

CASE STUDY 1: XÂY DỰNG HỆ THỐNG ĐIỀU KHIỂN DÀN RAU THỎ CANH BẰNG BLYNK

1. Mô tả bài toán

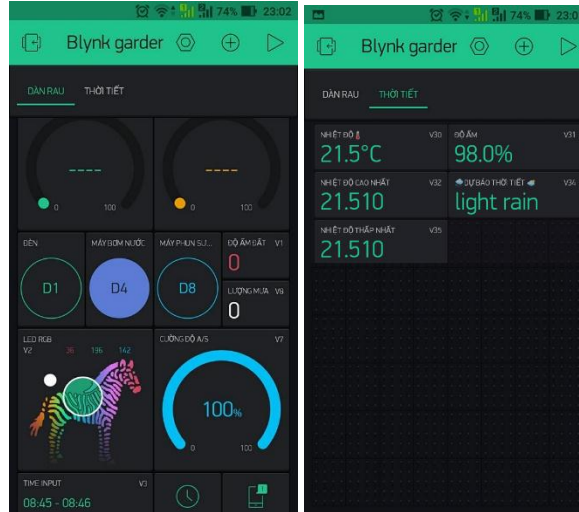
Xây dựng hệ thống điều khiển dàn rau tự động nhằm mục đích điều khiển các thông số để dàn rau có thể dễ dàng phát triển một cách tốt nhất hạn chế các tác động từ môi trường gây ảnh hưởng đến quá trình sinh trưởng cũng như phát triển của cây. Để xây dựng một hệ thống dàn rau điều khiển tự động trước hết phải thu thập số liệu các tác nhân từ môi trường và điều khiển các thiết bị điều chỉnh các thông số đó như, tăng giảm độ ẩm đất bằng cách bật máy bơm, đo lượng mưa, đo cường độ ánh sáng để bật đèn nhằm tăng khả năng sinh trưởng hay bật bóng đèn màu vào từng thời điểm để cây hấp thụ ánh sáng tốt. Để giải quyết vấn đề đó cần các thiết bị hiển thị dữ liệu của các tác nhân sau đó lập các tập lệnh để thực hiện các yêu cầu đã đặt ra.



Mô hình dàn rau thổ canh

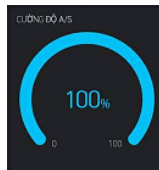
Các chức năng cần thực hiện:

Hệ thống được thiết kế trên ứng dụng Blynk dành cho thiết bị di động. Ứng dụng này điều khiển qua tài khoản được tạo gồm tên, mật khẩu và một token để phân cứng gửi dữ liệu đến cũng như gửi các tác động từ ứng dụng đến hệ thống phân cứng.

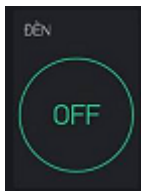


Giao diện hệ thống điều khiển dàn rau thủy canh

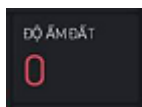
Hệ thống này gồm 2 phần: Phần 1 là thông tin dữ liệu và điều khiển dàn rau thủy canh, phần 2 là phần thông tin thời tiết trong ngày, hệ thống này giúp cho người sử dụng có thể biết được thời tiết để chăm sóc cây trồng tốt hơn.



Hệ thống sử dụng hai các thanh Gauge dùng để hiển thị thông số nhiệt độ, độ ẩm của không khí cũng như theo dõi tình hình của hệ thống.



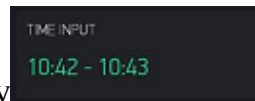
Các nút này dùng để điều khiển các chân vật lý tác động đến các relay bật tắt các thiết bị như đèn, van nước, phun sương. Các nút này cho phép người dùng có thể tự điều khiển.



Khung dữ liệu này được sử dụng để hiển thị thông số các biến như độ ẩm đất, lượng mưa từ các chân ảo đọc dữ liệu từ cảm biến.



Bộ điều khiển này cho phép người dùng tương tác trực tiếp đến hệ thống đèn leds, hệ thống này nhận dữ liệu khi người dùng tương tác sẽ gửi dữ liệu ba màu Red, Green, Blue từ vị trí mà người dùng tương tác đó. Sau đó gửi đến hệ thống đèn và điều khiển đèn màu thông qua dữ liệu ba màu đã được nhận.



Hệ thống hẹn giờ được sử dụng qua chức năng này, từ các biến thời gian mà người dùng thiết lập giờ, phút bật và giờ phút tắt từ đó hệ thống sẽ bật tắt theo dữ liệu đó và lặp lại liên tục theo ngày.



Tất cả dữ liệu ở phần 2 để sử dụng làm khung lưu trữ dữ liệu.

- Sau khi đã tìm hiểu được các thiết bị thì cần lập trình điều khiển để cho hệ thống hiện thị dữ liệu của các cảm biến, hay điều khiển relay các thiết bị khác.
- Xây dựng chức năng hiển thị các thông số từ các thiết bị cảm biến đo được. Để xây dựng các chức năng đó trước hết cần khai báo các thư viện có liên quan đến các thiết bị hệ thống.

```
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <WiFiClientSecure.h>
#include <DHT.h>
#include <WiFiUdp.h>
#include <NTPClient.h> // thư viện time
// den leds -----
#include <Adafruit_NeoPixel.h>
```

Để giải quyết vấn đề nhiệt độ, độ ẩm cần sử dụng thiết bị là DHT11, thiết bị này sử dụng để đọc nhiệt độ, độ ẩm không khí thu thập dữ liệu và thông tin được lấy thông qua

chân Digital (D0) trên thiết bị, chân GND là chân đất, chân VCC dùng để cung cấp dòng điện (3V) giúp cho thiết bị có thể hoạt động. Sử dụng thư viện DHT.h để đọc dữ liệu.

Độ ẩm đất được đọc qua một thiết bị đo độ ẩm kết nối đến một module cảm biến, thiết bị này sẽ đọc được độ ẩm thông qua việc cắm đầu dò độ ẩm xuống đất đất càng ướt thì độ ẩm càng cao. Lấy dữ liệu thông qua chân Analog (A0) chân GND và VCC để cung cấp nguồn điện. Dữ liệu được đọc trong khoảng từ 0 đến 1023, sau đó sử dụng một hàm chuyển đổi từ 0 đến 1023 thành 0 đến 100 để lấy thông tin theo %.

Đo lượng mưa cũng giống như đo độ ẩm đất, sử dụng thiết bị đo lượng mưa nhưng thiết bị được đặt trên mặt đất để lấy giá trị khi nước mưa rơi xuống thiết bị. Thiết bị cũng lấy dữ liệu qua chân A0 và chân GND, VCC để cung cấp nguồn điện. Dữ liệu được đọc trong khoảng từ 0 đến 1023, sau đó sử dụng một hàm chuyển đổi từ 0 đến 1023 thành 0 đến 100 để lấy thông tin theo %.

Thiết bị đo cường độ ánh sáng photodiode sử dụng để đo cường độ ánh sáng, đọc dữ liệu thông qua chân A0 sau đó chuyển dữ liệu số từ 0 đến 1023 sang 0 đến 100 %.

Tất cả dữ liệu từ các thiết bị được thiết bị là ESP8266 được thu thập thông qua việc kết nối các chân thiết bị vào các chân trong ESP8266, thông qua phương thức truyền tín hiệu wifi để gửi tín hiệu lên server Blynk sau đó nhận dữ liệu lên ứng dụng Blynk trên thiết bị di động để điều khiển và kiểm soát. Trong ứng dụng, Blynk được sử dụng như công cụ để hiển thị và điều khiển hệ thống. Ngoài ra hệ thống này còn hiển thị thông tin về dự báo thời tiết, giúp cho người dùng nắm bắt được thời tiết thông qua hệ thống điều khiển đó.

2. Phân tích hệ thống

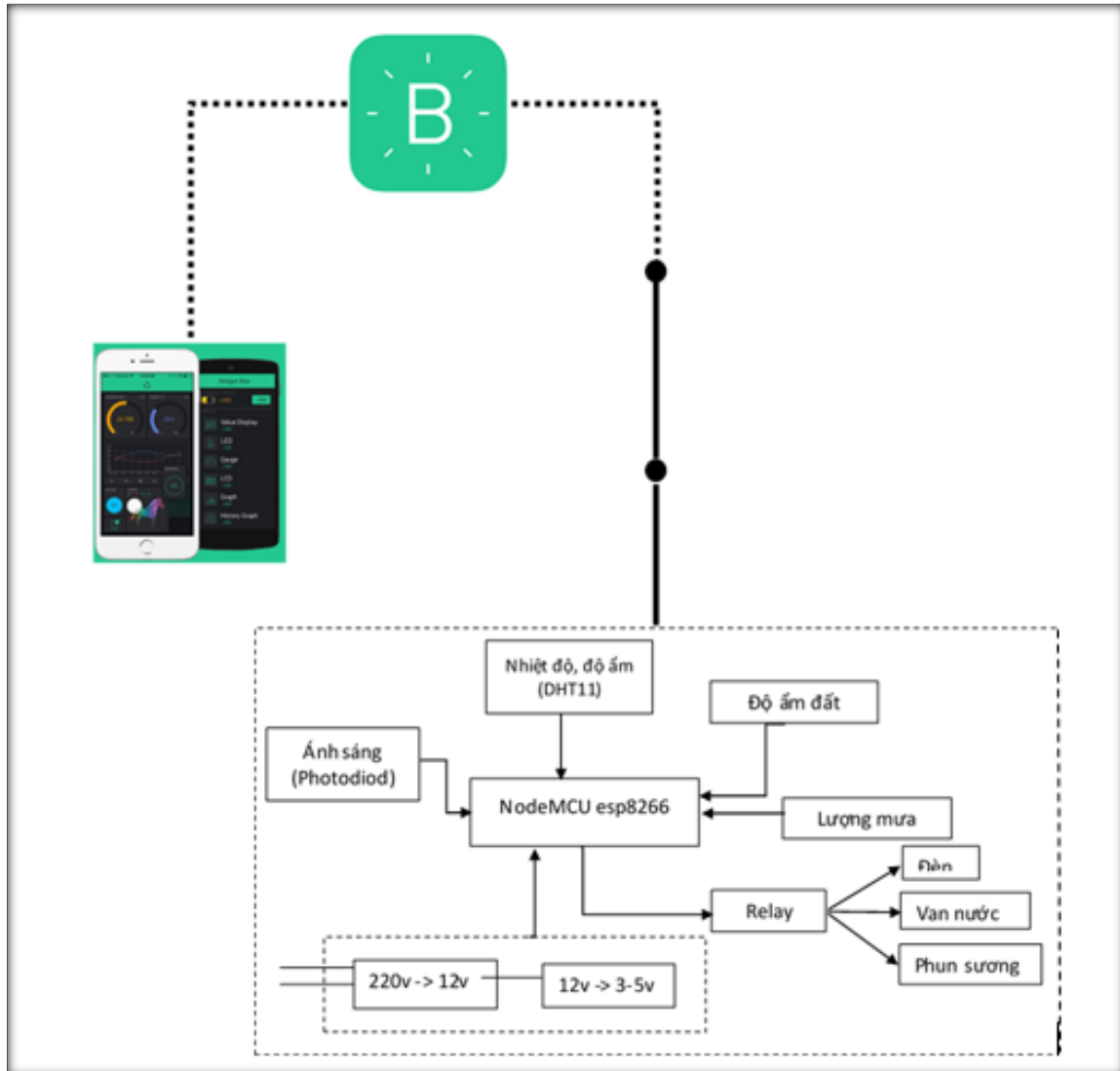
a. Mô tả thiết bị

Trước hết để hệ thống hoàn thiện thì cần có đầy đủ các thiết bị cũng như các kiến thức cơ bản để sử dụng các thiết bị đó thì chúng ta mới có thể xây dựng hệ thống hoàn thiện được. Các thiết bị được sử dụng trong hệ thống này là:

- Dht11: thiết bị này được dùng để đo nhiệt độ, độ ẩm.
- Photodiode : thiết bị dùng để đo cường độ ánh sáng.
- Cảm biến độ ẩm đất : dùng để đo độ ẩm đất.
- Cảm biến lượng mưa: dùng để đo lượng mưa.
- Mạch hạ áp: hạ áp điện từ 220v xuống 3-5v để cung cấp điện cho tất cả các thiết bị sử dụng điện áp 3A. Hệ thống mạch hạ áp này chia làm 2 phần, phần 1 là hạ áp từ 220v đến 12v, phần 2 là hạ áp từ 12v đến 3-5v.
- Relay: dùng để bật tắt các thiết bị như là bật tắt bóng đèn, van nước, máy phun sương

b. Sơ đồ hệ thống

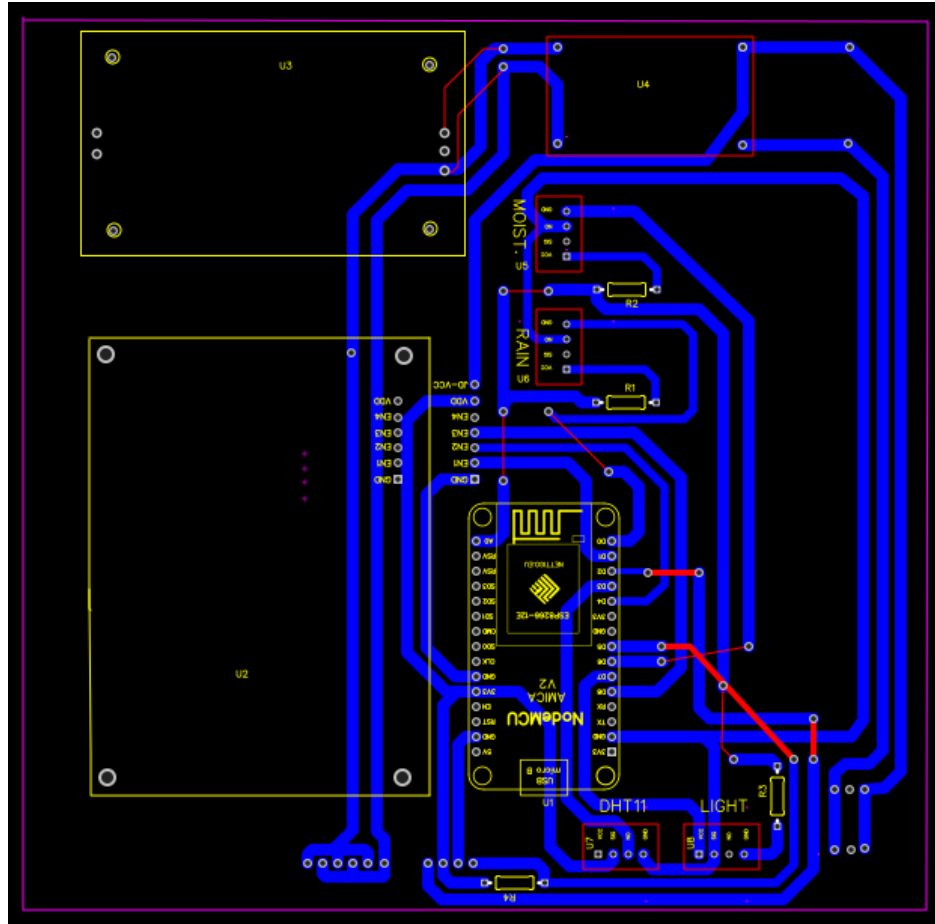
Từ các thiết bị được nêu ở phần trên chúng ta xây dựng được một hệ thống điều khiển dàn rau thổ canh. Hệ thống dàn rau thổ canh được xây dựng dựa trên sơ đồ logic của bảng mạch. Hệ thống này gồm 3 phần, phần cứng (thiết bị, thư viện, mạng kết nối), server Blynk và ứng dụng Blynk trên thiết bị di động.



Hệ thống dàn rau thổ canh

c. Mạch PBC

Dựa trên sơ đồ hệ thống và các thiết lập thiết bị điều khiển phần cứng chúng ta xây dựng được hệ thống mạch PBC. Trong đó thể hiện sự kết nối giữa các thiết bị với nhau để tạo thành một bảng mạch. Các chân nối của các thiết bị được nối trực tiếp đến esp8266.



Sơ đồ PBC của dàn thủy canh

i. Phương pháp thực hiện

Để xây dựng hệ thống điều khiển dàn rau thủy canh trước tiên cần xây dựng bo mạch điều khiển và lấy dữ liệu từ các thiết bị cảm biến dựa trên các chức năng đã được xây dựng ở phần 3.1 thì chúng ta tiến hành xây dựng hệ thống tự động cho dàn rau. Trong hệ thống các thiết bị cảm biến được đọc và sử dụng chân ảo để gửi dữ liệu như chân V5, V6 để gửi thông số nhiệt độ, độ ẩm và sử dụng thiết bị đo (phần 2.7.12) để hiển thị dữ liệu; chân V1, V7, V8 để lưu trữ dữ liệu độ ẩm đất, cường độ ánh sáng, lượng mưa và sử dụng thiết bị hiển thị giá trị (phần 2.7.12). Sử dụng các chân D1, D4, D8 để điều khiển relay bật tắt bằng nút button (phần 2.7.11). Các cách xây dựng dưới đây nằm trong phần xây dựng chương trình trong phần 3.1.9.


```
//--- đo cường độ ánh sáng
digitalWrite(photodiod,HIGH); //pin[6], HIGH);
delay(500);
sensorValue2 = analogRead(A0);
sensorValue2 = constrain(sensorValue2, 0, 1023);
sensorValue2 = map(sensorValue2, 0, 1023, 100, 0);
Blynk.virtualWrite(V7, sensorValue2); // cường độ ánh sáng
digitalWrite(photodiod,LOW);
//---
//-- lượng mưa
digitalWrite(luongmua,HIGH); //pin[6], HIGH);
delay(500);
sensorValue3 = analogRead(A0);
sensorValue3 = constrain(sensorValue3, 552, 915); //915, 552
sensorValue3 = map(sensorValue3, 552, 915, 100, 0);
Blynk.virtualWrite(V8, sensorValue3); // lượng mưa
digitalWrite(luongmua,LOW);
//---
```

- **Hẹn giờ bật tắt relay**

Sau khi thiết lập các dữ liệu từ thiết bị chúng ta thực hiện chức năng hẹn giờ bật tắt máy bơm. Chức năng này sử dụng bộ hẹn giờ (phần 2.7.11 và phần 3.1.8) để thiết lập hệ thống hẹn giờ. Sau khi thiết lập dữ liệu lấy từ server về ta tiến hành thiết lập hệ thống.

```
void hengio()
{
    int timehour;
    int timeminute;
    timehour=(n.getEpochTime()/86400L)/3600;
    timeminute=(n.getEpochTime()/3600)/60;
    if((time_hour==timehour)&&(time_minute==timeminute))
    {
        digitalWrite(D8, LOW);
    }
    if((time_hourstop==timehour)&&(time_minutestop==timeminute))
    {
        digitalWrite(D8, HIGH);
    }
}
```

Chương trình này thiết lập khoảng thời gian thực sau đó so sánh thời gian thực với thời gian mà người dùng hẹn, nếu như thời gian của hệ thống bằng thời gian hẹn thì hệ thống sẽ tự động bật/ tắt máy bơm.

*

- **Xây dựng điều khiển hệ thống đèn màu:**

Hệ thống này xong xây dựng đèn leds đổi màu, dựa vào các dữ liệu màu mà người dùng điều khiển thiết bị bộ điều khiển **ZeRGBa** (phần 2.7.11) để điều khiển hệ thống màu. Mỗi lần điều khiển hệ thống sẽ gửi dữ liệu ba màu Red, Green, Blue qua chân V2 về phần cứng. Sau đó chúng ta sử dụng các biến dữ liệu đó để gửi dữ liệu đến đèn màu và bật đèn theo dữ liệu đó.

```
BLYNK_WRITE(V2)
{
    int pinValue = param[0].asInt(); //
    int pinValue1 = param[1].asInt();
    int pinValue2 = param[2].asInt();
    Serial.print("Red value is: ");
    Serial.println(pinValue);
    red = pinValue;
    Serial.print("Blue value is: ");
    Serial.println(pinValue2);
    blue = pinValue2;
    Serial.print("Green value is: ");
    Serial.println(pinValue1);
    green = pinValue1;
}
```

Sau khi lấy dữ liệu từ server thì chúng ta thiết lập cấu hình sử dụng đèn màu.

```
// den leds -----
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
    #include <avr/power.h>
#endif
#define PWM_VALUE 63 // số đèn leds
WiFiUDP Udp; // khai báo tự động nhận mạng wifi

#define PIN D2 //for LEDs
#define NUMPIXELS 50 // How many NeoPixels are attached to the Arduino?
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
int red = 0;
int green = 0;
int blue = 0;
```

Tiếp đến chúng ta xây dựng chương trình chạy cho đèn leds với chân D2.

```
void denleds()
{
    for(int i=0;i<NUMPIXELS;i++){
        // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
        pixels.setPixelColor(i, pixels.Color(red,green,blue)); // Moderately bright green color.
    }
    pixels.show(); // This sends the updated pixel color to the hardware.
}
```

- **Xây dựng hệ thống hiển thị dự báo thời tiết**

Sau khi thực hiện hệ thống điều khiển chúng ta tiếp tục thiết lập hệ thống thông báo thời tiết cho người dùng. Trước hết cần tạo tài khoản và thiết lập kết nối đến hệ thống.

```
WidgetLCD lcd(V4);
String APIKEY = "c2d6e255caff443e1383e44ddl5f5927";
String CityID = "1586896"; //rabat malta
WiFiClient client;
char servername[] = "api.openweathermap.org"; // remote
String result;
//--- Weather
char auth[] = "8016f70dl70746eeaf885bae4e08dc5c";
```

Sau đó thiết lập gửi dữ liệu chân ảo lên server.

```
void weathear() {
    Blynk.virtualWrite(V30, Temperature);
    Blynk.virtualWrite(V31, Humidity);
    Blynk.virtualWrite(V35, temp_min);
    Blynk.virtualWrite(V32, temp_max);
    Blynk.virtualWrite(V34, weatherDescription);
    count++;
}
```

Khi đã thiết lập được các chân ảo nhận dữ liệu thì chúng ta thiết lập chương trình lấy dữ liệu từ openweathermap.

```
void getWeatherData() //client function to send/receive GET request data.
{
    if (client.connect(servername, 80)) { //starts client connection, checks for connection
        client.println("GET /data/2.5/weather?id=" + CityID + "&units=metrics&APPID=" + APIKEY);
        client.println("Host: api.openweathermap.org");
        client.println("language: vi");
        client.println("User-Agent: ArduinoWiFi/1.1");
        client.println("Connection: close");
        client.println();
    }
    else {
        Serial.println("connection failed"); //error message if no client connect
        Serial.println();
    }

    while (client.connected() && !client.available()) delay(1); //waits for data
    while (client.connected() || client.available()) { //connected or data available
        char c = client.read(); //gets byte from ethernet buffer
        result = result + c;
    }
}
```

```
String location = root["name"];
String country = root["sys"]["country"];
float temperature = root["main"]["temp"];
float humidity = root["main"]["humidity"];
String weather = root["weather"]["main"];
String description = root["weather"]["description"];
float pressure = root["main"]["pressure"];
    float temperature_min = root["main"]["temp_min"];
    float temperature_max = root["main"]["temp_max"];
    String cloud=root["main"]["Cloudiness"];

weatherDescription = description;
weatherLocation = location;
Country = country;
Temperature = temperature;
Humidity = humidity;
Pressure = pressure;
Weather = weather;
temp_min = temperature_min;
temp_max = temperature_max;
cloudiness=cloud;
```

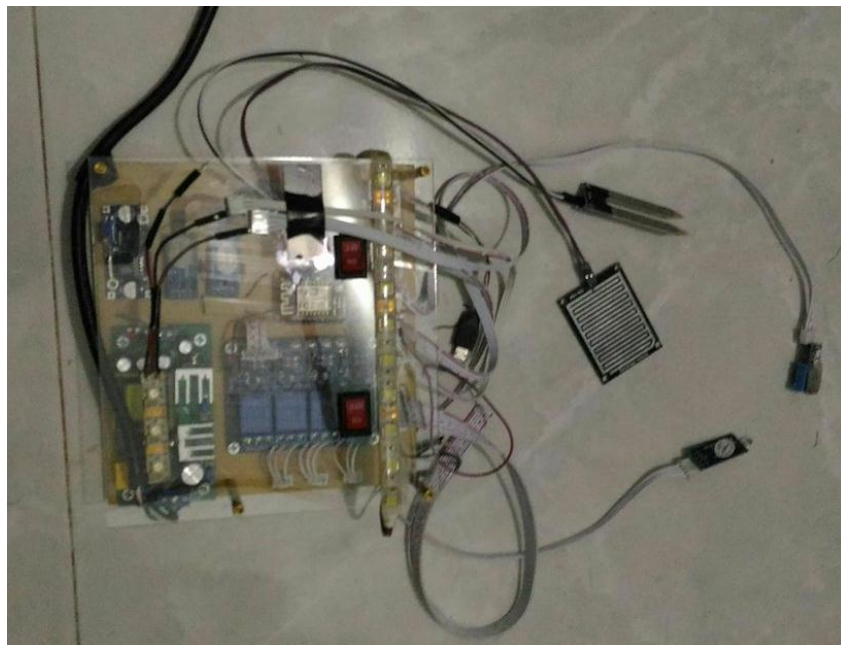
Sau khi xây dựng chương trình chúng ta thực hiện khai báo các chân dữ liệu được sử dụng trong hệ thống ở hàm setup(). Ở hàm này sẽ thực hiện khởi động các chân vật lý và thu nhận dữ liệu từ các chân đó.

```
pinMode(D1, OUTPUT); // đèn 1
pinMode(D4, OUTPUT); // đèn 2
pinMode(D6, OUTPUT); // đo độ ẩm kk
pinMode(D7, OUTPUT);
pinMode(D0, OUTPUT);
pixels.begin(); // khai báo đèn led
dht.begin();
n.begin();
pinMode(0, OUTPUT);
timer.setInterval(1000L, sendSensor);
timer.setInterval(1000L, hengio);
timer.setInterval(1000L, denleds);
```

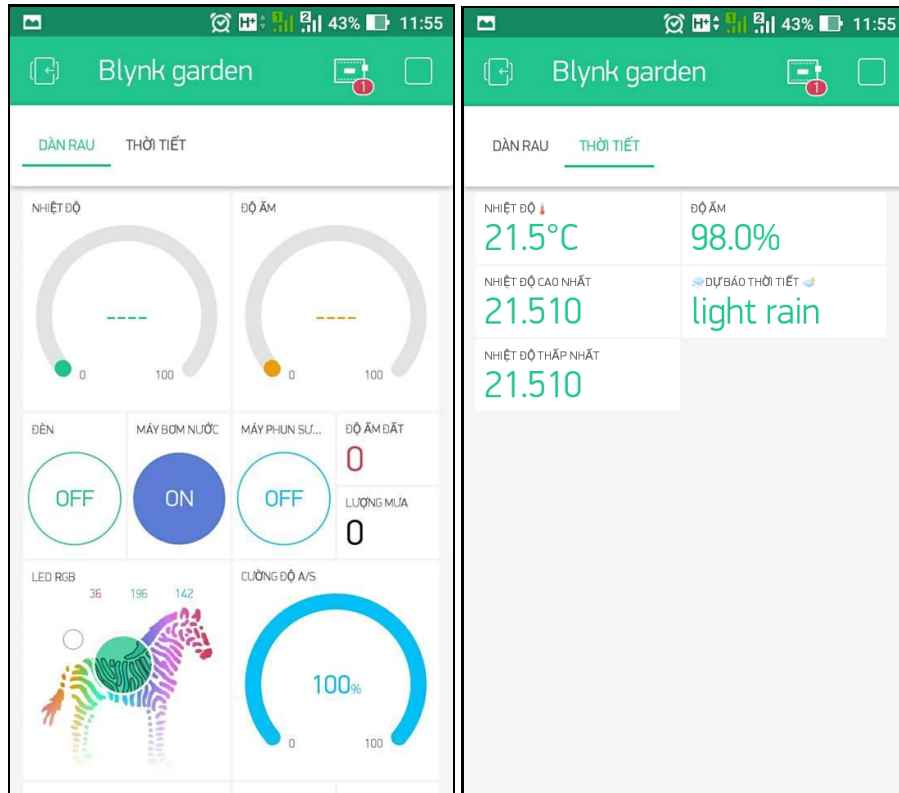
Như vậy là chúng ta đã thực hiện việc xây dựng hệ thống điều khiển dàn rau thủy canh.

3. Kết quả

Sau khi tiến hành thực hiện xây dựng hệ thống chúng ta xây dựng được một hệ thống điều khiển dàn rau thủy canh trên server Blynk. Sử dụng ứng dụng trên thiết bị di động để điều khiển hệ thống gồm bật tắt bóng đèn, van nước, phun sương, đo nhiệt độ, độ ẩm, cường độ ánh sáng để xây dựng hệ thống tự động.



Hệ thống điều khiển dàn thủy canh



Giao diện hệ thống điều khiển dàn rau thổ canh

Giao diện điều khiển thông qua ứng dụng Blynk trên thiết bị di động.