#include <WiFi.h>

#include <Wire.h>

#include <PubSubClient.h>

#include <LiquidCrystal\_I2C.h>//lcd

#include "ArduinoJson.h"//nhận dl web

//oled

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

//DHT

#include "DHTesp.h"

//khai bao oled

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

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

Adafruit\_SSD1306 oled(SCREEN\_WIDTH, SCREEN\_HEIGHT, &**Wire**, -1);

//DHT

const int DHT\_PIN = 14;

DHTesp dhtSensor;

//ldr

#define ldrPin 36

const float gama = 0.7;

const float rl10 = 50;

//hcsr04

const int trigPin = 16;

const int echoPin = 4;

//define sound speed in cm/uS

#define SOUND\_SPEED 0.034

// #define CM\_TO\_INCH 0.393701

long duration;

float distanceCm;

// float distanceInch;

//BUZZER

#define Buzzer 12

//lcd

LiquidCrystal\_I2C lcd = LiquidCrystal\_I2C(0x27, 20, 4); //lcd

const char \* MQTTServer = "broker.emqx.io";

const char \* MQTT\_Topic = "VLUTE/20004045";

const char \* MQTTID = "";

int ledred = 19;

int muctb = 0; // Biến muctb để lưu giá trị gtri từ web

String ms = "";

String send = "";

WiFiClient espClient;

PubSubClient client(espClient);

IPAddress localIP;

void WIFIConnect() {

**Serial**.println("KET NOI DEN SSID: WOKWI-GUEST");

  WiFi.begin("Wokwi-GUEST", "");

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

    delay(500);

**Serial**.print(".");

  }

**Serial**.print("");

**Serial**.print("WiFi da ket noi");

**Serial**.print(" , Dia chi IP: ");

**Serial**.print(WiFi.localIP());

}

void MQTTReconnect() {

  while (!client.connected()) {

    if (client.connect(MQTTID)) {

**Serial**.println("MQTT lcd: ket noi den VLUTE/20004045");

      lcd.setCursor(2, 3);

      lcd.print("DA KET NOI WIFI!");

      lcd.setCursor(2, 2);

      lcd.print("DA KET NOI MQTT!");

      oled.setCursor(0, 20);

      oled.println("MQTT Connected");

      oled.display();

      client.subscribe(MQTT\_Topic);

    } else {

**Serial**.print(client.state());

      delay(5000);

    }

  }

}

void callback(char\* topic, byte\* message, unsigned int lenght) {

**Serial**.println("da gui den: ");

**Serial**.println(topic);

**Serial**.print("tin nhan: ");

  String stMessage = "";

**Serial**.println();

  for (int i = 0; i < lenght; i++) {

**Serial**.print((char)message[i]);

    stMessage += (char)message[i];

  }

  if (ms != stMessage) {

    ms = stMessage;

**Serial**.println("Received message: " + ms);

    // Cập nhật giá trị muctb từ thông điệp nhận được

    StaticJsonDocument<256> doc;

    deserializeJson(doc, ms);

    // Kiểm tra xem "gtri" có tồn tại không

    if (doc.containsKey("gtri")) {

      if (!doc["gtri"].isNull()) {

        muctb = doc["gtri"];

**Serial**.print("muctb updated to: ");

**Serial**.println(muctb);

      } else {

**Serial**.println("Key 'gtri' tồn tại nhưng có giá trị null.");

      }

    } else {

**Serial**.println("Không tìm thấy key 'gtri' trong JSON data.");

    }

    npublish();

  }

  // lcd.setCursor(0,1);

  // lcd.print(stMessage);

  // delay(1000);

}

void setup() {

  // put your setup code here, to run once:

**Serial**.begin(115200);

**Serial**.println("Hello, ESP32!");

  //WiFi

  WIFIConnect();

  client.setServer(MQTTServer, 1883);

  client.setCallback(callback);

  //hcsr04

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  //BUZZER

  pinMode(Buzzer, OUTPUT);

  //lcd

  lcd.init();

  lcd.backlight();

  pinMode(ledred, OUTPUT);

  //DHT

  dhtSensor.setup(DHT\_PIN, DHTesp::DHT22);

  //oled

  if (!oled.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {

**Serial**.println(F("failed to start SSD1306 OLED"));

    while (1);

  }

  oled.clearDisplay(); // clear display

  oled.setTextSize(1);         // set text size

  oled.setTextColor(WHITE);    // set text color

  // oled.setCursor(0, 5);       // set position to display (x,y)

  // oled.println("KNN IOT"); // set text

  // oled.display();              // display on OLED

  localIP = WiFi.localIP();

  oled.setCursor(0, 0);

  oled.print("WiFi connected");

  oled.setCursor(0, 10);

  oled.print("IP: ");

  oled.print(localIP);

  oled.display();

}

void loop() {

  //hcsr04

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds

  duration = pulseIn(echoPin, HIGH);

  // Calculate the distance

  distanceCm = duration \* SOUND\_SPEED / 2;

  // // Convert to inches

  // distanceInch = distanceCm \* CM\_TO\_INCH;

  // Prints the distance in the Serial Monitor

**Serial**.print("Distance (cm): ");

**Serial**.println(distanceCm);

  // Serial.print("Distance (inch): ");

  // Serial.println(distanceInch);

  //IN LCD

  lcd.setCursor(2, 0);

  lcd.print("KY NANG NGHE IOT");

  //IN DHT

  TempAndHumidity  data = dhtSensor.getTempAndHumidity();

**Serial**.println("Temp: " + String(data.temperature, 2) + "°C");

**Serial**.println("Humidity: " + String(data.humidity, 1) + "%");

  // //IN LDR

  int analogValue = analogRead(ldrPin);

  float voltage = analogValue \* 5 / 4095.0;

  float resistance = 2000 \* voltage / (1 - voltage / 5);

  float kecerahan = pow(rl10 \* 1e3 \* pow(10, gama) / resistance, (1 / gama));

**Serial**.print("Kecerahan = ");

**Serial**.println(kecerahan);

  //BUZZER

  // digitalWrite(Buzzer, HIGH);

  // tone(Buzzer, 262, 1000);

  // delay(1000);

  // digitalWrite(Buzzer, LOW);

  // noTone(Buzzer);

  // delay(1000);

  //GUI MQTT

  String sendData = "{\"nhietdo\":\"" + String(data.temperature) + "\",\"doam\":\"" + String(data.humidity) + "\"}";

  client.publish(MQTT\_Topic, sendData.c\_str());

  if (!client.connected()) {

    MQTTReconnect();

  }

  client.loop();

  delay(1000); // this speeds up the simulation

}

void npublish()

{

  lcd.setCursor(0, 1);

  lcd.print(muctb);

  if (muctb != 0) {

    digitalWrite(ledred, HIGH);

    // oled.clearDisplay(); // clear display

    oled.setTextSize(1);         // set text size

    oled.setTextColor(WHITE);    // set text color

    oled.setCursor(1, 2);       // set position to display (x,y)

    oled.println("KET NOI WEB THANH CONG"); // set text

    oled.display();              // display on OLED

  } else {

    digitalWrite(ledred, LOW);

  }

}

<?php

$conn = mysqli\_connect("localhost", "root", "", "plant\_disease");

if (!$conn) {

    die("thất bại: " . mysqli\_connect\_error());

} else {

    echo "thành công";

}

$sql = "SELECT \* FROM iot";

$data = mysqli\_query($conn, $sql);

while ($row = mysqli\_fetch\_assoc($data)) {

    // echo $row['name'];

    // echo $row['status'];

    // Lưu giá trị cuối cùng vào biến $lastStatus

    $lastStatus = $row['status'];

}

// Gán giá trị cuối cùng vào biến JavaScript gtri

echo "<script>var gtri = " . json\_encode($lastStatus) . ";</script>";

?>

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js" type="text/javascript"></script>

    <script src="https://code.jquery.com/jquery-3.6.1.min.js" type="text/javascript"></script>

</head>

<body>

  <h2 style="text-align: center; ">HỆ THỐNG ĐO NHIỆT ĐỘ - ĐỘ ẨM</h2><br>

        <label>Nhiệt độ hiện tại: </label><label id="nhietdo"></label> <br>

        <label>Độ ẩm hiện tại: </label><label id="doam"></label> <br>

    <table>

        <tr>

            status: <td id="gtri"><?php echo $lastStatus ?></td>

        </tr>

    </table>

    <script>

        var gtri = document.getElementById("gtri").innerText;

        var dulieu = {

            'gtri': gtri

        }

        var chuoi = JSON.stringify(dulieu);

        alert(chuoi);

        var client = new Paho.MQTT.Client("broker.emqx.io", Number(8083), "");

        client.onConnectionLost = onConnectionLost;

        client.onMessageArrived = onMessageArrived;

        client.connect({

            onSuccess: onConnect

        });

        function onConnect() {

            console.log("onConnect");

            client.subscribe("VLUTE/20004045");

            // Gửi thông điệp MQTT khi kết nối thành công

            sendMessage();

        }

        function onConnectionLost(responseObject) {

            if (responseObject.errorCode !== 0) {

                console.log("onConnectionLost:" + responseObject.errorMessage);

            }

        }

        function onMessageArrived(message) {

            console.log("onMessageArrived:" + message.payloadString);

            const obj = JSON.parse(message.payloadString);

            console.log(obj.send)

            // Xử lý thông điệp MQTT

            // var nhietdotb = parseFloat(obj.nhietdo);

            // var nhietdotb = parseFloat(obj.nhietdo);

          document.getElementById("nhietdo").innerHTML = obj.nhietdo + " độ C";

          document.getElementById("doam").innerHTML = obj.doam + " %";

        }

        function sendMessage() {

            var message = new Paho.MQTT.Message(chuoi);

            message.destinationName = "VLUTE/20004045";

            client.send(message);

        }

    </script>

</body>

</html>

!pip install imutils

from tensorflow.keras.preprocessing.image import img\_to\_array

from tensorflow.keras.models import load\_model

import numpy as np

import mimetypes

import argparse

import imutils

import pickle

import cv2

import os

from PIL import Image

model = load\_model('./20004045\_model.h5')

from PIL import Image

from keras.preprocessing.image import img\_to\_array

image = Image.open('./moth (2).jpg')

image = image.resize((112, 112))

image = img\_to\_array(image) / 255.0

image = np.expand\_dims(image, axis=0)

(boxPreds, labelPreds) = model.predict(image)

lb = ['beetle', 'grasshopper', 'mango\_hopper', 'mango\_mealybug', 'moth']

i = np.argmax(labelPreds, axis=1)

result = lb[np.argmax(labelPreds)]

accuracy = float(labelPreds[0][i]) \* 100

print(accuracy)

print(result)

# Lấy tên đã dự đoán

predicted\_name = result

status = None

# Kiểm tra điều kiện

if predicted\_name == "beetle" or predicted\_name == "moth":

    status = "1"

else:

    status = "0"

!pip install mysql-connector

import mysql.connector

name = str(predicted\_name)

st = status

mydb = mysql.connector.connect(

    host="localhost",

    user="root",

    password="",

    database="plant\_disease"

)

mycursor = mydb.cursor()

sql = "INSERT INTO iot (name,status) VALUES (%s,%s)"

val = (name,st)

mycursor.execute(sql, val)

mydb.commit()

print(mycursor.rowcount, "record inserted.")