ESP32 Blynk + BLE Car Control Code

```
/*************** BLYNK SETUP ***************/
#define BLYNK_TEMPLATE_ID   "TMPL3GNtZJ5Jo"
#define BLYNK_TEMPLATE_NAME "CAR"
#define BLYNK_AUTH_TOKEN    "gNwkHgO6ERswJIEGG3QITo8xIbdtJpi8"

#define BLYNK_NO_FLOAT   // reduce flash usage

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include "BluetoothSerial.h"

/*************** WIFI CREDENTIALS ***************/
char ssid[] = "PARUVATHAM_2.4Ghz";
char pass[] = "9500027179";

/*************** BLE SETUP ***************/
BluetoothSerial SerialBT;

/*************** MOTOR DRIVER PINS ***************/
#define IN1 14
#define IN2 27
#define IN3 26
#define IN4 25

/*************** EXTRA PINS ***************/
#define BATTERY_PIN   34
#define BUZZER_PIN    32
#define HEADLIGHT_PIN 33
#define MODE_SWITCH   13   // LOW = WiFi, HIGH = BLE (inverted logic)

unsigned long lastUpdate = 0;
bool wifiMode = false;

/*************** BUTTON STATES ***************/
bool fwdState = false;
bool backState = false;
bool leftState = false;
bool rightState = false;

/*************** MOTOR FUNCTIONS ***************/
void stopCar() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}
void forward() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
void backward() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}
void left() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}
```

```
void right() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

/*************** BLYNK BUTTON HANDLERS ***************/
BLYNK_WRITE(V1) { fwdState   = param.asInt(); }
BLYNK_WRITE(V2) { backState  = param.asInt(); }
BLYNK_WRITE(V3) { leftState  = param.asInt(); }
BLYNK_WRITE(V4) { rightState = param.asInt(); }

BLYNK_WRITE(V7) { digitalWrite(BUZZER_PIN, param.asInt()); }
BLYNK_WRITE(V8) { digitalWrite(HEADLIGHT_PIN, param.asInt()); }

/*************** BATTERY % CALC ***************/
int getBatteryPercent() {
  int raw = analogRead(BATTERY_PIN);
  float voltage = (raw / 4095.0) * 25.0;  // ADC scaling (0–25V)

  // Map: 4V -> 0%, 11V -> 100%
  int percent = map((int)(voltage * 100), 400, 1100, 0, 100);
  if (percent > 100) percent = 100;
  if (percent < 0) percent = 0;
  return percent;
}

/*************** SETUP ***************/
void setup() {
  Serial.begin(115200);

  pinMode(IN1, OUTPUT); pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT); pinMode(IN4, OUTPUT);
  stopCar();

  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(HEADLIGHT_PIN, OUTPUT);
  digitalWrite(BUZZER_PIN, LOW);
  digitalWrite(HEADLIGHT_PIN, LOW);

  pinMode(MODE_SWITCH, INPUT);

  // ■ Inverted logic
  wifiMode = !digitalRead(MODE_SWITCH);

  if (wifiMode) {
    Serial.println("Starting in WiFi Mode...");
    WiFi.begin(ssid, pass);
    while (WiFi.status() != WL_CONNECTED) { delay(500); Serial.print("."); }
    Serial.println("\nWiFi Connected");
    Blynk.config(BLYNK_AUTH_TOKEN);
    while (!Blynk.connect()) { delay(500); Serial.print("B"); }
    Serial.println("\nBlynk Connected");
  } else {
    Serial.println("Starting in BLE Mode...");
    stopCar();                  // stop motors before BLE
    WiFi.disconnect(true);      // disable WiFi completely
    WiFi.mode(WIFI_OFF);
    delay(200);
    SerialBT.begin("Taran_Car");  // BLE name
    Serial.println("BLE Ready: Taran_Car");
  }
}

/*************** LOOP ***************/
void loop() {
  if (wifiMode) {
    Blynk.run();

    // Decide motor state instantly
    if (fwdState) forward();
    else if (backState) backward();
    else if (leftState) left();
    else if (rightState) right();
    else stopCar();

    // Battery + WiFi update every 30s
    if (millis() - lastUpdate >= 30000) {
      lastUpdate = millis();
      int batt = getBatteryPercent();
      int rssi = WiFi.RSSI();
      Blynk.virtualWrite(V5, batt);
      Blynk.virtualWrite(V6, rssi);
    }
  } else {
    if (SerialBT.available()) {
      char c = SerialBT.read();
      if (c == 'F' || c == 'f') forward();
      else if (c == 'B' || c == 'b') backward();
      else if (c == 'L' || c == 'l') left();
      else if (c == 'R' || c == 'r') right();
      else if (c == 'S' || c == 's') stopCar();
    }
  }
}
```