#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <OneWire.h>

#include <DallasTemperature.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

#define BLYNK\_PRINT Serial

#define ONE\_WIRE\_BUS 12

int fanPin = 16;

int dutyCycle = 0;

float temp = 0;

int threshold = 30;

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire);

WidgetLED FAN(V0);

char auth[] = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

char ssid[] = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

char pass[] = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

void setup()

{

  Serial.begin(115200);

  sensors.begin();

  pinMode(fanPin, OUTPUT);

  lcd.init();

  lcd.backlight();

  lcd.setCursor(0, 0);

  lcd.print("  Temperature ");

  lcd.setCursor(0, 1);

  lcd.print("Monitoring System");

  delay(4000);

  lcd.clear();

  analogWriteRange(100);

  analogWriteFreq(10000);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

}

BLYNK\_WRITE(V7)

{

  threshold = param.asInt();

  Serial.print(" The Threshhold thresholdue is: ");

  Serial.println(threshold);

  Serial.println();

}

void controlFanSpeed(int fanSpeedPercent)

{

  analogWrite(fanPin, fanSpeedPercent);

  Serial.print("Fan Speed: ");

  Serial.print(fanSpeedPercent);

  Serial.println("%");

  lcd.setCursor(0, 1);

  lcd.print("Fan Speed: ");

  lcd.print(fanSpeedPercent);

  lcd.print("%");

}

void loop()

{

  Blynk.run();

  sensors.requestTemperatures();

  temp = sensors.getTempCByIndex(0);

  Serial.print("Temperature: ");

  Serial.print(temp);

  Serial.println("\*C");

  lcd.setCursor(0, 0);

  lcd.print("Temp: ");

  lcd.print(temp);

  lcd.print("\*C");

  Blynk.virtualWrite(V3, temp);

  if (temp >= threshold)

  {

    FAN.on();

    int fanSpeedPercent = map(temp, threshold, 55, 10, 100);

    controlFanSpeed(fanSpeedPercent);

    Blynk.virtualWrite(V4, fanSpeedPercent);

  }

  else if (temp < threshold)

  {

    FAN.off();

    int fanSpeedPercent = 0;

    controlFanSpeed(fanSpeedPercent);

    Blynk.virtualWrite(V4, fanSpeedPercent);

  }

}