#include <ESP8266WiFi.h>

#include <Firebase\_ESP\_Client.h>

#include "addons/TokenHelper.h"

#include "addons/RTDBHelper.h"

#include <Servo.h>

#include <DHT.h>

#define WIFI\_SSID "123456789"

#define WIFI\_PASSWORD "123456789"

#define API\_KEY "AIzaSyCLNpnnqKZE40UnVSCWMYhmDlzzH7ffpjI"

#define DATABASE\_URL "https://cradle-c01ff-default-rtdb.firebaseio.com/"

FirebaseData fbdo;

FirebaseAuth auth;

FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;

bool signupOK = false;

const int micPin = D1;

const int servoPin = D4;

const int dhtPin = D3;

const int wetSensorPin = D5;

const int wetLED = D7;

Servo servo;

DHT dht(dhtPin, DHT11);

void setup() {

pinMode(micPin, INPUT);

pinMode(wetSensorPin, INPUT);

pinMode(wetLED, OUTPUT);

servo.attach(servoPin);

Serial.begin(9600);

dht.begin();

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

Serial.print("Connecting to Wi-Fi");

while (WiFi.status() != WL\_CONNECTED) {

Serial.print(".");

delay(300);

}

Serial.println();

Serial.print("Connected with IP: ");

Serial.println(WiFi.localIP());

// Configure Firebase

config.api\_key = API\_KEY;

config.database\_url = DATABASE\_URL;

if (Firebase.signUp(&config, &auth, "", "")){ //in double quotes we have to give our email and password

Serial.println("Firebase authentication successful");

signupOK = true;

}

else{

Serial.printf("Firebase signup error: %s\n", config.signer.signupError.message.c\_str());

}

Firebase.begin(&config, &auth);

Firebase.reconnectWiFi(true);

}

void loop() {

int micValue = digitalRead(micPin);

float temperature = dht.readTemperature();

float humidity = dht.readHumidity();

int wetValue = digitalRead(wetSensorPin);

if (micValue == HIGH) {

// Rotate servo from 0 to 90 degrees

for (int angle = 0; angle <= 90; angle += 1) {

servo.write(angle);

delay(15);

}

// Rotate servo from 180 to 0 degrees

for (int angle = 90; angle >= 0; angle -= 1) {

servo.write(angle);

delay(15);

}

Serial.println("soundValue:");

Serial.println(micValue);

Serial.println("Sound detected");

} else {

servo.write(0);

Serial.println("No sound detected");

}

if (wetValue == HIGH){

digitalWrite(wetLED, HIGH);

Serial.println("Wetness detected");

} else{

digitalWrite(wetLED,LOW);

Serial.println("No Wetness detected");

}

// Control temperature LED

if (temperature > 30) {

digitalWrite(wetLED, HIGH);

Serial.println("Temperature is HIGH: Fan ON");

} else {

digitalWrite(wetLED, LOW);

Serial.println("Temperature is LOW: Fan OFF");

}

// Print temperature to the serial monitor

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.println(" °C");

Serial.print("Humidity: ");

Serial.print(humidity);

Serial.println("%");

// Send micValue to Firebase

// setting time to read data

if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 1000 || sendDataPrevMillis == 0)){

sendDataPrevMillis = millis();

if (Firebase.RTDB.setInt(&fbdo, "main/micValue", micValue)){

Serial.println("micValue data sent to Firebase");

Serial.println("PATH: " + fbdo.dataPath());

Serial.println("TYPE: " + fbdo.dataType());

}

else {

Serial.println("Failed to send micvalue data to Firebase"+ fbdo.errorReason());

}

if (Firebase.RTDB.setFloat(&fbdo, "main/temperature", temperature)){

Serial.println("temperature data sent to Firebase");

Serial.println("PATH: " + fbdo.dataPath());

Serial.println("TYPE: " + fbdo.dataType());

}

else {

Serial.println("Failed to send temperature data to Firebase"+ fbdo.errorReason());

}

if (Firebase.RTDB.setFloat(&fbdo, "main/humidity", humidity)){

Serial.println("humidity data sent to Firebase");

Serial.println("PATH: " + fbdo.dataPath());

Serial.println("TYPE: " + fbdo.dataType());

}

else {

Serial.println("Failed to send humidity data to Firebase"+ fbdo.errorReason());

}

if (Firebase.RTDB.setInt(&fbdo, "main/wetValue", wetValue)){

Serial.println("WetValue sent to Firebase");

Serial.println("PATH: " + fbdo.dataPath());

Serial.println("TYPE: " + fbdo.dataType());

}

else {

Serial.println("Failed to send WetValue to Firebase"+ fbdo.errorReason());

}

delay(1000); // Adjust the delay time as needed

}

}