**Hands Free Home Automation**

#define BLYNK\_PRINT Serial

#define BLYNK\_TEMPLATE\_ID "TMPL3N9NRPUZg"

#define BLYNK\_TEMPLATE\_NAME "Home Automation"

#define BLYNK\_AUTH\_TOKEN "hkPdzXdkHkiqiIEg4fuYWFb65IZIRXjL"

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

// Wi-Fi credentials

char ssid[] = "vijay";

char pass[] = "annayya1";

// Relay pin definitions

#define FAN\_RELAY 16 // GPIO16

#define LIGHT\_RELAY 17 // GPIO17

#define TV\_RELAY 18 // GPIO18

#define COOLER\_RELAY 19 // GPIO19

// VC-02 UART pins

#define VC02\_RX 21 // ESP32 RX (connect to VC02 TX)

#define VC02\_TX 22 // ESP32 TX (connect to VC02 RX)

// Set relay (Active LOW)

void setRelay(int pin, bool on) {

digitalWrite(pin, on ? LOW : HIGH);

}

// Blynk virtual pin handlers

BLYNK\_WRITE(V0) { setRelay(FAN\_RELAY, param.asInt()); }

BLYNK\_WRITE(V1) { setRelay(LIGHT\_RELAY, param.asInt()); }

BLYNK\_WRITE(V2) { setRelay(TV\_RELAY, param.asInt()); }

BLYNK\_WRITE(V3) { setRelay(COOLER\_RELAY, param.asInt()); }

void setup() {

Serial.begin(9600);

Serial2.begin(9600, SERIAL\_8N1, VC02\_RX, VC02\_TX); // VC-02 UART

// Setup relay pins

int relays[] = {FAN\_RELAY, LIGHT\_RELAY, TV\_RELAY, COOLER\_RELAY};

for (int i = 0; i < 4; i++) {

pinMode(relays[i], OUTPUT);

digitalWrite(relays[i], HIGH); // OFF (Active LOW)

}

// Connect to Wi-Fi (optional)

Serial.print("Connecting to WiFi");

WiFi.begin(ssid, pass);

unsigned long startTime = millis();

while (WiFi.status() != WL\_CONNECTED && millis() - startTime < 10000) {

Serial.print(".");

delay(500);

}

Serial.println();

if (WiFi.status() == WL\_CONNECTED) {

Serial.println("WiFi connected.");

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

} else {

Serial.println("WiFi connection failed. Running without Blynk.");

}

Serial.println("Setup complete.");

}

void loop() {

checkIR(); // Always process IR

if (WiFi.status() == WL\_CONNECTED) {

Blynk.run(); // Run Blynk only if Wi-Fi is up

}

}

// Function to handle VC-02 IR commands

void checkIR() {

if (Serial2.available() >= 2) {

byte highByte = Serial2.read();

byte lowByte = Serial2.read();

unsigned int cmd = (highByte << 8) | lowByte;

Serial.print("Received IR command: 0x");

Serial.println(cmd, HEX);

switch (cmd) {

case 0xC31D: setRelay(FAN\_RELAY, true); Blynk.virtualWrite(V0, 1); break;

case 0xC30D: setRelay(FAN\_RELAY, false); Blynk.virtualWrite(V0, 0); break;

case 0xA11B: setRelay(LIGHT\_RELAY, true); Blynk.virtualWrite(V1, 1); break;

case 0xA10B: setRelay(LIGHT\_RELAY, false); Blynk.virtualWrite(V1, 0); break;

case 0xD41E: setRelay(TV\_RELAY, true); Blynk.virtualWrite(V2, 1); break;

case 0xD40E: setRelay(TV\_RELAY, false); Blynk.virtualWrite(V2, 0); break;

case 0xE51F: setRelay(COOLER\_RELAY, true); Blynk.virtualWrite(V3, 1); break;

case 0xE50F: setRelay(COOLER\_RELAY, false); Blynk.virtualWrite(V3, 0); break;

default:

Serial.println("Unknown IR command");

break;

}

}

}