart RGB LED system using an ESP32, allowing users to remotely change the color and brightness of the LEDs through a mobile application.

Components: ESP32, Micro USB Cable, RGB Led Strip, Jumper wires.

Software Requirements: Arduino IDE.

Wiring Diagram:



Procedure:

- Gather your components: an ESP32 board, a 3-pin RGB LED strip, jumper wires, and a USB cable for powering the ESP32.
- Connect the 5V pin of the LED strip to the 5V pin on the ESP32 to supply power to the LED strip.
- Connect the GND pin of the LED strip to the GND pin on the ESP32 to establish a common ground.
- Connect the Data pin of the LED strip to GPIO 18 (D18) on the ESP32. This pin will control the colors of the LEDs.
- Once the wiring is complete, open the Arduino IDE or another compatible development environment for the ESP32.
- Install the required libraries: Adafruit NeoPixel for LED control and BluetoothSerial for Bluetooth communication.
- Paste the provided code into the IDE and upload it to the ESP32, selecting the correct board and port.
- After uploading, pair your mobile phone with the ESP32 via Bluetooth, and use a mobile app to send RGB values or ON/OFF commands to control the LED strip colors.
- Finally, power the ESP32 through a USB cable to power both the ESP32 and the LED strip.

Troubleshooting points:

LED Strip Not Lighting Up:

- **Check power supply:** Ensure the LED strip is properly powered (5V). If your strip requires more power, use an external power source instead of directly powering it from the ESP32.
- **Check connections:** Make sure the Data pin of the LED strip is connected to the correct GPIO (18 in this case).
- **Check ground:** Ensure that the ground (GND) of the ESP32 is connected to the ground of the LED strip and power supply.
- **Verify code:** Ensure that the number of LEDs defined in the code matches the actual number of LEDs on the strip.

Bluetooth Not Connecting:

- **Enable Bluetooth:** Make sure Bluetooth is enabled on your phone and that you're connecting to the correct device name ("ESP32_NeoPixel_Control").
- **Device pairing:** Sometimes, clearing the Bluetooth device from your phone's settings and pairing again can help.
- **Check serial monitor:** Use the serial monitor to view debug messages. It will show whether Bluetooth communication is being established.

LED Colors Not Changing Properly:

- **Check color format:** Ensure that the app is sending the correct RGB format as expected (e.g., "255.0.0)" for red).
- **Brightness issues:** If colors appear dim, check if the brightness is set correctly in the code (strip.setBrightness(255)).
- **Incorrect RGB order:** If colors appear wrong (e.g., green instead of red), the RGB order might be incorrect. Change NEO_GRB to another order (e.g., NEO_RGB) depending on your strip.

Bluetooth Delays or Slow Responses:

- **Signal interference:** Bluetooth performance might be affected by interference from other devices. Try moving away from other Bluetooth devices.
- **Connection distance:** Make sure your mobile device is within range of the ESP32 (typically around 10 meters).

Output :

- Upon powering the ESP32, the RGB LED strip will initialize with the default brightness and color (white).
- Pair your mobile phone with the ESP32 over Bluetooth to send commands from the app.
- You can send RGB color values (e.g., 255.0.0 for red) through Bluetooth, and the LED strip will change color accordingly.
- Sending the "ON" command will turn the LEDs on with the last selected color.
- Sending the "OFF" command will turn off the entire strip, leaving no lights on.



Applications :

- Bluetooth-controlled smart lighting systems for homes or offices.
- Decorative RGB lighting for events, parties, or holidays.
- Ambient lighting for gaming setups or media centers.
- DIY lighting projects for hobbyists.

Future Enhancements:

- **Wi-Fi Control:** Add Wi-Fi to control the LEDs from anywhere via the internet.
- **Voice Control:** Integrate with voice assistants like Alexa for hands-free control.
- **Music Sync:** Make the LEDs change color based on the rhythm of the music.
- **Custom Mobile App:** Build a dedicated app for more control, like animations and timers.
- **Multi-Zone Lighting:** Control different parts of the LED strip independently for more effects.
- **Motion Sensors:** Turn on or change lights when movement is detected

Project By :

Nagiri Sai Akshaya,
BTech(IoT),
ACE Engineering College.