**Tap-Based Smart LED System**

**📘 Project Title:**

**Tap-Based Smart LED System with Serial and Blynk Monitoring**

**🎯 Objective:**

To control two LEDs (white and blue) using tap gestures detected via a touch sensor, where each specific tap count triggers LED actions. The LED states are shown on the Serial Monitor and updated in real time on the Blynk dashboard. A buzzer gives audible feedback when LEDs are toggled.

**🧾 Components Required:**

| **Component Name** | **Pin No (ESP32)** | **Destination Component** | **Pin No** | **Special Remark** |
| --- | --- | --- | --- | --- |
| ESP32 Dev Board | - | - | - | WiFi-enabled MCU |
| TTP223 Touch Sensor | GPIO 4 | ESP32 | D4 | Active LOW output |
| White LED | GPIO 23 | ESP32 | D23 | Turns ON at 1 tap, OFF at 2 taps |
| Blue LED | GPIO 22 | ESP32 | D22 | Turns ON at 3 taps, OFF at 4 taps |
| Buzzer | GPIO 21 | ESP32 | D21 | Beeps once on each LED toggle |
| Blynk Cloud | V0, V1 | ESP32 Virtual Pins | - | Shows LED states as ON/OFF |
| Serial Monitor | USB | Computer | - | Displays real-time LED status |

**🔌 Circuit Connections:**

* **TTP223 OUT → ESP32 GPIO 4**
* **White LED anode → GPIO 23 → 220Ω resistor → GND**
* **Blue LED anode → GPIO 22 → 220Ω resistor → GND**
* **Buzzer + → GPIO 21**, **Buzzer – → GND**
* **VCC & GND of all components → ESP32 3.3V & GND respectively**

**📲 Blynk Dashboard Setup:**

1. Go to [https://blynk.cloud](https://blynk.cloud/) and log in.
2. Create a new template:
   * **Name:** Tap-Based Smart LED
   * **Board:** ESP32 Dev Board
   * **Connection Type:** WiFi
3. Get your:
   * BLYNK\_TEMPLATE\_ID
   * BLYNK\_TEMPLATE\_NAME
   * BLYNK\_AUTH\_TOKEN
4. Add two **Value Display widgets**:
   * **V0** → White LED status
   * **V1** → Blue LED status

**🧠 Tap Logic:**

| **Tap Count** | **Action** | **Affected Component** |
| --- | --- | --- |
| 1 Tap | Turn ON White LED | WHITE\_LED |
| 2 Taps | Turn OFF White LED | WHITE\_LED |
| 3 Taps | Turn ON Blue LED | BLUE\_LED |
| 4 Taps | Turn OFF Blue LED | BLUE\_LED |
| >4 Taps | Ignored | None |

**🧾 Final Code:**

#define BLYNK\_TEMPLATE\_ID "TMPL3EPcy8UyB"

#define BLYNK\_TEMPLATE\_NAME "Tap Based Smart LED"

#define BLYNK\_AUTH\_TOKEN "nzxu\_3Z1ILc3JIDH6cTf8XVYHaSDxOgK"

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

// Pin Definitions

#define TOUCH\_PIN 4

#define WHITE\_LED 23

#define BLUE\_LED 22

#define BUZZER 21

// WiFi Credentials

char ssid[] = "Student";

char pass[] = "Learn@123";

// Tap Handling Variables

int tapCount = 0;

unsigned long lastTapTime = 0;

unsigned long tapTimeout = 600;

bool whiteLEDState = false;

bool blueLEDState = false;

void setup() {

Serial.begin(115200);

Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, pass);

pinMode(WHITE\_LED, OUTPUT);

pinMode(BLUE\_LED, OUTPUT);

pinMode(BUZZER, OUTPUT);

pinMode(TOUCH\_PIN, INPUT);

whiteLEDState = false;

blueLEDState = false;

digitalWrite(WHITE\_LED, LOW);

digitalWrite(BLUE\_LED, LOW);

digitalWrite(BUZZER, LOW);

updateStatesToBlynk(); // Send initial states to Blynk

Serial.println("Tap-Based Smart LED System Initialized");

}

// Sync with Blynk after connection

BLYNK\_CONNECTED() {

updateStatesToBlynk();

}

void loop() {

Blynk.run();

static bool lastTouchState = HIGH;

bool currentTouchState = digitalRead(TOUCH\_PIN);

if (currentTouchState == LOW && lastTouchState == HIGH) {

if (millis() - lastTapTime > 50) {

tapCount++;

lastTapTime = millis();

}

}

lastTouchState = currentTouchState;

if (tapCount > 0 && millis() - lastTapTime > tapTimeout) {

handleTaps(tapCount);

tapCount = 0;

}

}

void handleTaps(int count) {

Serial.print("Taps Detected: ");

Serial.println(count);

switch (count) {

case 1:

whiteLEDState = true;

digitalWrite(WHITE\_LED, HIGH);

beep();

Serial.println("White LED ON");

break;

case 2:

whiteLEDState = false;

digitalWrite(WHITE\_LED, LOW);

beep();

Serial.println("White LED OFF");

break;

case 3:

blueLEDState = true;

digitalWrite(BLUE\_LED, HIGH);

beep();

Serial.println("Blue LED ON");

break;

case 4:

blueLEDState = false;

digitalWrite(BLUE\_LED, LOW);

beep();

Serial.println("Blue LED OFF");

break;

default:

Serial.println("No defined action for this tap count");

break;

}

updateStatesToBlynk();

printStatusToSerial();

}

void beep() {

digitalWrite(BUZZER, HIGH);

delay(100);

digitalWrite(BUZZER, LOW);

}

void updateStatesToBlynk() {

Blynk.virtualWrite(V0, whiteLEDState ? "ON" : "OFF");

Blynk.virtualWrite(V1, blueLEDState ? "ON" : "OFF");

}

void printStatusToSerial() {

Serial.print("White LED: ");

Serial.print(whiteLEDState ? "ON" : "OFF");

Serial.print(" | Blue LED: ");

Serial.println(blueLEDState ? "ON" : "OFF");

}

**Code-2(optional)**

#define BLYNK\_TEMPLATE\_ID "TMPL3EPcy8UyB"

#define BLYNK\_TEMPLATE\_NAME "Tap Based Smart LED"

#define BLYNK\_AUTH\_TOKEN "nzxu\_3Z1ILc3JIDH6cTf8XVYHaSDxOgK"

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

#define TOUCH\_PIN     4

#define WHITE\_LED     23

#define BLUE\_LED      22

#define BUZZER        21

char ssid[] = "Student";

char pass[] = "Learn@123";

// States

int tapCount = 0;

unsigned long lastTapTime = 0;

unsigned long tapTimeout = 600;

bool whiteLEDState = false;

bool blueLEDState = false;

bool blinkMode = false;

unsigned long lastBlinkTime = 0;

bool blinkState = false;

void setup() {

  Serial.begin(115200);

  Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, pass);

  pinMode(WHITE\_LED, OUTPUT);

  pinMode(BLUE\_LED, OUTPUT);

  pinMode(BUZZER, OUTPUT);

  pinMode(TOUCH\_PIN, INPUT);

  digitalWrite(WHITE\_LED, LOW);

  digitalWrite(BLUE\_LED, LOW);

  digitalWrite(BUZZER, LOW);

  updateStatesToBlynk();

  Serial.println("Tap-Based Smart LED System Initialized");

}

BLYNK\_CONNECTED() {

  updateStatesToBlynk();

}

void loop() {

  Blynk.run();

  // Handle blinking mode

  if (blinkMode && millis() - lastBlinkTime >= 500) {

    lastBlinkTime = millis();

    blinkState = !blinkState;

    digitalWrite(WHITE\_LED, blinkState);

    digitalWrite(BLUE\_LED, blinkState);

  }

  // Tap detection

  static bool lastTouchState = HIGH;

  bool currentTouchState = digitalRead(TOUCH\_PIN);

  if (currentTouchState == LOW && lastTouchState == HIGH) {

    if (millis() - lastTapTime > 50) {

      tapCount++;

      lastTapTime = millis();

    }

  }

  lastTouchState = currentTouchState;

  if (tapCount > 0 && millis() - lastTapTime > tapTimeout) {

    handleTaps(tapCount);

    tapCount = 0;

  }

}

void handleTaps(int count) {

  Serial.print("Taps Detected: ");

  Serial.println(count);

  switch (count) {

    case 1:

      whiteLEDState = true;

      blinkMode = false;

      updateOutputs();

      beep();

      Serial.println("White LED ON");

      break;

    case 2:

      whiteLEDState = false;

      blinkMode = false;

      updateOutputs();

      beep();

      Serial.println("White LED OFF");

      break;

    case 3:

      blueLEDState = true;

      blinkMode = false;

      updateOutputs();

      beep();

      Serial.println("Blue LED ON");

      break;

    case 4:

      blueLEDState = false;

      blinkMode = false;

      updateOutputs();

      beep();

      Serial.println("Blue LED OFF");

      break;

    case 5:

      blinkMode = true;

      whiteLEDState = true;

      blueLEDState = true;

      Serial.println("Blinking Both LEDs");

      beep();

      break;

    case 6:

      blinkMode = false;

      whiteLEDState = false;

      blueLEDState = false;

      updateOutputs();

      beep();

      Serial.println("Blinking Mode OFF, Both LEDs OFF");

      break;

    default:

      Serial.println("No defined action for this tap count");

      break;

  }

  updateStatesToBlynk();

  printStatusToSerial();

}

void updateOutputs() {

  digitalWrite(WHITE\_LED, whiteLEDState ? HIGH : LOW);

  digitalWrite(BLUE\_LED, blueLEDState ? HIGH : LOW);

}

void beep() {

  digitalWrite(BUZZER, HIGH);

  delay(100);

  digitalWrite(BUZZER, LOW);

}

void updateStatesToBlynk() {

  Blynk.virtualWrite(V0, whiteLEDState ? (blinkMode ? "Blinking" : "ON") : "OFF");

  Blynk.virtualWrite(V1, blueLEDState ? (blinkMode ? "Blinking" : "ON") : "OFF");

}

void printStatusToSerial() {

  Serial.print("White LED: ");

  Serial.print(blinkMode ? "Blinking" : (whiteLEDState ? "ON" : "OFF"));

  Serial.print(" | Blue LED: ");

  Serial.println(blinkMode ? "Blinking" : (blueLEDState ? "ON" : "OFF"));

}

**✅ Expected Serial Monitor Output Example:**

Tap-Based Smart LED System Initialized

Taps Detected: 1

White LED ON

White LED: ON | Blue LED: OFF

Taps Detected: 2

White LED OFF

White LED: OFF | Blue LED: OFF

Taps Detected: 3

Blue LED ON

White LED: OFF | Blue LED: ON

Taps Detected: 4

Blue LED OFF

White LED: OFF | Blue LED: OFF