**FPT UNIVERSITY**

**Vườn Sao Băng**

Member:

1. Huỳnh Đức Chính – DE160540
2. Nguyễn Sỹ Đức – DE160488

***DANANG – Jul 10, 2023***

Table of Contents

[INTRODUCTION 6](#_Toc24050550)

[PROCESS DIAGRAM 6](#_Toc24050551)

[FLOWCHART 6](#_Toc24050552)

[RESULT 6](#_Toc24050553)

[CONCLUSION 6](#_Toc24050554)

[REFERENCE 6](#_Toc24050555)

# INTRODUCTION

* In this part, you need to short description about project:
  + What is the problem?

+ Năm 3000 môi trường trái đất bị hủy hoại nghiêm trọng, tài nguyên cạn kiệt, con người không còn đủ chổ để sinh sống và trồng trọt. Khi đó họ cần phải tìm một nơi (vùng đất mới) để tiếp tục sinh sống, và môi trường lý tưởng nhất của con người là sao hỏa.

+ Tiếp tục đề tài, chúng xem xin trình bài dự án khu vườn khép kín (vườn sao băng), nơi mà mọi thứ được bán tự động hóa thông qua internet, giúp tiết kiệm sức người và nâng cao năng suất, hiệu quả lao động.

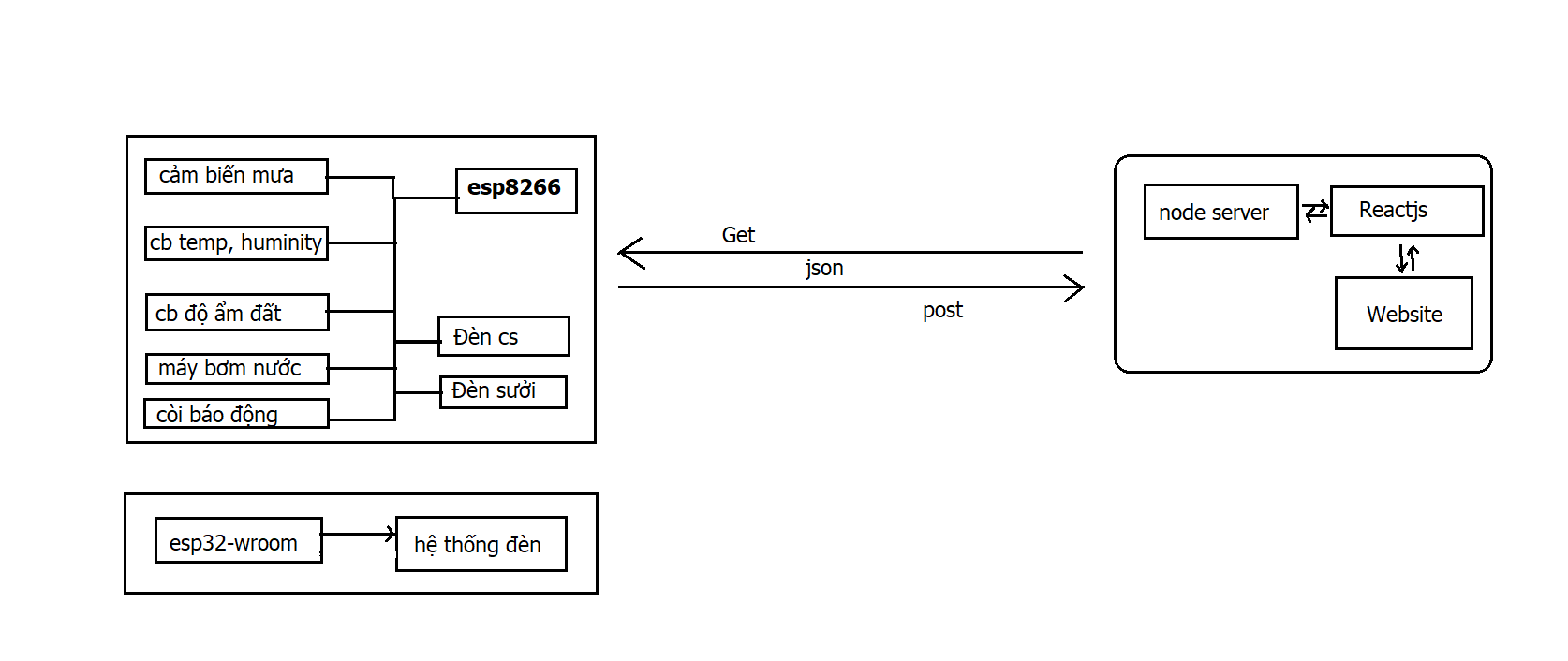
* + What are the devices used in the project?
    - Led
    - 1 esp8266, 1 esp32
    - 1 arduino
    - 1 RTC I2C
    - 2 Relay
    - 1 Cảm biến mưa
    - 1 Cảm biến nhiệt độ độ ẩm không khí
    - 1 Cảm biến độ ẩm đất
    - 1 loa
    - 1 motor (bơm nước)
    - 1 Cảm biến ánh sáng
  + What are characteristics and role of each device in your design?
    - RTC I2C: module thời gian thực
    - Relay: công tắc bật ngắt dòng diện
    - Cảm biến mưa: nhận diện mưa (1,0)
    - Cảm biến nhiệt độ độ ẩm không khí: sử dụng thư viện DHT.h đọc kết quả.
    - Cảm biến độ ẩm đất: đọc tín hiệu analog trả về giá trị 0 – 1023 map (0 – 100)
    - Cảm biến ánh sáng: sử dụng tín hiệu digital ( khi có ánh sáng điện trở giảm).

# PROCESS DIAGRAM

* In this part, you draw the process diagram and provide design in Tinkercad or Fritzing

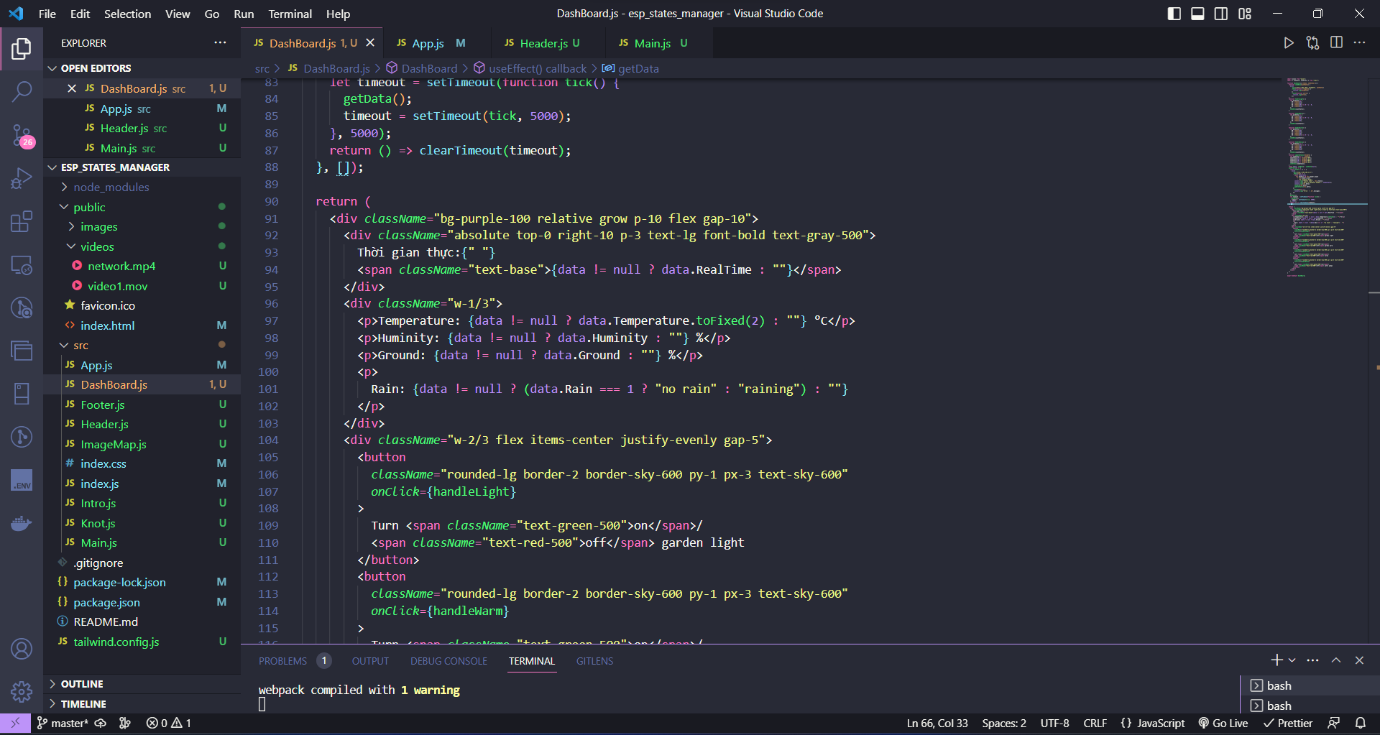
# FLOWCHART

* You have to draw the flowchart of code here



# RESULT

* **Copy Source code in here**

****

#include <ESP8266WiFi.h>

#include <ESP8266WebServer.h>

#include <ArduinoJson.h>

#include <DHT.h>

#include <RTClib.h>

#define pinGround A0

#define pinSiren D8

#define pinDHT D6

#define pinRain D7

#define pinRelay D0

#define pinGardenLight D4

#define pinGardenWarm D3

// statusPins = [D8, D0, D4, D3]; pinSiren - pinRelay - pinGardenLight - pinGardenWarn

#define DHTTYPE DHT11

DHT dht(pinDHT, DHTTYPE);

RTC\_DS1307 DS1307\_RTC;

const char\* ssid = "LE THI NHUNG";

const char\* password = "lethinhung123";

ESP8266WebServer server(80);

char Week\_days[7][12] = { "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" };

float t = 0.0; // temperature

float h = 0.0; // humidity

int g = 0;

int r = 0;

String realTime = "";

void handleRoot() {

StaticJsonDocument<200> response;

response["Temperature"] = t;

response["Huminity"] = h;

response["Ground"] = g;

response["Rain"] = r;

response["RealTime"] = realTime;

response["D0"] = digitalRead(pinRelay);

response["D3"] = digitalRead(pinGardenWarm);

response["D4"] = digitalRead(pinGardenLight);

response["D8"] = digitalRead(pinSiren);

// Serialize the response JSON to a string

String jsonResponse;

serializeJson(response, jsonResponse);

// Send the response JSON

server.sendHeader("Access-Control-Allow-Origin", "\*");

server.sendHeader("Content-Type", "application/json");

server.send(200, "application/json", jsonResponse);

}

void handleJson() {

DynamicJsonDocument doc(1024);

DeserializationError error = deserializeJson(doc, server.arg("plain"));

if (error) {

server.send(400, "text/plain", "Bad Request");

} else {

StaticJsonDocument<200> response;

digitalWrite(D0, doc["D0"]);

digitalWrite(D3, doc["D3"]);

digitalWrite(D4, doc["D4"]);

digitalWrite(D8, doc["D8"]);

response["D0"] = doc["D0"];

response["D3"] = doc["D3"];

response["D4"] = doc["D4"];

response["D8"] = doc["D8"];

// Serialize the response JSON to a string

String jsonResponse;

serializeJson(response, jsonResponse);

// Send the response JSON

server.sendHeader("Access-Control-Allow-Origin", "\*");

server.sendHeader("Content-Type", "application/json");

server.send(200, "application/json", jsonResponse);

}

}

void handleOptions() {

server.sendHeader("Access-Control-Allow-Origin", "\*");

server.sendHeader("Access-Control-Allow-Methods", "POST,GET,OPTIONS");

server.sendHeader("Access-Control-Allow-Headers", "Content-Type");

server.send(200);

}

void setup() {

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

dht.begin();

Serial.begin(9600);

DS1307\_RTC.begin();

delay(500);

DS1307\_RTC.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));

pinMode(pinGround, INPUT);

pinMode(pinSiren, OUTPUT);

pinMode(pinRain, INPUT);

pinMode(pinGardenLight, OUTPUT);

pinMode(pinGardenWarm, OUTPUT);

pinMode(pinRelay, OUTPUT);

digitalWrite(pinRelay, LOW);

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

Serial.begin(9600);

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

delay(500);

Serial.println("Connecting to WiFi...");

}

Serial.println("");

Serial.print("Connected to Wifi ");

Serial.println(ssid);

Serial.print("IP Address: ");

Serial.println(WiFi.localIP());

//-------------------- phat wifi

server.on("/", HTTP\_OPTIONS, handleOptions);

server.on("/", HTTP\_GET, handleRoot);

server.on("/update", HTTP\_OPTIONS, handleOptions);

server.on("/update", HTTP\_POST, handleJson);

server.begin();

Serial.println("HTTP server started");

}

void loop() {

// put your main code here, to run repeatedly:

server.handleClient();

DateTime now = DS1307\_RTC.now();

realTime = String(now.year()) + "/" + String(now.month()) + "/" + String(now.day()) + " (" + String(Week\_days[now.dayOfTheWeek()]) + ") " + String(now.hour()) + ":" + String(now.minute()) + ":" + String(now.second());

// Serial.print(now.year(), DEC);

// Serial.print('/');

// Serial.print(now.month(), DEC);

// Serial.print('/');

// Serial.print(now.day(), DEC);

// Serial.print(" (");

// Serial.print(Week\_days[now.dayOfTheWeek()]);

// Serial.print(") ");

// Serial.print(now.hour(), DEC);

// Serial.print(':');

// Serial.print(now.minute(), DEC);

// Serial.print(':');

// Serial.print(now.second(), DEC);

// Serial.println();

// Serial.print(" since midnight 1/1/1970 = ");

// Serial.print(now.unixtime());

// Serial.print("s = ");

// Serial.print(now.unixtime() / 86400L);

// Serial.println("d");

// Serial.println();

// Serial.println();

g = 100 - map(analogRead(pinGround), 0, 1023, 0, 100);

if (g > 50) { // chỉnh lại

// digitalWrite(pinRelay, HIGH);

} else {

// digitalWrite(pinRelay, LOW);

}

// Serial.print("Analog Ground: ");

// Serial.println(g);

r = digitalRead(pinRain);

// Serial.print("Digital Rain: ");

// Serial.println(r);

float newT = dht.readTemperature();

if (isnan(newT)) {

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

} else {

t = newT;

}

// Serial.print("Temperature: ");

// Serial.println(t);

float newH = dht.readHumidity();

if (isnan(newH)) {

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

} else {

h = newH;

}

// Serial.print("Huminity: ");

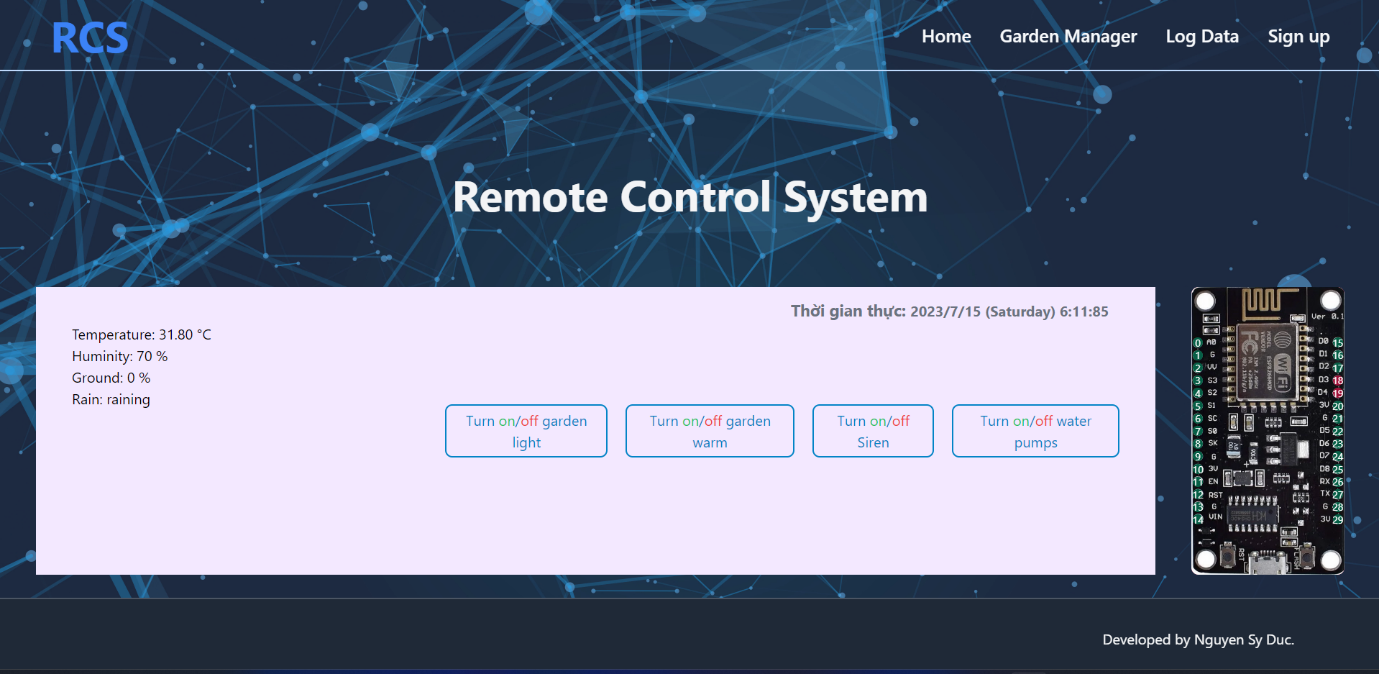
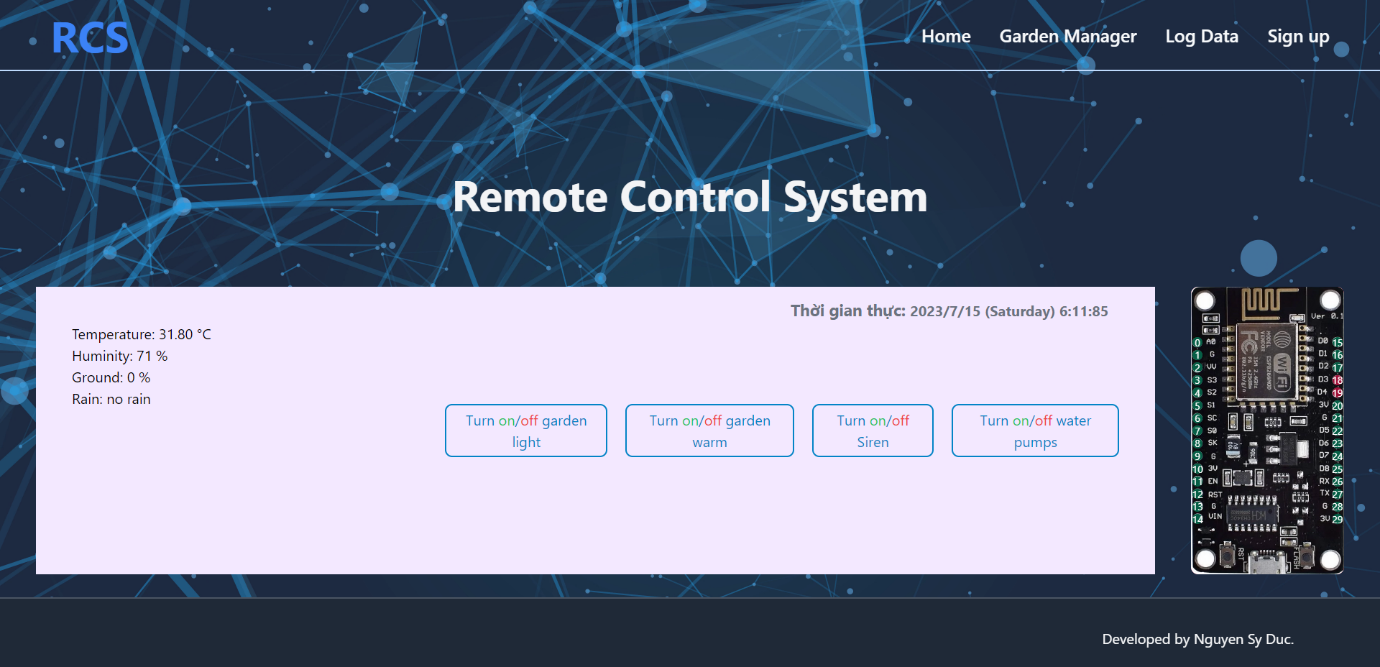
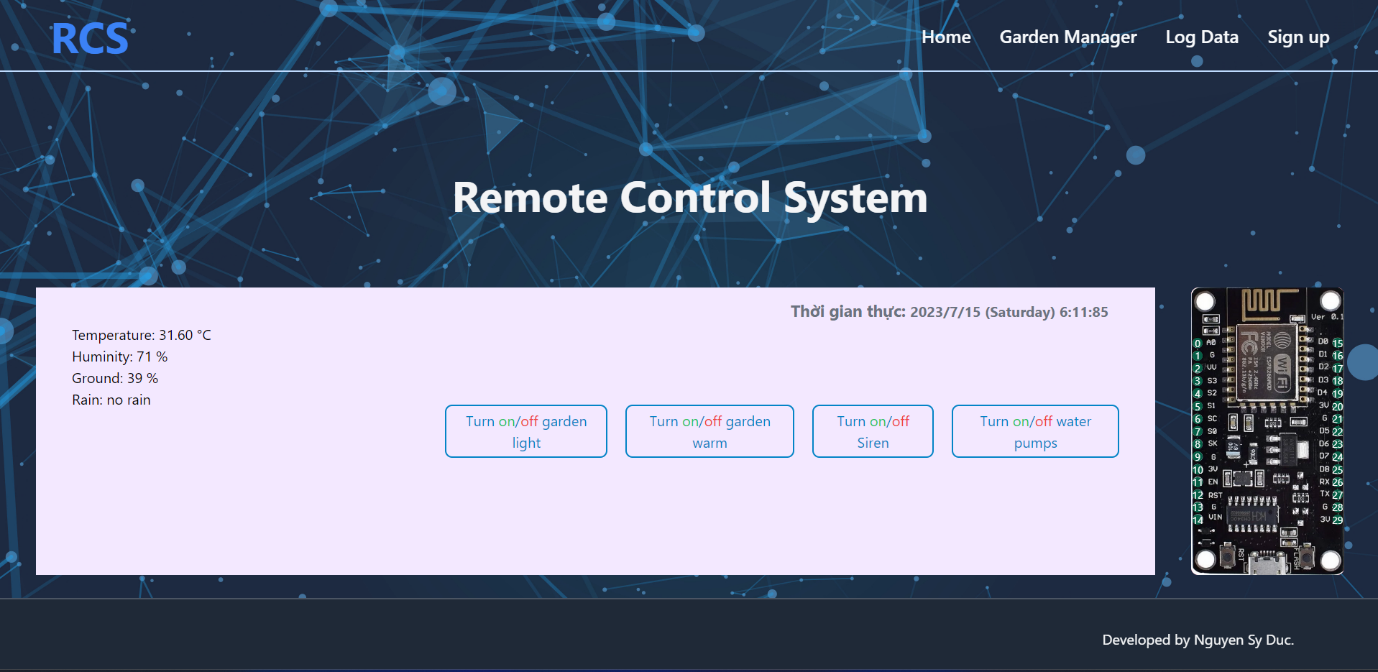
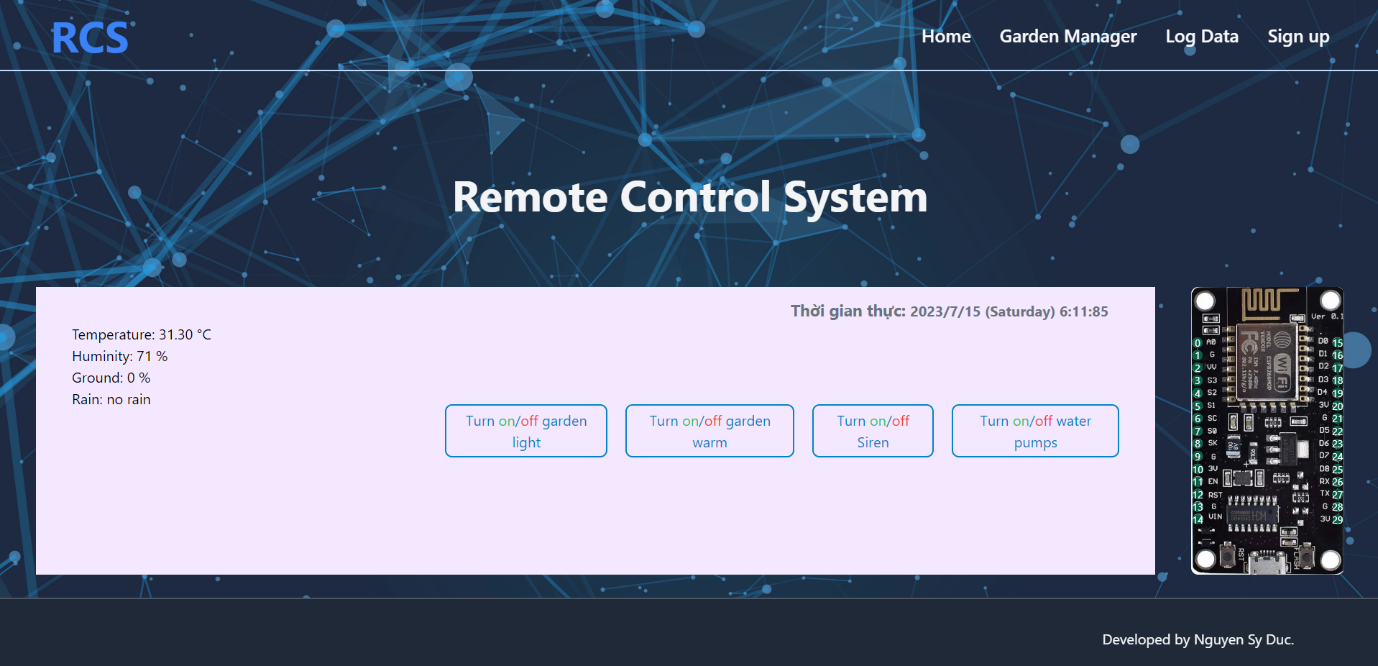
// Serial.println(h);

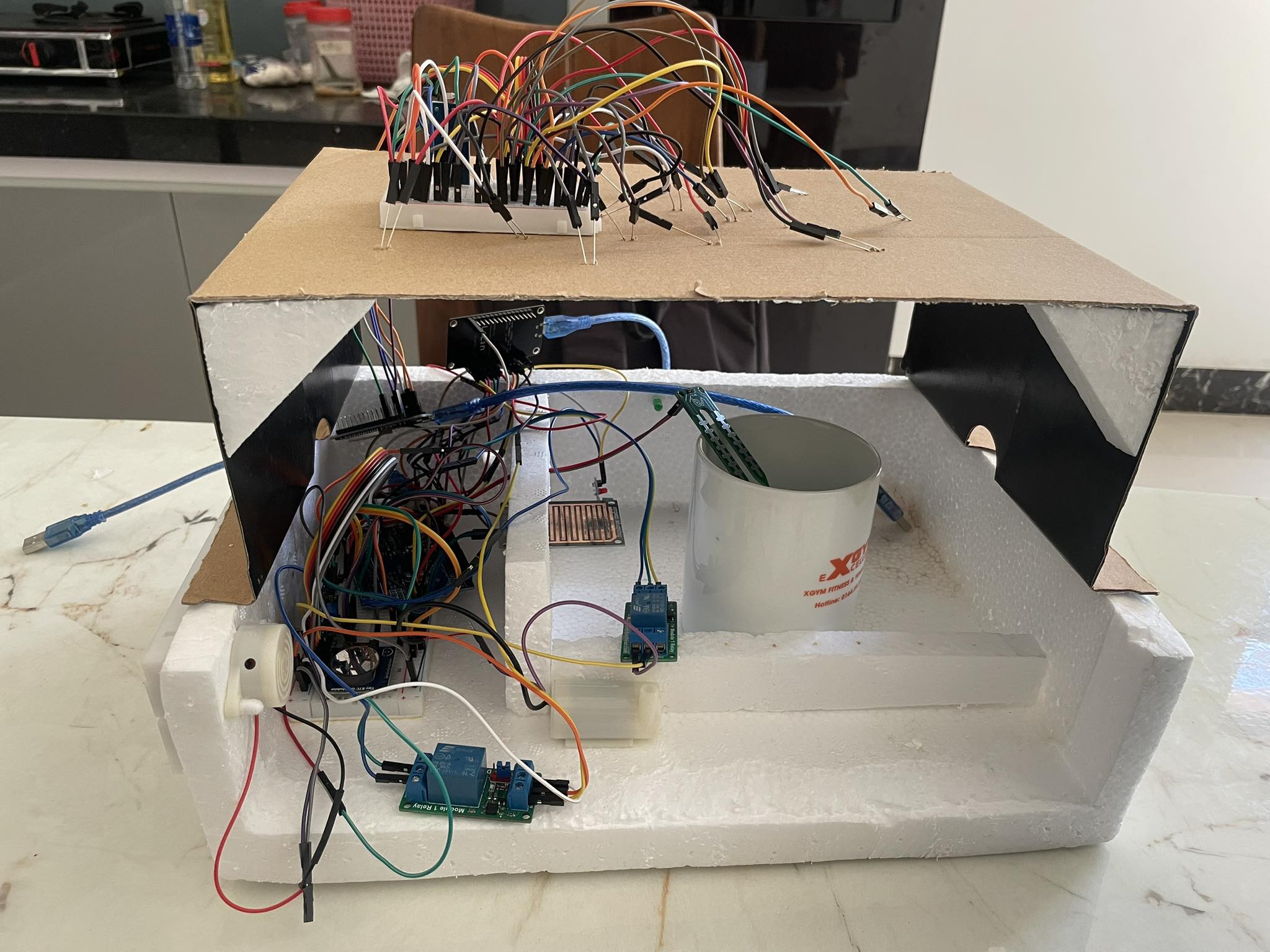
// Serial.println("----------------------------------------");

delay(50);

}

* **Show the picture of your project here**





# CONCLUSION

* Give some ideas about problem and how to improved it.
  + - Tối ưu lại đường truyền thiết bị mạng để việc truyền tải diễn ra chính xác và nhanh hơn.
    - Xây dựng cơ chế xác thực đăng nhâp để giữ an toàn cho khu vườn
* What have you had learn when finish project:
  + Skill program
    - Lâp trình Arduino, esp
    - Dựng cơ chế liên lạc qua mạng wifi
    - Xử lý lỗi và fix bug (stackoverflow, doc, …)
  + Team building
    - Kỹ năng làm việc, phối hợp nhóm
    - Phân chia công việc
    - Kỹ năng giải quyết vấn đề

* + Communication together.
    - Kỹ năng trình bày hiệu quả
    - Tranh luận và giải quyết vấn đề
    - Chọn lọc và đưa ra ý tưởng

# REFERENCE

* <https://www.edx.org/>
* <https://arduino-esp8266.readthedocs.io/en/latest/>
* <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/>
* <https://www.analog.com/media/en/technical-documentation/data-sheets/DS1307.pdf>

THANK YOU