**Smart LED Controller with ESP32 - Exploring Advanced Fundamentals of Electricity**

**Objective:**

This practical aims to deepen participants’ understanding of electricity by guiding them through the creation of a smart LED controller using an ESP32 microcontroller. Throughout this hands-on experience, participants will delve into fundamental concepts such as voltage, current, and the implementation of microcontroller-based control systems.

**Components Needed:**

* ESP32 Development Board
* RGB LED (Common Cathode)
* Resistors: 220 ohms (3), 10k ohms (1)
* Breadboard and Jumper Wires
* Smartphone or Computer with Bluetooth Capability

**Circuit Design:**

* Connect the RGB LED's common cathode to the ESP32 ground (GND).
* Connect each color (red, green, blue) of the RGB LED to individual GPIO pins on the ESP32.
* Connect a 220-ohm resistor in series with each color of the RGB LED.
* Connect a 10k-ohm resistor from one GPIO pin to the ESP32 3.3V (for button input).

**Arduino IDE Code:**

#include <BLEDevice.h>

#include <BLEUtils.h>

#include <BLEServer.h>

BLECharacteristic \*pCharacteristic;

int redPin = 13; // GPIO pin connected to the red color of the RGB LED

int greenPin = 14; // GPIO pin connected to the green color of the RGB LED

int bluePin = 15; // GPIO pin connected to the blue color of the RGB LED

int buttonPin = 2; // GPIO pin connected to the button

bool ledState = false;

class MyServerCallbacks : public BLEServerCallbacks {

void onConnect(BLEServer\* pServer) {

// Code to run when connected

};

void onDisconnect(BLEServer\* pServer) {

// Code to run when disconnected

}

};

void setup() {

Serial.begin(115200);

pinMode(redPin, OUTPUT);

pinMode(greenPin, OUTPUT);

pinMode(bluePin, OUTPUT);

pinMode(buttonPin, INPUT);

BLEDevice::init("Smart\_LED\_Controller");

BLEServer \*pServer = BLEDevice::createServer();

pServer->setCallbacks(new MyServerCallbacks());

BLEService \*pService = pServer->createService(BLEUUID((uint16\_t)0x180F));

pCharacteristic = pService->createCharacteristic(

BLEUUID((uint16\_t)0x2A19),

BLECharacteristic::PROPERTY\_READ

);

pService->start();

BLEAdvertising \*pAdvertising = BLEDevice::getAdvertising();

pAdvertising->setScanResponse(true);

pAdvertising->setMinPreferred(0x06);

pAdvertising->setMinPreferred(0x12);

BLEDevice::startAdvertising();

}

void loop() {

if (digitalRead(buttonPin) == HIGH) {

ledState = !ledState;

delay(500);

}

analogWrite(redPin, ledState ? 255 : 0);

analogWrite(greenPin, ledState ? 255 : 0);

analogWrite(bluePin, ledState ? 255 : 0);

delay(50);

}

**Procedure:**

* Upload the provided code to your ESP32 using the Arduino IDE.
* Install a Bluetooth Low Energy (BLE) scanner app on your smartphone or computer.
* Connect to the ESP32 via BLE and observe the RGB LED changing colors based on button presses.

**Voltage and Current Requirements for RGB LED Control:**

The RGB LED used in this smart LED controller project requires careful consideration of voltage and current for proper operation. Each color (red, green, and blue) in the RGB LED is controlled independently, and they typically have specific forward voltage (Vf) and forward current (If) requirements. It's crucial to meet these specifications to ensure optimal brightness and color accuracy.

**Forward Voltage (Vf):** The forward voltage is the voltage required for the LED to emit light. It varies for different LED colors and is usually provided by the LED manufacturer. In this project, the ESP32's GPIO pins provide the necessary voltage to illuminate the RGB LED.

**Forward Current (If):** The forward current is the current flowing through the LED when it is forward-biased. It is essential to limit the current to prevent damage to the LED. The 220-ohm resistors in series with each color help regulate the current, ensuring safe operation.

**Microcontroller-Based Control:**

The ESP32 microcontroller serves as the brain of the smart LED controller, enabling precise and dynamic control of the RGB LED. The control mechanism involves pulse-width modulation (PWM) to adjust the brightness of each LED color independently.

**Pulse-Width Modulation (PWM):** The ESP32 utilizes PWM to control the output voltage to the RGB LED. By rapidly switching the GPIO pins between high and low states, it simulates varying levels of voltage. This dynamic control allows for smooth color transitions and brightness adjustments.

**Wireless Control using Bluetooth:** The ESP32 communicates wirelessly with external devices, such as smartphones or computers, through Bluetooth Low Energy (BLE). This wireless capability enables users to control the RGB LED remotely, enhancing user interaction and making the smart LED controller versatile and convenient.

**BLE Communication:**

Bluetooth Low Energy (BLE) plays a pivotal role in establishing communication between the ESP32 and external devices, such as smartphones or computers.

**BLE Server-Client Architecture:** The ESP32 operates as a BLE server, providing services and characteristics that can be accessed by a BLE client (smartphone or computer). The server advertises its presence, allowing clients to connect

**BLE Characteristics:** In this project, a BLE characteristic is created to handle the reading of data. The ESP32's Bluetooth service is initialized with a unique UUID, and the characteristic within that service is defined. This characteristic can be read by the BLE client to exchange data.

**Connectivity and Interactivity:** BLE facilitates a seamless connection between the ESP32 and external devices, allowing users to interact with the smart LED controller wirelessly. Users can send commands or data via a BLE-enabled application, influencing the behavior of the RGB LED.

**Conclusion:**

This practical not only covers fundamental concepts of electricity but also introduces participants to advanced microcontroller-based control systems. By creating a smart LED controller with Bluetooth capabilities, participants gain insights into both basic and sophisticated aspects of electronics. This hands-on experience fosters a deeper understanding of practical applications and paves the way for further exploration in the field of electronics.