#define BLYNK\_TEMPLATE\_ID "TMPL3oyFS2zTp"

#define BLYNK\_TEMPLATE\_NAME "Quickstart Template"

#define BLYNK\_AUTH\_TOKEN "0nSU38eiRHNwJ4Z0\_T3FZAu4bQqPEySG"

char ssid[] = "";

char pass[] = "";

// HEX code

#define IR\_Button\_1   0x80BF49B6

#define IR\_Button\_2   0x80BFC936

#define IR\_Button\_3   0x80BF33CC

#define IR\_Button\_4   0x80BF718E

#define IR\_Fan\_Up     0x80BF53AC

#define IR\_Fan\_Down   0x80BF4BB4

#define IR\_All\_Off    0x80BF3BC4

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include <Preferences.h>

#include <IRremote.h>

#include <DHT.h>

Preferences pref;

// define the GPIO connected with Relays and switches

#define RelayPin1 23  //D23

#define RelayPin2 22  //D22

#define RelayPin3 21  //D21

#define RelayPin4 19  //D19

#define SwitchPin1 13  //D13

#define SwitchPin2 12  //D12

#define SwitchPin3 14  //D14

#define SwitchPin4 27  //D27

#define FanRelay1 18  //D18

#define FanRelay2 5   //D5

#define FanRelay3 25  //D25

#define FanSwitch1 33  //D33

#define FanSwitch2 32  //D32

#define FanSwitch3 15  //D15

#define FanSwitch4 4   //D4

#define wifiLed   2   //D2

#define IR\_RECV\_PIN   35 // D35 (IR receiver pin)

#define DHTPIN        16 //D16  pin connected with DHT

//Change the virtual pins according the rooms

#define VPIN\_BUTTON\_FAN  V0

#define VPIN\_BUTTON\_1    V1

#define VPIN\_BUTTON\_2    V2

#define VPIN\_BUTTON\_3    V3

#define VPIN\_BUTTON\_4    V4

#define VPIN\_BUTTON\_C    V5

#define VPIN\_TEMPERATURE V6

#define VPIN\_HUMIDITY    V7

// Uncomment whatever type you're using!

#define DHTTYPE DHT11     // DHT 11

//#define DHTTYPE DHT22   // DHT 22, AM2302, AM2321

//#define DHTTYPE DHT21   // DHT 21, AM2301

int currSpeed = 0;

// Relay State

bool toggleState\_1 = LOW;

bool toggleState\_2 = LOW;

bool toggleState\_3 = LOW;

bool toggleState\_4 = LOW;

bool fanSpeed\_0 = LOW;

bool fanSpeed\_1 = LOW;

bool fanSpeed\_2 = LOW;

bool fanSpeed\_3 = LOW;

bool fanSpeed\_4 = LOW;

// Switch State

bool SwitchState\_1 = LOW;

bool SwitchState\_2 = LOW;

bool SwitchState\_3 = LOW;

bool SwitchState\_4 = LOW;

float temperature1 = 0;

float humidity1   = 0;

int wifiFlag = 0;

IRrecv irrecv(IR\_RECV\_PIN);

decode\_results results;

DHT dht(DHTPIN, DHTTYPE);

char auth[] = BLYNK\_AUTH\_TOKEN;

BlynkTimer timer;

BLYNK\_WRITE(VPIN\_BUTTON\_1) {

  toggleState\_1 = param.asInt();

  digitalWrite(RelayPin1, !toggleState\_1);

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

}

BLYNK\_WRITE(VPIN\_BUTTON\_2) {

  toggleState\_2 = param.asInt();

  digitalWrite(RelayPin2, !toggleState\_2);

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

}

BLYNK\_WRITE(VPIN\_BUTTON\_3) {

  toggleState\_3 = param.asInt();

  digitalWrite(RelayPin3, !toggleState\_3);

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

}

BLYNK\_WRITE(VPIN\_BUTTON\_4) {

  toggleState\_4 = param.asInt();

  digitalWrite(RelayPin4, !toggleState\_4);

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

}

BLYNK\_WRITE(VPIN\_BUTTON\_FAN) {

  currSpeed = param.asInt();

  fanSpeedControl(currSpeed);

  pref.putInt("Fan", currSpeed);

}

BLYNK\_WRITE(VPIN\_BUTTON\_C) {

  all\_SwitchOff();

}

void checkBlynkStatus() { // called every 3 seconds by SimpleTimer

  bool isconnected = Blynk.connected();

  if (isconnected == false) {

    wifiFlag = 1;

    digitalWrite(wifiLed, LOW);

    Serial.println("Blynk Not Connected");

  }

  if (isconnected == true) {

    wifiFlag = 0;

    digitalWrite(wifiLed, HIGH);

    //Serial.println("Blynk Connected");

  }

}

BLYNK\_CONNECTED() {

  // update the latest state to the server

  Blynk.virtualWrite(VPIN\_BUTTON\_1, toggleState\_1);

  Blynk.virtualWrite(VPIN\_BUTTON\_2, toggleState\_2);

  Blynk.virtualWrite(VPIN\_BUTTON\_3, toggleState\_3);

  Blynk.virtualWrite(VPIN\_BUTTON\_4, toggleState\_4);

  Blynk.virtualWrite(VPIN\_BUTTON\_FAN, currSpeed);

  Blynk.syncVirtual(VPIN\_TEMPERATURE);

  Blynk.syncVirtual(VPIN\_HUMIDITY);

}

void readSensor(){

  float h = dht.readHumidity();

  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {

    Serial.println("Failed to read from DHT sensor!");

    return;

  }

  else {

    humidity1 = h;

    temperature1 = t;

  }

}

void sendSensor()

{

  readSensor();

  // You can send any value at any time.

  Blynk.virtualWrite(VPIN\_HUMIDITY, humidity1);

  Blynk.virtualWrite(VPIN\_TEMPERATURE, temperature1);

}

void manual\_control()

{

  if (digitalRead(SwitchPin1) == LOW && SwitchState\_1 == LOW) {

    digitalWrite(RelayPin1, LOW);

    toggleState\_1 = HIGH;

    SwitchState\_1 = HIGH;

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

    Blynk.virtualWrite(VPIN\_BUTTON\_1, toggleState\_1);

    Serial.println("Switch-1 on");

  }

  if (digitalRead(SwitchPin1) == HIGH && SwitchState\_1 == HIGH) {

    digitalWrite(RelayPin1, HIGH);

    toggleState\_1 = LOW;

    SwitchState\_1 = LOW;

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

    Blynk.virtualWrite(VPIN\_BUTTON\_1, toggleState\_1);

    Serial.println("Switch-1 off");

  }

  if (digitalRead(SwitchPin2) == LOW && SwitchState\_2 == LOW) {

    digitalWrite(RelayPin2, LOW);

    toggleState\_2 = HIGH;

    SwitchState\_2 = HIGH;

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

    Blynk.virtualWrite(VPIN\_BUTTON\_2, toggleState\_2);

    Serial.println("Switch-2 on");

  }

  if (digitalRead(SwitchPin2) == HIGH && SwitchState\_2 == HIGH) {

    digitalWrite(RelayPin2, HIGH);

    toggleState\_2 = LOW;

    SwitchState\_2 = LOW;

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

    Blynk.virtualWrite(VPIN\_BUTTON\_2, toggleState\_2);

    Serial.println("Switch-2 off");

  }

  if (digitalRead(SwitchPin3) == LOW && SwitchState\_3 == LOW) {

    digitalWrite(RelayPin3, LOW);

    toggleState\_3 = HIGH;

    SwitchState\_3 = HIGH;

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

    Blynk.virtualWrite(VPIN\_BUTTON\_3, toggleState\_3);

    Serial.println("Switch-3 on");

  }

  if (digitalRead(SwitchPin3) == HIGH && SwitchState\_3 == HIGH) {

    digitalWrite(RelayPin3, HIGH);

    toggleState\_3 = LOW;

    SwitchState\_3 = LOW;

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

    Blynk.virtualWrite(VPIN\_BUTTON\_3, toggleState\_3);

    Serial.println("Switch-3 off");

  }

  if (digitalRead(SwitchPin4) == LOW && SwitchState\_4 == LOW) {

    digitalWrite(RelayPin4, LOW);

    toggleState\_4 = HIGH;

    SwitchState\_4 = HIGH;

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

    Blynk.virtualWrite(VPIN\_BUTTON\_4, toggleState\_4);

    Serial.println("Switch-4 on");

  }

  if (digitalRead(SwitchPin4) == HIGH && SwitchState\_4 == HIGH) {

    digitalWrite(RelayPin4, HIGH);

    toggleState\_4 = LOW;

    SwitchState\_4 = LOW;

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

    Blynk.virtualWrite(VPIN\_BUTTON\_4, toggleState\_4);

    Serial.println("Switch-4 off");

  }

}

void ir\_remote(){

  if (irrecv.decode(&results)) {

      switch(results.value){

          case IR\_Button\_1:

            digitalWrite(RelayPin1, toggleState\_1);

            toggleState\_1 = !toggleState\_1;

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

            Blynk.virtualWrite(VPIN\_BUTTON\_1, toggleState\_1);

            delay(100);

            break;

          case IR\_Button\_2:

            digitalWrite(RelayPin2, toggleState\_2);

            toggleState\_2 = !toggleState\_2;

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

            Blynk.virtualWrite(VPIN\_BUTTON\_2, toggleState\_2);

            delay(100);

            break;

          case IR\_Button\_3:

            digitalWrite(RelayPin3, toggleState\_3);

            toggleState\_3 = !toggleState\_3;

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

            Blynk.virtualWrite(VPIN\_BUTTON\_3, toggleState\_3);

            delay(100);

            break;

          case IR\_Button\_4:

            digitalWrite(RelayPin4, toggleState\_4);

            toggleState\_4 = !toggleState\_4;

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

            Blynk.virtualWrite(VPIN\_BUTTON\_4, toggleState\_4);

            delay(100);

            break;

          case IR\_Fan\_Up:

            if(currSpeed < 4){

              currSpeed = currSpeed + 1;

              fanSpeedControl(currSpeed);

              pref.putInt("Fan", currSpeed);

              Blynk.virtualWrite(VPIN\_BUTTON\_FAN, currSpeed);

            }

            delay(100);

            break;

          case IR\_Fan\_Down:

            if(currSpeed > 0){

              currSpeed = currSpeed - 1;

              fanSpeedControl(currSpeed);

              pref.putInt("Fan", currSpeed);

              Blynk.virtualWrite(VPIN\_BUTTON\_FAN, currSpeed);

            }

            delay(100);

            break;

          case IR\_All\_Off:

            all\_SwitchOff();

            break;

          default : break;

        }

        //Serial.println(results.value, HEX);

        irrecv.resume();

  }

}

void all\_SwitchOff(){

  toggleState\_1 = 0; digitalWrite(RelayPin1, HIGH); pref.putBool("Relay1", toggleState\_1); Blynk.virtualWrite(VPIN\_BUTTON\_1, toggleState\_1); delay(100);

  toggleState\_2 = 0; digitalWrite(RelayPin2, HIGH); pref.putBool("Relay2", toggleState\_2); Blynk.virtualWrite(VPIN\_BUTTON\_2, toggleState\_2); delay(100);

  toggleState\_3 = 0; digitalWrite(RelayPin3, HIGH); pref.putBool("Relay3", toggleState\_3); Blynk.virtualWrite(VPIN\_BUTTON\_3, toggleState\_3); delay(100);

  toggleState\_4 = 0; digitalWrite(RelayPin4, HIGH); pref.putBool("Relay4", toggleState\_4); Blynk.virtualWrite(VPIN\_BUTTON\_4, toggleState\_4); delay(100);

  currSpeed = 0; fanSpeedControl(currSpeed); pref.putInt("Fan", currSpeed); Blynk.virtualWrite(VPIN\_BUTTON\_FAN, currSpeed); delay(100);

  Blynk.virtualWrite(VPIN\_HUMIDITY, humidity1);

  Blynk.virtualWrite(VPIN\_TEMPERATURE, temperature1);

}

void getRelayState()

{

  //Serial.println("reading data from NVS");

  toggleState\_1 = pref.getBool("Relay1", 0);

  digitalWrite(RelayPin1, !toggleState\_1);

  Blynk.virtualWrite(VPIN\_BUTTON\_1, toggleState\_1);

  delay(200);

  toggleState\_2 = pref.getBool("Relay2", 0);

  digitalWrite(RelayPin2, !toggleState\_2);

  Blynk.virtualWrite(VPIN\_BUTTON\_2, toggleState\_2);

  delay(200);

  toggleState\_3 = pref.getBool("Relay3", 0);

  digitalWrite(RelayPin3, !toggleState\_3);

  Blynk.virtualWrite(VPIN\_BUTTON\_3, toggleState\_3);

  delay(200);

  toggleState\_4 = pref.getBool("Relay4", 0);

  digitalWrite(RelayPin4, !toggleState\_4);

  Blynk.virtualWrite(VPIN\_BUTTON\_4, toggleState\_4);

  delay(200);

  currSpeed = pref.getInt("Fan", 0);

  fanSpeedControl(currSpeed);

  Blynk.virtualWrite(VPIN\_BUTTON\_4, currSpeed);

  delay(200);

}

void setup()

{

  Serial.begin(9600);

  //Open namespace in read-write mode

  pref.begin("Relay\_State", false);

  pinMode(RelayPin1, OUTPUT);

  pinMode(RelayPin2, OUTPUT);

  pinMode(RelayPin3, OUTPUT);

  pinMode(RelayPin4, OUTPUT);

  pinMode(FanRelay1, OUTPUT);

  pinMode(FanRelay2, OUTPUT);

  pinMode(FanRelay3, OUTPUT);

  pinMode(wifiLed, OUTPUT);

  pinMode(SwitchPin1, INPUT\_PULLUP);

  pinMode(SwitchPin2, INPUT\_PULLUP);

  pinMode(SwitchPin3, INPUT\_PULLUP);

  pinMode(SwitchPin4, INPUT\_PULLUP);

  pinMode(FanSwitch1, INPUT\_PULLUP);

  pinMode(FanSwitch2, INPUT\_PULLUP);

  pinMode(FanSwitch3, INPUT\_PULLUP);

  pinMode(FanSwitch4, INPUT\_PULLUP);

  //During Starting all Relays should TURN OFF

  digitalWrite(RelayPin1, !toggleState\_1);

  digitalWrite(RelayPin2, !toggleState\_2);

  digitalWrite(RelayPin3, !toggleState\_3);

  digitalWrite(RelayPin4, !toggleState\_4);

  digitalWrite(FanRelay1, HIGH);

  digitalWrite(FanRelay2, HIGH);

  digitalWrite(FanRelay3, HIGH);

  digitalWrite(wifiLed, LOW);

  irrecv.enableIRIn(); // Enabling IR sensor

  dht.begin();    // Enabling DHT sensor

  WiFi.begin(ssid, pass);

  timer.setInterval(2000L, checkBlynkStatus); // check if Blynk server is connected every 2 seconds

  timer.setInterval(1000L, sendSensor);

  Blynk.config(auth);

  delay(1000);

  getRelayState(); //fetch data from NVS Flash Memory

//  delay(1000);

}

void loop()

{

  Blynk.run();

  timer.run(); // Initiates SimpleTimer

  manual\_control();

  fanRegularor(); //Control Fan Speed

  ir\_remote(); //IR remote Control

}

void fanRegularor(){

  if (digitalRead(FanSwitch1) == HIGH && digitalRead(FanSwitch2) == HIGH && digitalRead(FanSwitch3) == HIGH && digitalRead(FanSwitch4) == HIGH  && fanSpeed\_0 == LOW)

  {

    currSpeed = 0;

    fanSpeedControl(currSpeed);

    pref.putInt("Fan", currSpeed);

    Blynk.virtualWrite(VPIN\_BUTTON\_FAN, 0);

    fanSpeed\_1 = LOW;

    fanSpeed\_2 = LOW;

    fanSpeed\_3 = LOW;

    fanSpeed\_4 = LOW;

    fanSpeed\_0 = HIGH;

  }

  if (digitalRead(FanSwitch1) == LOW && fanSpeed\_1 == LOW)

  {

    currSpeed = 1;

    fanSpeedControl(currSpeed);

    pref.putInt("Fan", currSpeed);

    Blynk.virtualWrite(VPIN\_BUTTON\_FAN, 1);

    fanSpeed\_1 = HIGH;

    fanSpeed\_2 = LOW;

    fanSpeed\_3 = LOW;

    fanSpeed\_4 = LOW;

    fanSpeed\_0 = LOW;

  }

  if (digitalRead(FanSwitch2) == LOW && digitalRead(FanSwitch3) == HIGH && fanSpeed\_2 == LOW)

  {

    currSpeed = 2;

    fanSpeedControl(currSpeed);

    pref.putInt("Fan", currSpeed);

    Blynk.virtualWrite(VPIN\_BUTTON\_FAN, 2);

    fanSpeed\_1 = LOW;

    fanSpeed\_2 = HIGH;

    fanSpeed\_3 = LOW;

    fanSpeed\_4 = LOW;

    fanSpeed\_0 = LOW;

  }

  if (digitalRead(FanSwitch2) == LOW && digitalRead(FanSwitch3) == LOW && fanSpeed\_3 == LOW)

  {

    currSpeed = 3;

    fanSpeedControl(currSpeed);

    pref.putInt("Fan", currSpeed);

    Blynk.virtualWrite(VPIN\_BUTTON\_FAN, 3);

    fanSpeed\_1 = LOW;

    fanSpeed\_2 = LOW;

    fanSpeed\_3 = HIGH;

    fanSpeed\_4 = LOW;

    fanSpeed\_0 = LOW;

  }

  if (digitalRead(FanSwitch4) == LOW  && fanSpeed\_4 == LOW)

  {

    currSpeed = 4;

    fanSpeedControl(currSpeed);

    pref.putInt("Fan", currSpeed);

    Blynk.virtualWrite(VPIN\_BUTTON\_FAN, 4);

    fanSpeed\_1 = LOW;

    fanSpeed\_2 = LOW;

    fanSpeed\_3 = LOW;

    fanSpeed\_4 = HIGH;

    fanSpeed\_0 = LOW;

  }

}

void fanSpeedControl(int fanSpeed){

  switch(fanSpeed){

      case 0:

        digitalWrite(FanRelay1, HIGH);

        digitalWrite(FanRelay2, HIGH);

        digitalWrite(FanRelay3, HIGH);

      break;

      case 1:

        digitalWrite(FanRelay1, HIGH);

        digitalWrite(FanRelay2, HIGH);

        digitalWrite(FanRelay3, HIGH);

        delay(500);

        digitalWrite(FanRelay1, LOW);

      break;

      case 2:

        digitalWrite(FanRelay1, HIGH);

        digitalWrite(FanRelay2, HIGH);

        digitalWrite(FanRelay3, HIGH);

        delay(500);

        digitalWrite(FanRelay2, LOW);

      break;

      case 3:

        digitalWrite(FanRelay1, HIGH);

        digitalWrite(FanRelay2, HIGH);

        digitalWrite(FanRelay3, HIGH);

        delay(500);

        digitalWrite(FanRelay1, LOW);

        digitalWrite(FanRelay2, LOW);

      break;

      case 4:

        digitalWrite(FanRelay1, HIGH);

        digitalWrite(FanRelay2, HIGH);

        digitalWrite(FanRelay3, HIGH);

        delay(500);

        digitalWrite(FanRelay3, LOW);

      break;

      default : break;

    }

}