// Fill-in information from your Blynk Template here

#define BLYNK\_TEMPLATE\_ID "TMPLrey0IckG"

#define BLYNK\_TEMPLATE\_NAME "iot wifi"

#define BLYNK\_FIRMWARE\_VERSION "0.1.0"

#define BLYNK\_PRINT Serial

//#define BLYNK\_DEBUG

#define APP\_DEBUG

// Uncomment your board, or configure a custom board in Settings.h

//#define USE\_SPARKFUN\_BLYNK\_BOARD

#define USE\_NODE\_MCU\_BOARD

//#define USE\_WITTY\_CLOUD\_BOARD

//#define USE\_WEMOS\_D1\_MINI

#include "BlynkEdgent.h"

#include <DHT.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 64 // OLED display height, in pixels

#define OLED\_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)

#define DHTPIN D4

#define DHTTYPE DHT22 // DHT 22

DHT dht(DHTPIN, DHTTYPE);

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

float voltage;

int bat\_percentage;

int analogInPin = A0; // Analog input pin

int sensorValue;

float calibration = 0.40; // Check Battery voltage using multimeter & add/subtract the value

void setup()

{

Serial.begin(115200);

//delay(100);

BlynkEdgent.begin();

dht.begin();

display.begin(SSD1306\_SWITCHCAPVCC, 0x3C); //initialize with the I2C addr 0x3C (128x64)

display.clearDisplay();

display.setTextColor(WHITE);

//delay(100);

}

void loop() {

BlynkEdgent.run();

float t = dht.readTemperature();

float h = dht.readHumidity();

sensorValue = analogRead(analogInPin);

voltage = (((sensorValue \* 3.3) / 1024) \* 2 + calibration); //multiply by two as voltage divider network is 100K & 100K Resistor

bat\_percentage = mapfloat(voltage, 2.8, 4.2, 0, 100); //2.8V as Battery Cut off Voltage & 4.2V as Maximum Voltage

if (bat\_percentage >= 100)

{

bat\_percentage = 100;

}

if (bat\_percentage <= 0)

{

bat\_percentage = 1;

}

//send data to blynk

Blynk.virtualWrite(V1, t); //for Temperature

Blynk.virtualWrite(V2, h); //for Humidity

Blynk.virtualWrite(V3, voltage); // for battery voltage

Blynk.virtualWrite(V4, bat\_percentage); // for battery percentage

//Print data on serial monitor

Serial.print("Temperature: ");

Serial.print(t);

Serial.println(" \*C");

Serial.print("Humidity: ");

Serial.print(h);

Serial.println(" %");

Serial.print("Analog Value = ");

Serial.println(sensorValue);

Serial.print("Output Voltage = ");

Serial.println(voltage);

Serial.print("Battery Percentage = ");

Serial.println(bat\_percentage);

Serial.println();

Serial.println("");

Serial.println();

//delay(1000);

if (bat\_percentage <=30)

{

Serial.println("Battery level below 30%, Charge battery on time");

//send notification

Blynk.logEvent("battery\_low", "Battery is getting low.... Plugin to charge") ;

delay(500);

}

// display temperature on OLED

display.clearDisplay();

display.setTextColor(WHITE);

display.setTextSize(1);

display.setCursor(0, 0);

display.print("Temperature: ");

display.setTextSize(2);

display.setCursor(0, 10);

display.print(t);

display.print(" ");

display.setTextSize(1);

display.cp437(true);

display.write(167);

display.setTextSize(2);

display.print("C");

// display humidity on OLED

display.setTextSize(1);

display.setCursor(0, 35);

display.print("Humidity: ");

display.setTextSize(2);

display.setCursor(0, 45);

display.print(h);

display.print(" %");

display.display();

//delay(1500);

}

float mapfloat(float x, float in\_min, float in\_max, float out\_min, float out\_max)

{

return (x - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min;

}