#define BLYNK\_TEMPLATE\_ID "TMPL3jPQ\_zAmR"

#define BLYNK\_TEMPLATE\_NAME "Curtain Control"

#define BLYNK\_AUTH\_TOKEN "V4r9Qs5BmGkTyC3a3qTrYK96aTqTEcY0"

#include <WiFi.h>

#include <HTTPClient.h>

#include <BlynkSimpleEsp32.h>

const char\* ssid = "Wokwi-GUEST";

const char\* password = "";

const int LDR\_PIN = 18;      // Analog input

const int LED\_PIN = 2;       // Output for curtain simulation

int lightThresholdOpen = 500;

int lightThresholdClose = 200;

bool curtainOpen = false;

bool manualOverride = false;

int simulatedLight = -1; // -1 = disabled

bool useSimulatedLight = false;

BlynkTimer timer;

// Receive manual curtain command (V0)

BLYNK\_WRITE(V0) {

  int cmd = param.asInt();

  if (manualOverride) {

    if (cmd == 1) openCurtains();

    else closeCurtains();

  }

}

// Toggle manual override mode (V2)

BLYNK\_WRITE(V2) {

  manualOverride = param.asInt();

**Serial**.println(manualOverride ? "Manual override ON" : "Auto mode ON");

}

BLYNK\_WRITE(V3) {

  simulatedLight = param.asInt();

  useSimulatedLight = true;

}

void setup() {

**Serial**.begin(115200);

  delay(1000);

  pinMode(LED\_PIN, OUTPUT);

  digitalWrite(LED\_PIN, LOW);

  WiFi.begin(ssid, password);

**Serial**.print("Connecting to WiFi");

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

**Serial**.print(".");

  }

**Serial**.println("\nConnected to WiFi");

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

  timer.setInterval(5000L, checkLightSensor);  // Check every 5 seconds

}

void loop() {

  Blynk.run();

  timer.run();

}

void checkLightSensor() {

  int lightValue = useSimulatedLight ? simulatedLight : analogRead(LDR\_PIN);

**Serial**.println("Light Level: " + String(lightValue));

  Blynk.virtualWrite(V1, lightValue);  // Send LDR value to app

  if (!manualOverride) {

    if (lightValue > lightThresholdOpen && !curtainOpen) {

      openCurtains();

    } else if (lightValue < lightThresholdClose && curtainOpen) {

      closeCurtains();

    }

  }

}

void openCurtains() {

  curtainOpen = true;

**Serial**.println("Curtains closed.");

  digitalWrite(LED\_PIN, HIGH);  // LED ON

  Blynk.virtualWrite(V0, 1);    // Reflect state in app

}

void closeCurtains() {

  curtainOpen = false;

**Serial**.println("Curtains opened.");

  digitalWrite(LED\_PIN, LOW);   // LED OFF

  Blynk.virtualWrite(V0, 0);    // Reflect state in app

}