// WiFi Credentials

const char\* ssid1 = "Airtel\_prad\_5900";

const char\* password1 = "air97139";

const char\* ssid2 = "Deva";

const char\* password2 = "Vittala@26";

// device names (change names as per requirement)

String Device\_1\_Name = "Moon Light";

String Device\_2\_Name = "Balcony Light";

String Device\_3\_Name = "Light";

String Device\_4\_Name = "Fan";

//Update the HEX code of IR Remote buttons 0x<HEX CODE>

#define IR\_Button\_1 0x1FEE01F

#define IR\_Button\_2 0x1FE10EF

#define IR\_Button\_3 0x1FE906F

#define IR\_Button\_4 0x1FE50AF

#define IR\_Button\_5 0x1FEE0BF

#define IR\_Button\_6 0x1FE10BF

#define IR\_Button\_7 0x1FE806F

#define IR\_Button\_8 0x1FE50BF

#define IR\_All\_On 0x1FE807F

#define IR\_All\_Off 0x1FE48B7

#include <WiFi.h>

#include <Espalexa.h>

#include "BluetoothSerial.h"

#include <IRremote.h>

#include <Preferences.h>

#include <AceButton.h>

using namespace ace\_button;

#if !defined(CONFIG\_BT\_ENABLED) || !defined(CONFIG\_BLUEDROID\_ENABLED)

#error Bluetooth is not enabled! Please enable it in menuconfig

#endif

BluetoothSerial SerialBT;

Espalexa espalexa;

Preferences pref;

// define the GPIO connected with Relays and switches

#define RelayPin1 26 //fan

#define RelayPin2 25 //light

#define RelayPin3 33 //moon light

#define RelayPin4 32 // socket

// Use INPUT\_PULLUP for active-low buttons

#define SwitchPin1 35 // fan green

#define SwitchPin2 34 //light blue

#define SwitchPin3 39 //moon light // purple

#define SwitchPin4 36 //socket //white

#define wifiLed 22

#define IR\_RECV\_PIN 21

#define WiFiResetButton 0 //BOOT button

// Relay State (Active-low: LOW is ON, HIGH is OFF)

// Change initial state to HIGH (OFF) for active-low relays

bool toggleState\_1 = HIGH;

bool toggleState\_2 = HIGH;

bool toggleState\_3 = HIGH;

bool toggleState\_4 = HIGH;

// AceButton requires pull-up to be configured for active-low switches

ButtonConfig config1;

AceButton button1(&config1);

ButtonConfig config2;

AceButton button2(&config2);

ButtonConfig config3;

AceButton button3(&config3);

ButtonConfig config4;

AceButton button4(&config4);

// FIX: Change uint8\_ to uint8\_t in function prototypes

void handleEvent1(AceButton\*, uint8\_t, uint8\_t);

void handleEvent2(AceButton\*, uint8\_t, uint8\_t);

void handleEvent3(AceButton\*, uint8\_t, uint8\_t);

void handleEvent4(AceButton\*, uint8\_t, uint8\_t);

int wifiFlag = 0;

char bt\_data;

IRrecv irrecv(IR\_RECV\_PIN);

decode\_results results;

// prototypes

boolean connectWifi();

//callback functions

void firstLightChanged(uint8\_t brightness);

void secondLightChanged(uint8\_t brightness);

void thirdLightChanged(uint8\_t brightness);

void fourthLightChanged(uint8\_t brightness);

boolean wifiConnected = false;

// ====================== Device Control Callbacks ======================

void firstLightChanged(uint8\_t brightness) {

if (brightness == 255) {

digitalWrite(RelayPin1, LOW); // LOW turns relay ON for active-low

toggleState\_1 = LOW;

} else {

digitalWrite(RelayPin1, HIGH); // HIGH turns relay OFF for active-low

toggleState\_1 = HIGH;

}

pref.putBool("Relay1", (toggleState\_1 == LOW)); // Store the logical state (true for ON)

}

void secondLightChanged(uint8\_t brightness) {

if (brightness == 255) {

digitalWrite(RelayPin2, LOW); // LOW turns relay ON

toggleState\_2 = LOW;

} else {

digitalWrite(RelayPin2, HIGH); // HIGH turns relay OFF

toggleState\_2 = HIGH;

}

pref.putBool("Relay2", (toggleState\_2 == LOW));

}

void thirdLightChanged(uint8\_t brightness) {

if (brightness == 255) {

digitalWrite(RelayPin3, LOW); // LOW turns relay ON

toggleState\_3 = LOW;

} else {

digitalWrite(RelayPin3, HIGH); // HIGH turns relay OFF

toggleState\_3 = HIGH;

}

pref.putBool("Relay3", (toggleState\_3 == LOW));

}

void fourthLightChanged(uint8\_t brightness) {

if (brightness == 255) {

digitalWrite(RelayPin4, LOW); // LOW turns relay ON

toggleState\_4 = LOW;

} else {

digitalWrite(RelayPin4, HIGH); // HIGH turns relay OFF

toggleState\_4 = HIGH;

}

pref.putBool("Relay4", (toggleState\_4 == LOW));

}

void addDevices() {

espalexa.addDevice(Device\_1\_Name, firstLightChanged);

espalexa.addDevice(Device\_2\_Name, secondLightChanged);

espalexa.addDevice(Device\_3\_Name, thirdLightChanged);

espalexa.addDevice(Device\_4\_Name, fourthLightChanged);

espalexa.begin();

}

void sendFeedback(int relay, int value) {

EspalexaDevice\* d = espalexa.getDevice(relay);

d->setPercent(value);

}

// ====================== IR Remote ======================

void ir\_remote() {

if (irrecv.decode(&results)) {

switch (results.value) {

case IR\_Button\_1:

toggleState\_1 = !toggleState\_1; // Toggle the state (HIGH/LOW)

digitalWrite(RelayPin1, toggleState\_1); // Apply the new state

pref.putBool("Relay1", (toggleState\_1 == LOW)); // Save logical state

(toggleState\_1 == HIGH) ? sendFeedback(0, 0) : sendFeedback(0, 100);

break;

case IR\_Button\_2:

toggleState\_2 = !toggleState\_2;

digitalWrite(RelayPin2, toggleState\_2);

pref.putBool("Relay2", (toggleState\_2 == LOW));

(toggleState\_2 == HIGH) ? sendFeedback(1, 0) : sendFeedback(1, 100);

break;

case IR\_Button\_3:

toggleState\_3 = !toggleState\_3;

digitalWrite(RelayPin3, toggleState\_3);

pref.putBool("Relay3", (toggleState\_3 == LOW));

(toggleState\_3 == HIGH) ? sendFeedback(2, 0) : sendFeedback(2, 100);

break;

case IR\_Button\_4:

toggleState\_4 = !toggleState\_4;

digitalWrite(RelayPin4, toggleState\_4);

pref.putBool("Relay4", (toggleState\_4 == LOW));

(toggleState\_4 == HIGH) ? sendFeedback(3, 0) : sendFeedback(3, 100);

break;

case IR\_All\_Off: all\_SwitchOff(); break;

case IR\_All\_On: all\_SwitchOn(); break;

}

irrecv.resume();

}

}

// ====================== Bluetooth Control ======================

void bluetooth\_control() {

if (SerialBT.available()) {

bt\_data = SerialBT.read();

switch (bt\_data) {

case 'A': digitalWrite(RelayPin1, LOW); toggleState\_1 = LOW; sendFeedback(0, 100); pref.putBool("Relay1", true); break;

case 'a': digitalWrite(RelayPin1, HIGH); toggleState\_1 = HIGH; sendFeedback(0, 0); pref.putBool("Relay1", false); break;

case 'B': digitalWrite(RelayPin2, LOW); toggleState\_2 = LOW; sendFeedback(1, 100); pref.putBool("Relay2", true); break;

case 'b': digitalWrite(RelayPin2, HIGH); toggleState\_2 = HIGH; sendFeedback(1, 0); pref.putBool("Relay2", false); break;

case 'C': digitalWrite(RelayPin3, LOW); toggleState\_3 = LOW; sendFeedback(2, 100); pref.putBool("Relay3", true); break;

case 'c': digitalWrite(RelayPin3, HIGH); toggleState\_3 = HIGH; sendFeedback(2, 0); pref.putBool("Relay3", false); break;

case 'D': digitalWrite(RelayPin4, LOW); toggleState\_4 = LOW; sendFeedback(3, 100); pref.putBool("Relay4", true); break;

case 'd': digitalWrite(RelayPin4, HIGH); toggleState\_4 = HIGH; sendFeedback(3, 0); pref.putBool("Relay4", false); break;

case 'Z': all\_SwitchOn(); break;

case 'z': all\_SwitchOff(); break;

}

}

}

// ====================== Relay Control ======================

void all\_SwitchOn() {

toggleState\_1 = LOW; digitalWrite(RelayPin1, LOW); pref.putBool("Relay1", true); sendFeedback(0, 100);

toggleState\_2 = LOW; digitalWrite(RelayPin2, LOW); pref.putBool("Relay2", true); sendFeedback(1, 100);

toggleState\_3 = LOW; digitalWrite(RelayPin3, LOW); pref.putBool("Relay3", true); sendFeedback(2, 100);

toggleState\_4 = LOW; digitalWrite(RelayPin4, LOW); pref.putBool("Relay4", true); sendFeedback(3, 100);

}

void all\_SwitchOff() {

toggleState\_1 = HIGH; digitalWrite(RelayPin1, HIGH); pref.putBool("Relay1", false); sendFeedback(0, 0);

toggleState\_2 = HIGH; digitalWrite(RelayPin2, HIGH); pref.putBool("Relay2", false); sendFeedback(1, 0);

toggleState\_3 = HIGH; digitalWrite(RelayPin3, HIGH); pref.putBool("Relay3", false); sendFeedback(2, 0);

toggleState\_4 = HIGH; digitalWrite(RelayPin4, HIGH); pref.putBool("Relay4", false); sendFeedback(3, 0);

}

boolean connectWifi() {

int attempts = 0;

Serial.begin(115200);

WiFi.begin(ssid1, password1);

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

while (WiFi.status() != WL\_CONNECTED && attempts < 15) {

Serial.print(".");

digitalWrite(wifiLed, LOW); // Blink LED with LOW for active-low

delay(500);

digitalWrite(wifiLed, HIGH); // Turn LED off with HIGH

delay(500);

attempts++;

}

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

digitalWrite(wifiLed, LOW); // Keep LED on (LOW) when connected

Serial.println("Connected to WiFi");

return true;

} else {

digitalWrite(wifiLed, HIGH); // Keep LED off (HIGH) if not connected

Serial.println("Failed to connect to WiFi");

return false;

}

}

// ====================== AceButton Switch Control ======================

// This is the correct way to handle active-low buttons with AceButton

void handleEvent1(AceButton\* button, uint8\_t eventType, uint8\_t buttonState) {

if (eventType == AceButton::kEventPressed) {

toggleState\_1 = !toggleState\_1;

digitalWrite(RelayPin1, toggleState\_1);

pref.putBool("Relay1", (toggleState\_1 == LOW));

(toggleState\_1 == LOW) ? sendFeedback(0, 100) : sendFeedback(0, 0);

}

}

void handleEvent2(AceButton\* button, uint8\_t eventType, uint8\_t buttonState) {

if (eventType == AceButton::kEventPressed) {

toggleState\_2 = !toggleState\_2;

digitalWrite(RelayPin2, toggleState\_2);

pref.putBool("Relay2", (toggleState\_2 == LOW));

(toggleState\_2 == LOW) ? sendFeedback(1, 100) : sendFeedback(1, 0);

}

}

void handleEvent3(AceButton\* button, uint8\_t eventType, uint8\_t buttonState) {

if (eventType == AceButton::kEventPressed) {

toggleState\_3 = !toggleState\_3;

digitalWrite(RelayPin3, toggleState\_3);

pref.putBool("Relay3", (toggleState\_3 == LOW));

(toggleState\_3 == LOW) ? sendFeedback(2, 100) : sendFeedback(2, 0);

}

}

void handleEvent4(AceButton\* button, uint8\_t eventType, uint8\_t buttonState) {

if (eventType == AceButton::kEventPressed) {

toggleState\_4 = !toggleState\_4;

digitalWrite(RelayPin4, toggleState\_4);

pref.putBool("Relay4", (toggleState\_4 == LOW));

(toggleState\_4 == LOW) ? sendFeedback(3, 100) : sendFeedback(3, 0);

}

}

void setup() {

Serial.begin(115200);

// Initialize relays as outputs and start in the OFF (HIGH) state

pinMode(RelayPin1, OUTPUT);

pinMode(RelayPin2, OUTPUT);

pinMode(RelayPin3, OUTPUT);

pinMode(RelayPin4, OUTPUT);

digitalWrite(RelayPin1, HIGH);

digitalWrite(RelayPin2, HIGH);

digitalWrite(RelayPin3, HIGH);

digitalWrite(RelayPin4, HIGH);

// Initialize switches with pull-up resistors for active-low buttons

pinMode(SwitchPin1, INPUT\_PULLUP);

pinMode(SwitchPin2, INPUT\_PULLUP);

pinMode(SwitchPin3, INPUT\_PULLUP);

pinMode(SwitchPin4, INPUT\_PULLUP);

// Configure AceButton for active-low switches

config1.setFeature(ButtonConfig::kFeatureSuppressAfterClick);

config1.setEventHandler(handleEvent1);

config2.setFeature(ButtonConfig::kFeatureSuppressAfterClick);

config2.setEventHandler(handleEvent2);

config3.setFeature(ButtonConfig::kFeatureSuppressAfterClick);

config3.setEventHandler(handleEvent3);

config4.setFeature(ButtonConfig::kFeatureSuppressAfterClick);

config4.setEventHandler(handleEvent4);

// Initialize IR remote

irrecv.enableIRIn();

// Initialize preferences for state persistence

pref.begin("smart\_home", false);

toggleState\_1 = pref.getBool("Relay1", false) ? LOW : HIGH; // Read and apply last state

toggleState\_2 = pref.getBool("Relay2", false) ? LOW : HIGH;

toggleState\_3 = pref.getBool("Relay3", false) ? LOW : HIGH;

toggleState\_4 = pref.getBool("Relay4", false) ? LOW : HIGH;

digitalWrite(RelayPin1, toggleState\_1);

digitalWrite(RelayPin2, toggleState\_2);

digitalWrite(RelayPin3, toggleState\_3);

digitalWrite(RelayPin4, toggleState\_4);

// Initialize Bluetooth

SerialBT.begin("DNY\_AI\_E\_HALL");

// Connect to WiFi

pinMode(wifiLed, OUTPUT);

if (connectWifi()) {

addDevices();

}

}

void loop() {

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

espalexa.loop();

}

// Check for Bluetooth data

bluetooth\_control();

// Check for IR remote signals

ir\_remote();

// Check for button presses (using AceButton)

button1.check();

button2.check();

button3.check();

button4.check();

}