

Mình xin giới thiệu phiên bản cải tiến phần 2 của hệ thống nông nghiệp thông minh. Ở các phiên bản trước, mình đã giới thiệu với các bạn về hệ thống sử dụng [TIVA](#) / [Arduino](#) và ESP8266, tuy nhiên nó vẫn còn bộc lộ nhiều yếu điểm:

- Chưa có chức năng điều khiển qua điện thoại
- Vẫn phải xài kèm với TIVA/ Arduino, do đó lại tốn thêm 1 ít chi phí đầu tư

Do đó mình thử chuyển sang sử dụng với NodeMCU (ESP8266) để xem thử là với board này có đáp ứng đủ yêu cầu của mình hay không. Cũng nhân tiện cho các bạn thấy được ưu nhược điểm khi dùng board này

Bài viết này nằm trong serie về Nông nghiệp thông minh

- [Hệ thống nông nghiệp thông minh phiên bản Arduino](#)
- [Hệ thống nông nghiệp thông minh phiên bản ESP8266](#)
- [Hệ thống nông nghiệp thông minh phiên bản ARM TIVA](#)

Chuẩn bị

Phần cứng

- NodeMCU
- Cảm biến DHT22
- Cảm biến độ ẩm đất
- Modul 2 relay để điều khiển bơm
- Dây nối

Phần mềm

- Arduino IDE [có thể lập trình cho ESP8266](#)
- Thư viện [blynk](#) hoặc tải tại [đây](#)
- Thư viện [SimpleTimer](#)
- Thư viện [DHT](#)

Về cơ bản một số thông tin từ bài trước vẫn được giữ nguyên như các thông tin về nhiệt độ, độ ẩm, và độ ẩm đất, trừ thông tin về ánh sáng tạm thời loại do ESP8266 chỉ có một chân Analog, dựa trên các dữ liệu này thì sẽ thực hiện điều khiển tưới nước và cấp nhiệt cho cây, ở đây có một điểm khác nữa là modul 2 relay dùng để điều khiển 1 bơm nước và 1 đèn cấp nhiệt. LCD mình tạm thời bỏ không dùng ở trong ví dụ này.

Phân tích hệ thống

mình sẽ đi qua phân tích một chút về input và output của hệ thống

Input

- Các thông số của cảm biến: nhiệt độ, độ ẩm, độ ẩm đất

- Nút nhấn: điều khiển bơm và đèn

Output

- Relay điều khiển bơm
- Relay điều khiển đèn

Thông tin cần hiển thị lên điện thoại

- Bơm được bật
- Đèn được bật
- Hệ thống offline

Dữ liệu cần hiển thị và lưu trữ

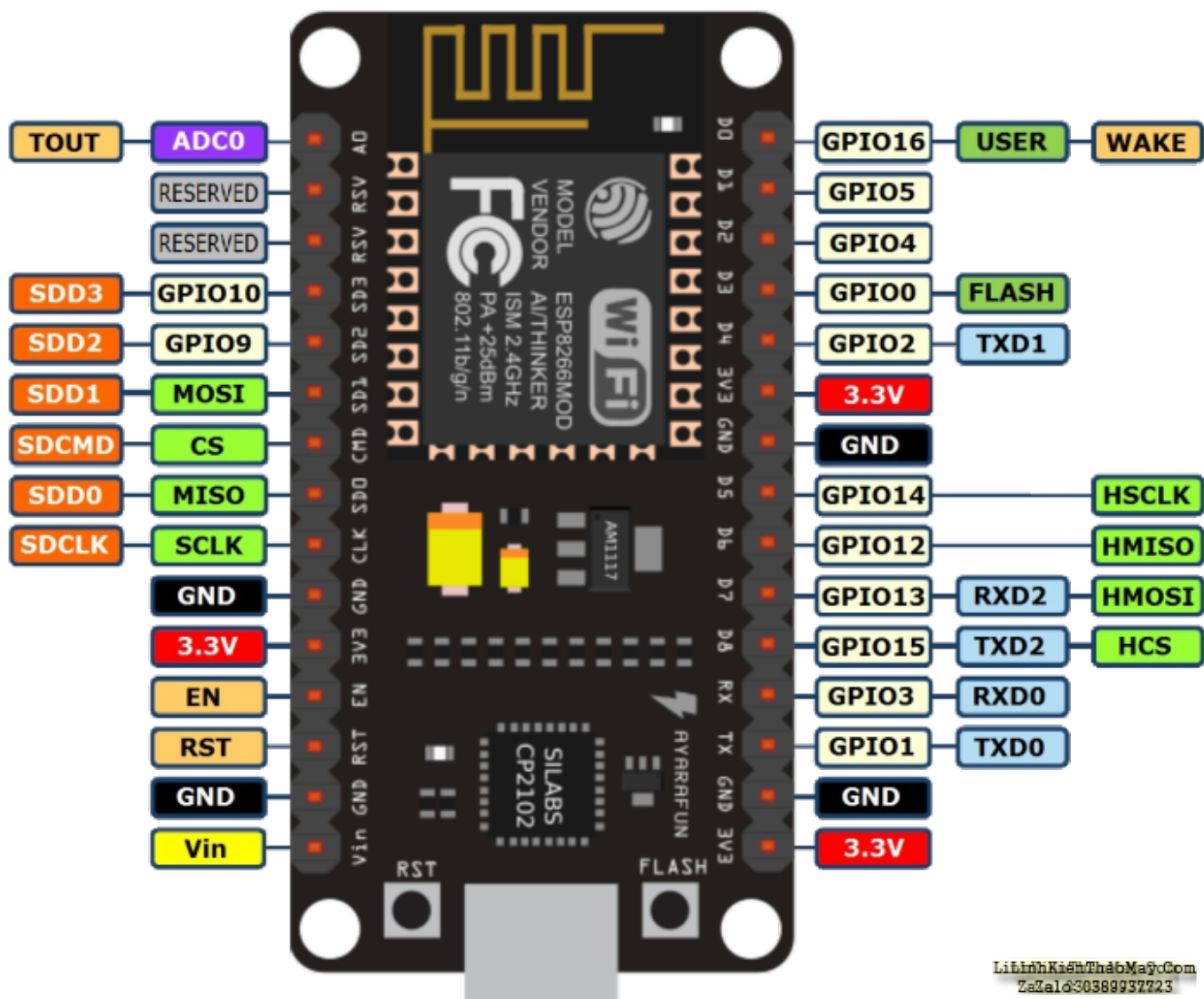
- Các thông số từ cảm biến

Thực hiện

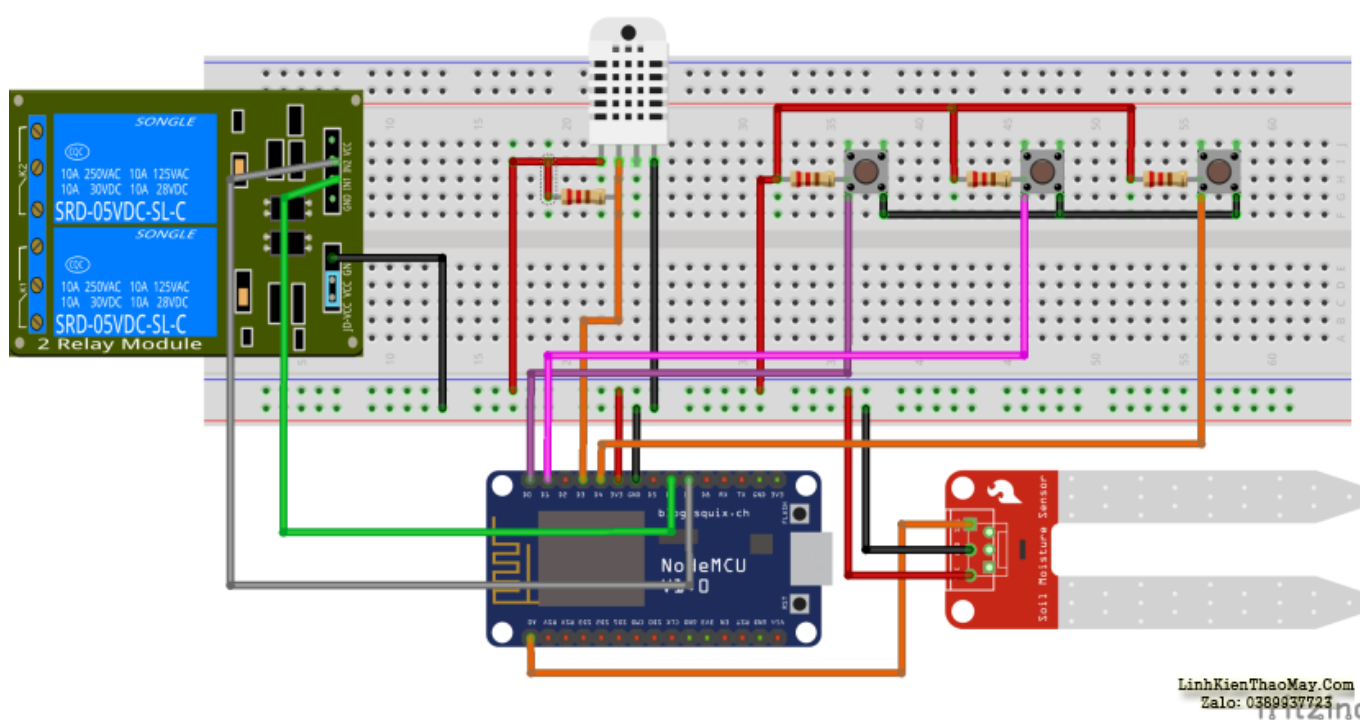
Trước khi thực hiện nếu bạn chưa có kiến thức gì về ESP8266 thì nên xem qua một số bài hướng dẫn về ESP8266 và các ví dụ ở [đây](#).

Do sử dụng phần cứng mới là ESP8266 nên mình trình bày lại một chút và kế thừa lại từ project [Tự làm hệ thống nông nghiệp phần 1](#) lại cho mọi người tiện theo dõi

Sơ đồ chân của NodeMCU



Sơ đồ kết nối hệ thống (nhớ **cấp nguồn cho Relay** nha)



Đọc tín hiệu từ DHT11/DHT22

Các bạn có thể kết nối DHT11/DHT22 theo sơ đồ như hình trên, có một lưu ý nhỏ là phải có kết nối trở 10k giữa chân Data out và chân VCC. Chân output sẽ được kết nối với chân D3 của nodeMCU, nguồn cấp cho DHT11/DHT22 là 3.3V được lấy từ NodeMCU.

Lưu ý:

Cần có sẵn thư viện DHT11/DHT22 để build không lỗi

Tùy loại cảm biến bạn sử dụng là DHT11 hay DHT22 mà comment out lại dòng

#define DHTTYPE DHT22 hoặc #define DHTTYPE DHT11

```
// Chương trình đọc nhiệt độ, độ ẩm từ cảm biến DHT
// Written by ladyada, public domain
// Chỉnh sửa cho ESP8266 bởi hocARM.org

#include "DHT.h"

#define DHTPIN D3          // Chân DATA nối với chân D3

// Uncomment loại cảm biến bạn sử dụng, nếu DHT11 thì uncomment DHT11
// và comment DHT22
// #define DHTTYPE DHT11    // DHT 11
#define DHTTYPE DHT22      // DHT 22 (AM2302), AM2321
// #define DHTTYPE DHT21    // DHT 21 (AM2301)

// Kết nối
// DHT      | ESP8266
// -----
// VCC(1)    | 3.3V
// DATA(2)   | D3
// NC(3)      | x
// GND(4)     | GND

// Kết nối chân 1 của DHT với 3.3V
// Nối trở 10k giữa chân 1 và chân 2

// Khởi tạo cảm biến
DHT dht(DHTPIN, DHTTYPE);

void setup() {
    // Khởi tạo cổng serial baud 115200
    Serial.begin(115200);
    Serial.println("DHTxx test!");
```

```
// Bật đầu đọc dữ liệu
dht.begin();
}

void loop() {
    // Đợi chuyển đổi dữ liệu khoảng 2s
    delay(2000);

    float h = dht.readHumidity();
    // Đọc giá trị nhiệt độ C (mặc định)
    float t = dht.readTemperature();
    // Đọc giá trị nhiệt độ F (isFahrenheit = true)
    float f = dht.readTemperature(true);

    // Kiểm tra quá trình đọc thành công hay không
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }

    // IN thông tin ra màn hình
    Serial.print("Do am: ");
    Serial.print(h);
    Serial.print(" %t");
    Serial.print("Nhiệt độ: ");
    Serial.print(t);
    Serial.println(" *C ");
}
}
```

Đọc tín hiệu từ cảm biến độ ẩm đất

Cảm biến độ ẩm đất sẽ được kết nối với chân A0 của NodeMCU, nguồn cấp là 3v3.

Chương trình kết hợp đọc DHT22 và cảm biến độ ẩm đất

```
// Chương trình đọc nhiệt độ, độ ẩm từ cảm biến DHT
// Thêm chức năng đọc cảm biến ánh sáng
// Thêm chức năng đọc cảm biến độ ẩm đất
// Chọn sơ đồ cho Arduino Uno tại hocARM.org
// Kết nối
// DHT | ESP8266
//-----
// VCC(1) | 3.3V
// DATA(2) | D3
// NC(3) | x
// GND(4) | GND
```

```
// Nối trỏ 10k giữa chân 1 và chân 2
//-----
//Cắm biến độ ẩm | ESP8266
//-----
// VCC(1)          | 3.3V
// GND(2)          | GND
// D0(3)           | x
// A0(4)           | A0

#include "DHT.h"

#define DHTPIN D3      // Chân DATA nối với 2
// #define LDR_PIN A0   // Chân A0 nối với chân OUT cảm biến as
#define SOIL_MOIST_1_PIN A0 // Chân A1 nối với cảm biến độ ẩm

// Uncomment loại cảm biến bạn sử dụng, nếu DHT11 thì uncomment DHT11
// và comment DHT22
// #define DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
// #define DHTTYPE DHT21 // DHT 21 (AM2301)
int humDHT;
int tempDHT;
//int lumen;
int soilMoist;
// Khởi tạo cảm biến
DHT dht(DHTPIN, DHTTYPE);

void setup() {
    // Khởi tạo cổng serial baud 115200
    Serial.begin(115200);
    Serial.println("DHTxx test!");
    // Bật đầu đọc dữ liệu
    dht.begin();
}

void loop() {
    readSensors();
    // IN thông tin ra màn hình
    Serial.print("Do am: ");
    Serial.print(humDHT);
    Serial.print(" %t");
    Serial.print("Nhiệt độ: ");
    Serial.print(tempDHT);
    Serial.print(" *Ct");
    // Serial.print("Ánh sáng: ");
    // Serial.print(lumen);
```

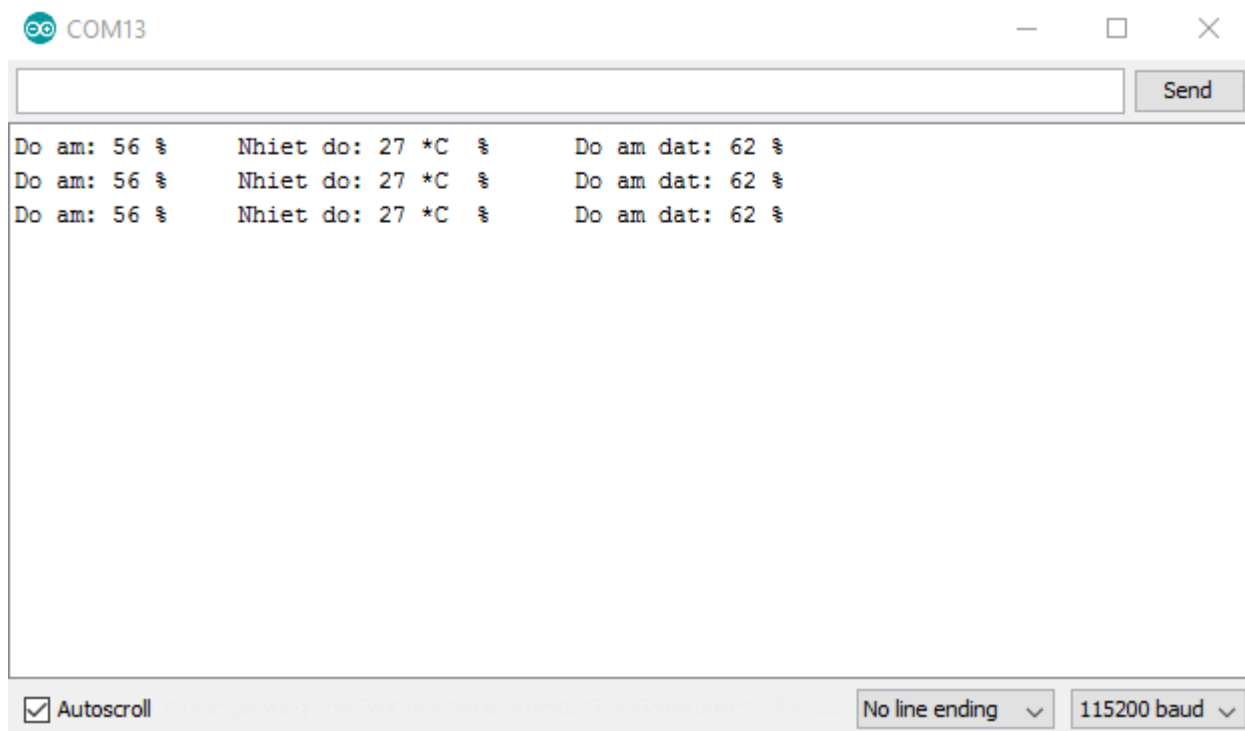
```
Serial.print(" %t");
Serial.print("Do am dat: ");
Serial.print(soilMoist);
Serial.println(" %");
}

int getSoilMoist()
{
    int i = 0;
    int anaValue = 0;
    for (i = 0; i < 10; i++) //
    {
        anaValue += analogRead(SOIL_MOIST_1_PIN); //Đọc giá trị cảm biến
        độ ẩm đất
        delay(50); // Đợi đọc giá trị ADC
    }
    anaValue = anaValue / (i);
    anaValue = map(anaValue, 1023, 0, 0, 100); //Ít nước:0% ==> Nhiều
    nước 100%
    return anaValue;
}

void readSensors(void)
{
    tempDHT = dht.readTemperature(); //Đọc nhiệt độ DHT22
    humDHT = dht.readHumidity(); //Đọc độ ẩm DHT22
    soilMoist = getSoilMoist(); //Đọc cảm biến độ ẩm đất
}
```

Kết quả

Ta sẽ có kết quả của việc đọc DHT và cảm biến độ ẩm đất



Thêm nút nhấn, relay

Ở đây mình thêm 3 nút nhấn

- Nút 1 (D0) điều khiển bơm (D6)
- Nút 2 (D1) điều khiển đèn (D7)
- Nút 3 (D4) dùng để đọc cảm biến và cập nhật trạng thái

/* Chương trình đọc nhiệt độ, độ ẩm từ cảm biến DHT

Thêm chức năng đọc cảm biến ánh sáng

Thêm chức năng đọc cảm biến độ ẩm đất

Thêm hiển thị LCD

Thêm chức năng điều khiển tưới tiêu/ cấp nhiệt bằng tay bơm và đèn

HocARM NDTR BOT for ESP8266 by hocARM.org

// Chương trình đọc nhiệt độ, độ ẩm từ cảm biến DHT

// Thêm chức năng đọc cảm biến ánh sáng

// Thêm chức năng đọc cảm biến độ ẩm đất

// Kết nối

// DHT | ESP8266

//-----

// VCC(1) | 3.3V

// DATA(2) | D3

// NC(3) | x

// GND(4) | GND

// Nối trở 10k giữa chân 1 và chân 2

//-----

//Cảm biến độ ẩm | ESP8266

//-----


```
// VCC(1)      | 3.3V
// GND(2)      | GND
// D0(3)       | x
// A0(4)       | A0
*/
#include "DHT.h"

#define DHTPIN D3      // Chân DATA nối với D3
// #define LDR_PIN A0    // Chân PE3 nối với chân OUT cảm biến as
#define SOIL_MOIST_1_PIN A0 // Chân PE4 nối với cảm biến độ ẩm
// Relay, nút nhấn
#define PUMP_ON_BUTTON D0    // Nút có sọc trên kit
#define LAMP_ON_BUTTON D1    // Nút có sọc trên kit
#define SENSORS_READ_BUTTON D4

#define PUMP_PIN D6    // Bơm
#define LAMP_PIN D7    // Đèn
// Uncomment loại cảm biến bạn sử dụng, nếu DHT11 thì uncomment DHT11
// và comment DHT22
// #define DHTTYPE DHT11    // DHT 11
#define DHTTYPE DHT22    // DHT 22 (AM2302), AM2321
// #define DHTTYPE DHT21    // DHT 21 (AM2301)

// Biến lưu các giá trị cảm biến
int humDHT;
int tempDHT;
int lumen;
int soilMoist;
// Biến lưu trạng thái bơm
boolean pumpStatus = 0;
boolean lampStatus = 0;

int timePumpOn = 10; // Thời gian bật bơm nước
// Biến cho timer
long sampleTimingSeconds = 50; // ==> Thời gian đọc cảm biến (s)
long startTiming = 0;
long elapsedTime = 0;
// Khởi tạo cảm biến
DHT dht(DHTPIN, DHTTYPE);

void setup() {
    pinMode(PUMP_PIN, OUTPUT);
    pinMode(LAMP_PIN, OUTPUT);
    pinMode(PUMP_ON_BUTTON, INPUT_PULLUP);
    pinMode(LAMP_ON_BUTTON, INPUT_PULLUP);
    pinMode(SENSORS_READ_BUTTON, INPUT_PULLUP);
}
```

```
aplyCmd();
// Khởi tạo cổng serial baud 115200
Serial.begin(115200);
Serial.println("HocARM NDTR Bot!");
// Bắt đầu đọc dữ liệu
dht.begin();
readSensors(); // Khởi tạo đọc cảm biến
startTiming = millis(); // Bắt đầu đếm thời gian
}

void loop() {
    // Khởi tạo timer
    elapsedTime = millis() - startTiming;
    readLocalCmd();
    if (elapsedTime > (sampleTimingSeconds * 1000))
    {
        readSensors();
        printData();
        startTiming = millis();
    }
}

int getSoilMoist()
{
    int i = 0;
    int anaValue = 0;
    for (i = 0; i < 10; i++) //
    {
        anaValue += analogRead(SOIL_MOIST_1_PIN); //Đọc giá trị cảm biến
        độ ẩm đất
        delay(50); // Đợi đọc giá trị ADC
    }
    anaValue = anaValue / (i);
    anaValue = map(anaValue, 1023, 0, 0, 100); //Ít nước:0% ==> Nhiều
    nước 100%
    return anaValue;
}

void readSensors(void)
{
    tempDHT = dht.readTemperature(); //Đọc nhiệt độ DHT22
    humDHT = dht.readHumidity(); //Đọc độ ẩm DHT22
    // lumen = getLumen(LDR_PIN); //Đọc ánh sáng
    soilMoist = getSoilMoist(); //Đọc cảm biến độ ẩm đất
}

void printData(void)
{

```

```
// IN thông tin ra màn hình
Serial.print("Do am: ");
Serial.print(humDHT);
Serial.print(" %t");
Serial.print("Nhiệt độ: ");
Serial.print(tempDHT);
Serial.print(" *Ct");
// Serial.print("Ánh sáng: ");
// Serial.print(lumen);
Serial.print(" %t");
Serial.print("Độ ẩm đất: ");
Serial.print(soilMoist);
Serial.println(" %");
}

/*****
Hàm đọc trạng thái bơm và kiểm tra nút nhấn
(Nút nhấn mặc định là mức "CAO"):
*****/
void readLocalCmd()
{
    boolean digiValue = debounce(PUMP_ON_BUTTON);
    if (!digiValue)
    {
        pumpStatus = !pumpStatus;
        applyCmd();
    }

    digiValue = debounce(LAMP_ON_BUTTON);
    if (!digiValue)
    {
        lampStatus = !lampStatus;
        applyCmd();
    }

    digiValue = debounce(SENSORS_READ_BUTTON);
    if (!digiValue)
    {
        readSensors();
        printData();
    }
}

/*****
Thực hiện điểu khiển các bơm
*****/
void applyCmd()
```

```
{
  if (pumpStatus == 1) digitalWrite(PUMP_PIN, LOW);
  if (pumpStatus == 0) digitalWrite(PUMP_PIN, HIGH);

  if (lampStatus == 1) digitalWrite(LAMP_PIN, LOW);
  if (lampStatus == 0) digitalWrite(LAMP_PIN, HIGH);
}
/*****
Hàm kiểm tra trạng thái phím bấm
*****/
boolean debounce(int pin)
{
  boolean state;
  boolean previousState;
  const int debounceDelay = 60;

  previousState = digitalRead(pin);
  for (int counter = 0; counter < debounceDelay; counter++)
  {
    delay(1);
    state = digitalRead(pin);
    if (state != previousState)
    {
      counter = 0;
      previousState = state;
    }
  }
  return state;
}
```

Đến đây coi như đã hoàn thành được các bước cơ bản đọc dữ liệu và điều khiển offline như bài trước:

- Đọc cảm biến
- Đọc nút nhấn
- Điều khiển bơm và đèn
- Hiển thị thông tin lên máy tính qua terminal (các bạn có thể thay thế bằng hiển thị lên LCD)

Vì thời gian cho mỗi nhiệm vụ là khác nhau nên mình sẽ dùng timer để điều khiển thời gian cho từng task (nhiệm vụ) thay cho cách dùng `millis()`.

Với thư viện SimpleTimer thì mình sẽ thực hiện các nhiệm vụ theo thời gian như sau:

```
timer.setInterval(1000L, readLocalCmd);          // 1s đọc trạng thái nút
nhấn 1
timer.setInterval(2000L, getDhtData);             // 2s đọc cảm biến DHT 1
```

Tài liệu này được tải từ website: <http://linhkienthaomay.com>. Zalo hỗ trợ: 0389937723

```
l\n
timer.setInterval(10000, getSoilMoist); // 10s đọc c[m bi[n độ [m đ[  
1 l\n
timer.setInterval(10000, printData);          // 10s hi[n thị thông tin  
debug lên máy tính 1 l\n
```

Ở đây mình xin nói thêm về nút nhấn đọc dữ liệu (nút nhấn thứ 3), trong quá trình hệ thống làm việc, như trên thì 10s sẽ có thông tin thông báo cho mình biết, nếu tăng lên 10ph, 30ph thì lại phải ngồi chờ mỗi mòn mới biết được thông tin ? Quá bất tiện, do đó mà nút nhấn đọc dữ liệu này ra đời, mục đích là khi ấn nút thì ta có được thông tin ngay tức thời thôi.

```
/* Chương trình đọc nhiệt độ, độ [m từ c[m bi[n DHT  
  Thêm chức năng đọc c[m bi[n ánh sáng  
  Thêm chức năng đọc c[m bi[n độ [m đ[  
  Thêm hi[n thị LCD  
  Thêm chức năng đi[u khi[n tưới tiêu với bơm và đi[u khi[n nhiệt với  
đèn
```

```
HocARM NDTR BOT for ESP8266 by hocARM.org
```

```
-----  
// Chương trình đọc nhiệt độ, độ [m từ c[m bi[n DHT  
// Thêm chức năng đọc c[m bi[n ánh sáng  
// Thêm chức năng đọc c[m bi[n độ [m đ[  
// Ch[nh s[ cho Arduino Uno b[i hocARM.org  
// K[  
// DHT      | ESP8266  
//-----  
// VCC(1)    | 3.3V  
// DATA(2)  | D3  
// NC(3)     | x  
// GND(4)    | GND  
// N[i tr[ 10k giữa chân 1 và chân 2  
//-----  
//C[m bi[n độ [m | ESP8266  
//-----  
// VCC(1)    | 3.3V  
// GND(2)    | GND  
// D0(3)     | x  
*/  
#include "DHT.h"  
  
#define DHTPIN D3      // Chân DATA n[i với D3  
  
#define SOIL_MOIST_1_PIN A0 // Chân PE4 n[i với c[m bi[n độ [m  
// Relay, nút nh[n  
#define PUMP_ON_BUTTON D0  
#define LAMP_ON_BUTTON D1
```

```
#define SENSORS_READ_BUTTON D4

#define PUMP_PIN D6    //Bơm
#define LAMP_PIN D7    //Đèn
// Uncomment loại cảm biến bạn sử dụng, nếu DHT11 thì uncomment DHT11
// và comment DHT22
// #define DHTTYPE DHT11    // DHT 11
#define DHTTYPE DHT22    // DHT 22 (AM2302), AM2321
// #define DHTTYPE DHT21    // DHT 21 (AM2301)

// Biến lưu các giá trị cảm biến
int humDHT;
int tempDHT;
int lumen;
int soilMoist;
// Biến lưu trạng thái bơm
boolean pumpStatus = 0;
boolean lampStatus = 0;

int timePumpOn = 10; // Thời gian bật bơm nước
// Biến cho timer
long sampleTimingSeconds = 50; // ==> Thời gian đọc cảm biến (s)
long startTiming = 0;
long elapsedTime = 0;
// Khởi tạo cảm biến
DHT dht(DHTPIN, DHTTYPE);

void setup() {
    pinMode(PUMP_PIN, OUTPUT);
    pinMode(LAMP_PIN, OUTPUT);
    pinMode(PUMP_ON_BUTTON, INPUT_PULLUP);
    pinMode(LAMP_ON_BUTTON, INPUT_PULLUP);
    pinMode(SENSORS_READ_BUTTON, INPUT_PULLUP);
    aplyCmd();
    // Khởi tạo cổng serial baud 115200
    Serial.begin(115200);
    Serial.println("HocARM NDTR Bot!");
    // Bật đầu đọc dữ liệu
    dht.begin();
    readSensors(); // Khởi tạo đọc cảm biến
    startTiming = millis(); // Bật đầu đếm thời gian
}

void loop() {
    // Khởi tạo timer
    elapsedTime = millis() - startTiming;
```

```
readLocalCmd();
if (elapsedTime > (sampleTimingSeconds * 1000))
{
    readSensors();
    printData();
    startTiming = millis();
}
}
int getSoilMoist()
{
    int i = 0;
    int anaValue = 0;
    for (i = 0; i < 10; i++) //
    {
        anaValue += analogRead(SOIL_MOIST_1_PIN); //Đọc giá trị cảm biến
        độ ẩm đất
        delay(50); // Đợi đọc giá trị ADC
    }
    anaValue = anaValue / (i);
    anaValue = map(anaValue, 1023, 0, 0, 100); //Ít nước:0% ==> Nhiều
    nước 100%
    return anaValue;
}
void readSensors(void)
{
    tempDHT = dht.readTemperature(); //Đọc nhiệt độ DHT22
    humDHT = dht.readHumidity(); //Đọc độ ẩm DHT22
    soilMoist = getSoilMoist(); //Đọc cảm biến độ ẩm đất
}
void printData(void)
{
    // IN thông tin ra màn hình
    Serial.print("Do am: ");
    Serial.print(humDHT);
    Serial.print(" %t");
    Serial.print("Nhiệt độ: ");
    Serial.print(tempDHT);
    Serial.print(" *Ct");
    Serial.print(" %t");
    Serial.print("Do ẩm đất: ");
    Serial.print(soilMoist);
    Serial.println(" %");
}
```

/*****

Hàm đọc trạng thái bơm và kiểm tra nút nhàn

(Nút nhấn mặc định là mức "CAO"):

*****/

```
void readLocalCmd()
```

```
{
    boolean digiValue = debounce(PUMP_ON_BUTTON);
    if (!digiValue)
    {
        pumpStatus = !pumpStatus;
        aplyCmd();
    }

```

```
    digiValue = debounce(LAMP_ON_BUTTON);
    if (!digiValue)
    {
        lampStatus = !lampStatus;
        aplyCmd();
    }

```

```
    digiValue = debounce(SENSORS_READ_BUTTON);
    if (!digiValue)
    {
        readSensors();
        printData();
    }

```

```
}
```

/*****

Thực hiện điểu khiển các bơm

*****/

```
void aplyCmd()
```

```
{
    if (pumpStatus == 1) digitalWrite(PUMP_PIN, LOW);
    if (pumpStatus == 0) digitalWrite(PUMP_PIN, HIGH);
    if (lampStatus == 1) digitalWrite(LAMP_PIN, LOW);
    if (lampStatus == 0) digitalWrite(LAMP_PIN, HIGH);
}

```

/*****

Hàm kiểm tra trạng thái phím bấm

*****/

```
boolean debounce(int pin)
```

```
{
    boolean state;
    boolean previousState;
    const int debounceDelay = 60;

    previousState = digitalRead(pin);
    for (int counter = 0; counter < debounceDelay; counter++)

```



```
{
    delay(1);
    state = digitalRead(pin);
    if (state != previousState)
    {
        counter = 0;
        previousState = state;
    }
}
return state;
}
```

Điều khiển tự động

Hệ thống coi như đã hoạt động được ở chế độ offline, giờ mình cho nó thông minh hơn một chút là có thể tự bơm nước hoặc tự bật đèn cấp nhiệt được dựa trên các tín hiệu từ cảm biến

Cảm biến độ ẩm đất:

- Ướt: trên 88%, không cần phải bơm thêm nước
- Bình thường: từ 66% tới 88%, là mức hoạt động bình thường, ổn định
- Khô: dưới 66%, cần phải bơm nước

Nhiệt độ môi trường

- Lạnh: Dưới 15 độ C, cần bật đèn sưởi
- Bình thường: từ 15 - 24 độ C
- Nóng: trên 24 độ C, không được bật đèn sưởi.

mình sẽ định nghĩa các thông số này trong chương trình

```
/* Thông số điều khiển tự động */
#define DRY_SOIL      66
#define WET_SOIL      88
#define COLD_TEMP     15
#define HOT_TEMP      24
#define TIME_PUMP_ON  15
#define TIME_LAMP_ON  15
```

Với TIME_PUMP_ON và TIME_LAMP_ON là thời gian bật bơm và bật đèn, với một số phân tích như trên thì khi áp dụng vào code sẽ rất đơn giản

- Nếu DRY => PUMP = ON
- Nếu COLD => LAMP = ON

mình sẽ có hàm điều khiển tự động là `autoControlPlantation()`, nhiệm vụ chính là điều khiển tự động bơm nước và đèn sưởi

Chương trình mình viết lại hàm đọc cảm biến DHT, cảm biến độ ẩm đất, thêm thư viện SimpleTimer và hàm điều khiển tự động

```
/* Chương trình đọc nhiệt độ, độ ẩm từ cảm biến DHT
Thêm chức năng đọc cảm biến ánh sáng
Thêm chức năng đọc cảm biến độ ẩm đất
Thêm hiển thị LCD
Thêm chức năng điều khiển tưới tiêu bằng tay 2 bơm
HocARM NDTR BOT for ESP8266 by hocARM.org
-----
// Chương trình đọc nhiệt độ, độ ẩm từ cảm biến DHT
// Thêm chức năng đọc cảm biến ánh sáng
// Thêm chức năng đọc cảm biến độ ẩm đất
// Chọn sạ cho Arduino Uno bởi hocARM.org
// Kết nối
// DHT      | ESP8266
//-----
// VCC(1)    | 3.3V
// DATA(2)   | 2
// NC(3)      | x
// GND(4)     | GND
// Nối trở 10k giữa chân 1 và chân 2
//-----
//Cảm biến độ ẩm | ESP8266
//-----
// VCC(1)      | 3.3V
// GND(2)      | GND
// D0(3)       | x
// A0(4)       | A1
*/
#include "DHT.h"
/* TIMER */
#include <SimpleTimer.h>

#define DHTPIN D3      // Chân DATA nối với D3
#define SOIL_MOIST_1_PIN A0 // Chân PE4 nối với cảm biến độ ẩm
// Relay, nút nhấn
#define PUMP_ON_BUTTON D0
#define LAMP_ON_BUTTON D1
#define SENSORS_READ_BUTTON D4

#define PUMP_PIN D6     //Bơm
#define LAMP_PIN D7     //Đèn
// Uncomment loại cảm biến bạn sử dụng, nếu DHT11 thì uncomment DHT11
// và comment DHT22
// #define DHTTYPE DHT11 // DHT 11
```

```
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
// #define DHTTYPE DHT21 // DHT 21 (AM2301)

/* Thông số cho chế độ tự động */
#define DRY_SOIL 66
#define WET_SOIL 88
#define COLD_TEMP 15
#define HOT_TEMP 24
#define TIME_PUMP_ON 15
#define TIME_LAMP_ON 15

/* TIMER */
#define READ_BUTTONS_TM 1L // Tương ứng với giây
// #define READ_SOIL_TEMP_TM 2L
#define READ_SOIL_HUM_TM 10L // Đọc cảm biến ẩm đất
#define READ_AIR_DATA_TM 2L // Đọc DHT
#define DISPLAY_DATA_TM 10L
#define AUTO_CTRL_TM 60L // Chế độ tự động

// Biến lưu các giá trị cảm biến
float humDHT = 0;
float tempDHT = 0;
// int lumen;
int soilMoist = 0;
// Biến lưu trạng thái bơm
boolean pumpStatus = 0;
boolean lampStatus = 0;

int timePumpOn = 10; // Thời gian bật bơm nước
// Biến cho timer
long sampleTimingSeconds = 50; // ==> Thời gian đọc cảm biến (s)
long startTiming = 0;
long elapsedTime = 0;
// Khởi tạo timer
SimpleTimer timer;

// Khởi tạo cảm biến
DHT dht(DHTPIN, DHTTYPE);

void setup() {
    pinMode(PUMP_PIN, OUTPUT);
    pinMode(LAMP_PIN, OUTPUT);
    pinMode(PUMP_ON_BUTTON, INPUT_PULLUP);
    pinMode(LAMP_ON_BUTTON, INPUT_PULLUP);
    pinMode(SENSORS_READ_BUTTON, INPUT_PULLUP);
    applyCmd();
```

```
// Khai tạo cổng serial baud 115200
Serial.begin(115200);
Serial.println("HocARM NDTR Bot!");
// Bật đầu đọc dữ liệu
dht.begin();
startTimers();
}

void loop() {
    timer.run(); // Chạy SimpleTimer
}
void getSoilMoist(void)
{
    int i = 0;
    soilMoist = 0;
    for (i = 0; i < 10; i++) //
    {
        soilMoist += analogRead(SOIL_MOIST_1_PIN); //Đọc giá trị cảm biến
        độ ẩm đất
        delay(50); // Đợi đọc giá trị ADC
    }
    soilMoist = soilMoist / (i);
    soilMoist = map(soilMoist, 1023, 0, 0, 100); //Ít nước:0% ==> Nhiều
    nước 100%
}

void getDhtData(void)
{
    tempDHT = dht.readTemperature();
    humDHT = dht.readHumidity();
    if (isnan(humDHT) || isnan(tempDHT)) // Kiểm tra kết nối lỗi thì
    thông báo.
    {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }
}

void printData(void)
{
    // IN thông tin ra màn hình
    Serial.print("Do am: ");
    Serial.print(humDHT);
    Serial.print(" %t");
    Serial.print("Nhiệt độ: ");
    Serial.print(tempDHT);
```

```
Serial.print(" *Ct");
Serial.print(" %t");
Serial.print("Do am dat: ");
Serial.print(soilMoist);
Serial.println(" %");
}

/*****
Hàm đọc trạng thái bơm và kiểm tra nút nhàn
(Nút nhàn mặc định là mức "CA0"):
*****/
void readLocalCmd()
{
    boolean digiValue = debounce(PUMP_ON_BUTTON);
    if (!digiValue)
    {
        pumpStatus = !pumpStatus;
        aplyCmd();
    }

    digiValue = debounce(LAMP_ON_BUTTON);
    if (!digiValue)
    {
        lampStatus = !lampStatus;
        aplyCmd();
    }

    digiValue = debounce(SENSORS_READ_BUTTON);
    if (!digiValue)
    {
        getDhtData();
        getSoilMoist();
        printData();
    }
}

/*****
Thực hiện điều khiển các bơm
*****/
void aplyCmd()
{
    if (pumpStatus == 1) digitalWrite(PUMP_PIN, LOW);
    if (pumpStatus == 0) digitalWrite(PUMP_PIN, HIGH);

    if (lampStatus == 1) digitalWrite(LAMP_PIN, LOW);
    if (lampStatus == 0) digitalWrite(LAMP_PIN, HIGH);
}
```

```
/******  
Hàm kiểm tra trạng thái phím bấm  
*****/  
boolean debounce(int pin)  
{  
    boolean state;  
    boolean previousState;  
    const int debounceDelay = 60;  
  
    previousState = digitalRead(pin);  
    for (int counter = 0; counter < debounceDelay; counter++)  
    {  
        delay(1);  
        state = digitalRead(pin);  
        if (state != previousState)  
        {  
            counter = 0;  
            previousState = state;  
        }  
    }  
    return state;  
}  
/******  
* Chế độ tự động dựa trên thông số cảm biến  
*****/  
void autoControlPlantation(void)  
{  
    if (soilMoist < DRY_SOIL)  
    {  
        turnPumpOn();  
    }  
  
    if (tempDHT < COLD_TEMP)  
    {  
        turnLampOn();  
    }  
}  
/******  
* Bật bơm trong thời gian định sẵn  
*****/  
void turnPumpOn()  
{  
    pumpStatus = 1;  
    applyCmd();  
    delay (TIME_PUMP_ON*1000);  
    pumpStatus = 0;
```

```
    aplyCmd();
}
/*****
*  Bật đèn trong thời gian định s□n
*****/
void turnLampOn()
{
    lampStatus = 1;
    aplyCmd();
    delay (TIME_LAMP_ON*1000);
    lampStatus = 0;
    aplyCmd();
}

/*****
    Kh□i động Timers
*****/
void startTimers(void)
{
    timer.setInterval(READ_BUTTONS_TM * 1000, readLocalCmd);
    timer.setInterval(READ_AIR_DATA_TM * 1000, getDhtData);
    timer.setInterval(READ_SOIL_HUM_TM * 1000, getSoilMoist);
    timer.setInterval(AUTO_CTRL_TM * 1000, autoControlPlantation);
    timer.setInterval(DISPLAY_DATA_TM*1000, printData);
}
```

Điều khiển qua điện thoại với Blynk

Đây là phần được mong đợi nhất, mình xin trình bày phương án đơn giản nhất để có thể điều khiển từ xa thông qua điện thoại.

Trước tiên sẽ là tạo giao diện thông qua app Blynk trên smartphone có các chức năng sau

- Đọc và hiển thị toàn bộ dữ liệu từ cảm biến và trạng thái của bơm, đèn
- Điều khiển từ xa Đèn và Bơm
- Gửi thông báo hệ thống offline hoặc khi đèn hoặc bơm được bật
- Lưu lại dữ liệu từ cảm biến

Mình sẽ tạo ra 2 tab để hiển thị thông tin trên

- Cảm biến
- Nhiệt độ, input là V10 value 0 - 50, frequency: 5s
- Độ ẩm, input là V11 value 0 - 100, frequency: 5s
- Độ ẩm đất input là V12 value 0 - 100, frequency: 5s
- LED cho bơm Red, V0
- LED cho đèn Green, V1
- Đồ thị hiển thị thông tin V10 - V12

- Điều khiển
- Nút PUMP, output: V3 0 - 1, mode:push, label: on - ACT, off - OK
- Nút LAMP, output: V4 0 - 1, mode:push, label: on - ACT, off - OK
- LED PUMP, V0
- LED LAMP, V1
- Thông báo Notification: when HW goes offline: ON
- Biểu đồ hiển thị
- Graph hiển thị thông tin cảm biến từ V10-V12

Để chạy được app Blynk thì cần có một số cái mới cần phải thêm vào chương trình

- Thêm thư viện BlynkSimpleEsp8266 ở đầu chương trình
- Trong `Setup()`, cần cấu hình Blynk với token, tên wifi và mật khẩu với hàm `Blynk.begin(auth, ssid, pass);`
- Xác định thời gian gửi dữ liệu lên server Blynk : `timer.setInterval(5000L, sendUptime);`
- Gọi hàm `Blynk.run();` tại vòng lặp `loop()`
- Tạo hàm `sendUptime();` để gửi dữ liệu từ cảm biến lên Blynk Server: `Blynk.virtualWrite(VirtualPin, sensor data);`

Bạn cần thay đổi một số thông số tương ứng với thông số của bạn như sau

```
//Token Blynk và wifi
char auth[] = "api_token_blynk"; // Điền api token blynk của bạn
char ssid[] = "ten_wifi"; //Tên wifi
char pass[] = "pass_wifi";//Password
```

Ngoài ra cần có “LED báo trạng thái” trên Blynk để lưu trạng thái được định nghĩa

```
WidgetLED PUMP(V0); // Nhận tín hiệu từ nút V0 của Blynk App
WidgetLED LAMP(V1); // Nhận tín hiệu từ nút V1 của Blynk App
```

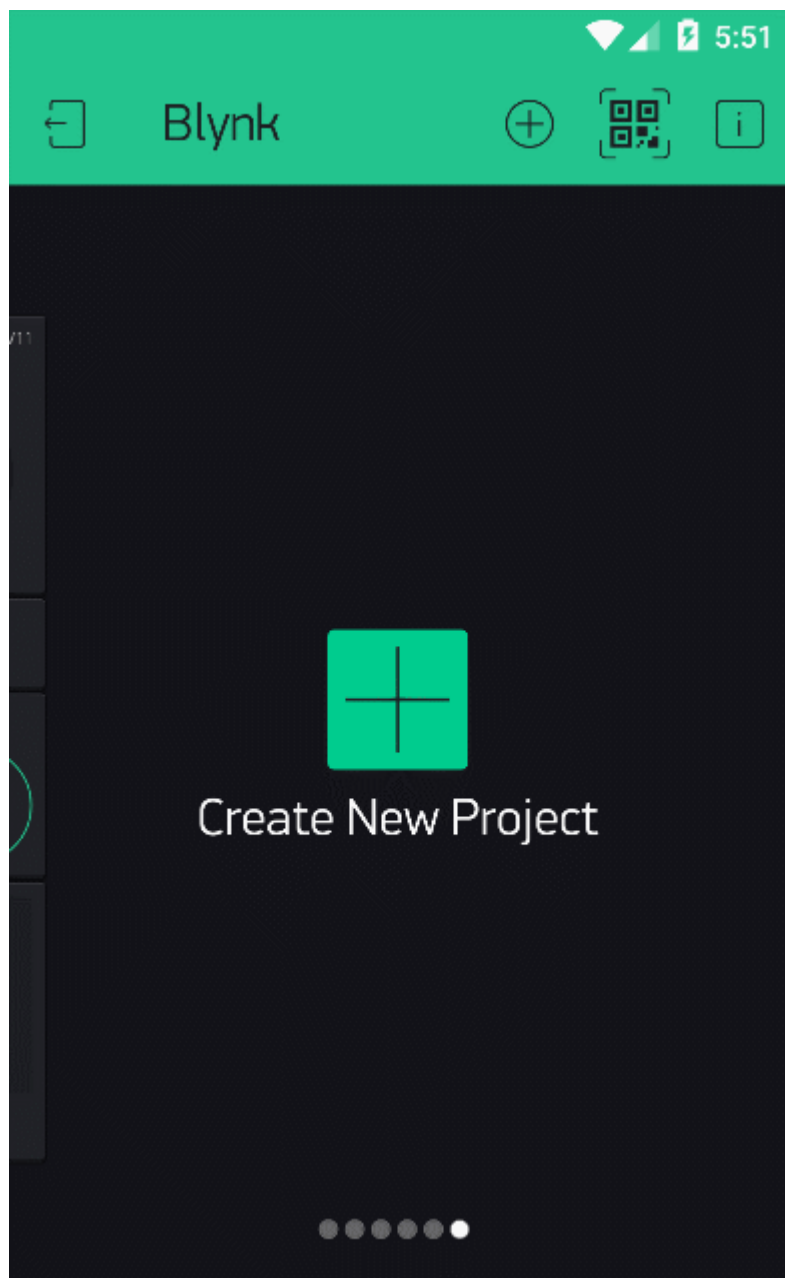
Để bật và tắt bơm và đèn kết nối với PIN ảo V0 và V1 tương ứng với các hàm

- `PUMP.on();` và `PUMP.off();`
- `LAMP.on();` và `LAMP.off();`

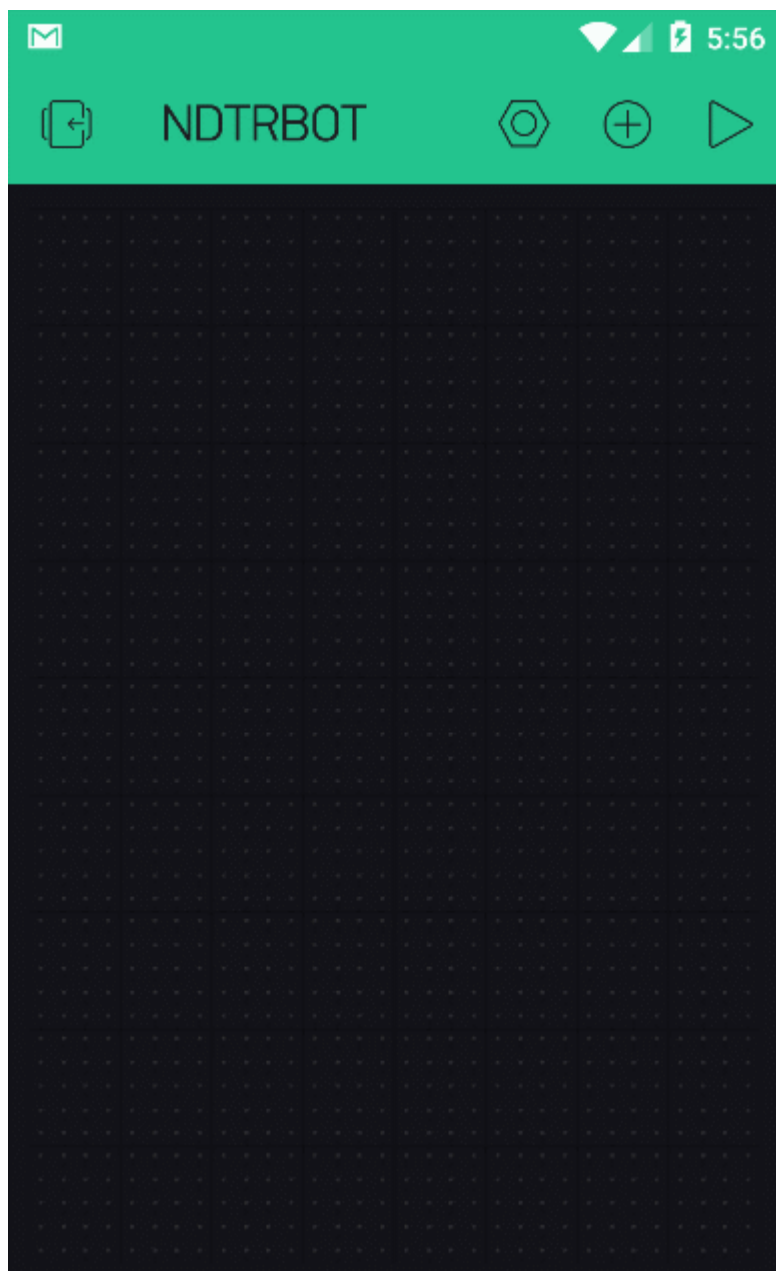
2 hàm này được thêm vào trong hàm `aplyCmd()`, ngoài ra mình còn thêm một cái thông báo trên điện thoại mỗi khi bơm hay đèn được bật thông qua hàm **`Blynk.notify("xxx");`**

Các bạn có thể xem qua một số hình minh họa bên dưới nếu vẫn còn khó khăn trong việc tạo giao diện

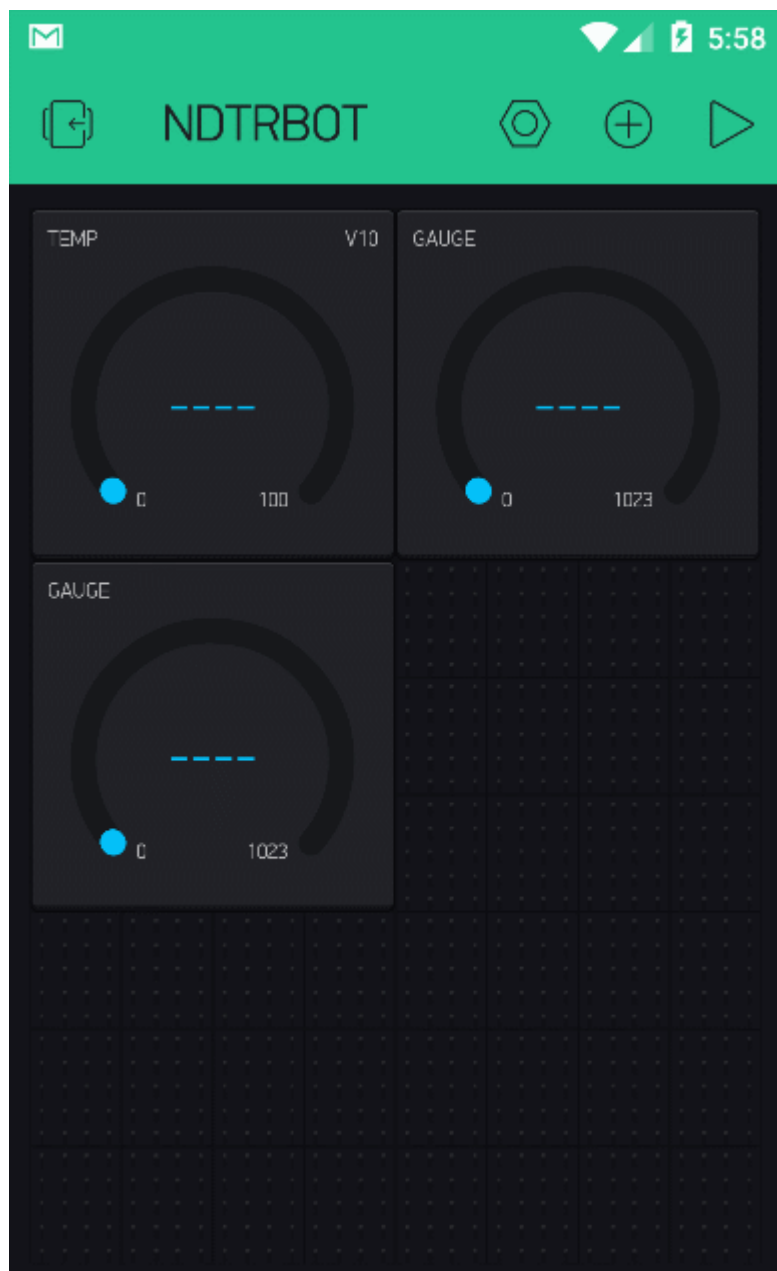
Bước 1: Tạo project



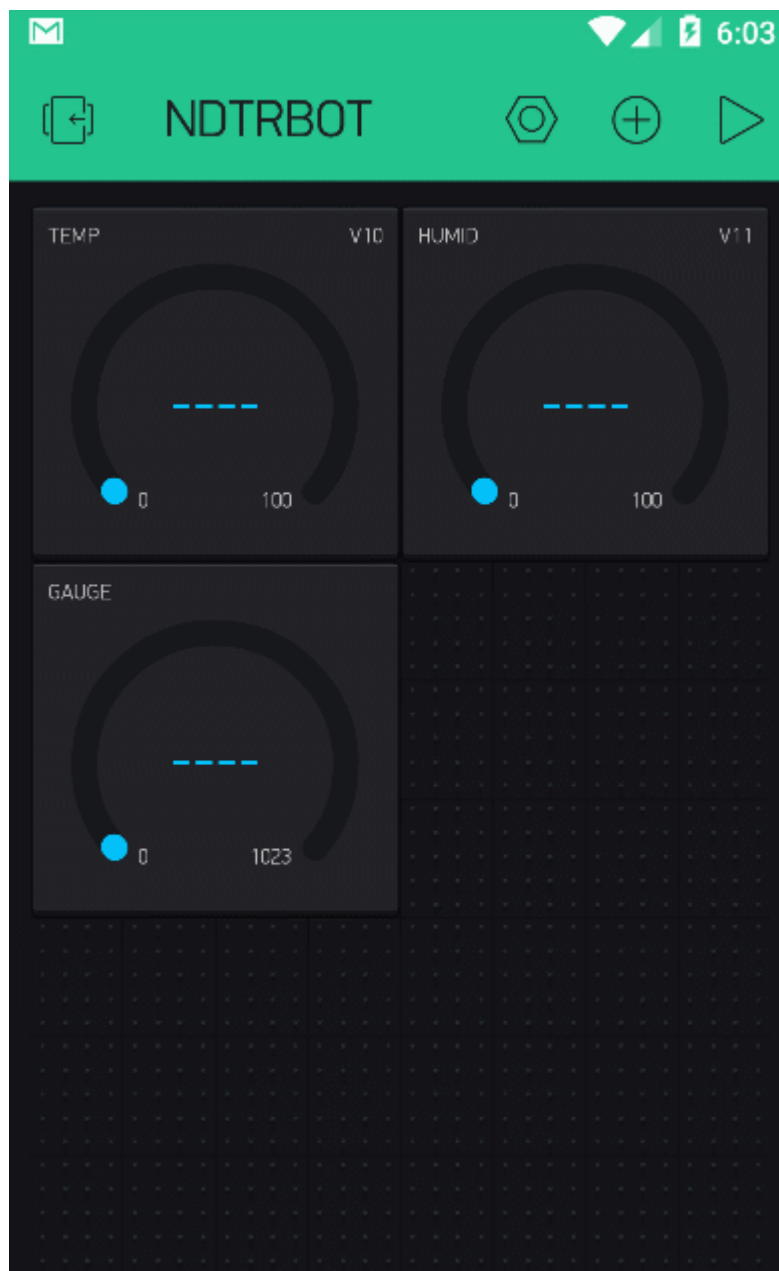
Bước 2: Thêm giao diện hiển thị thông số cảm biến và điều chỉnh thông số nhiệt độ TEMP



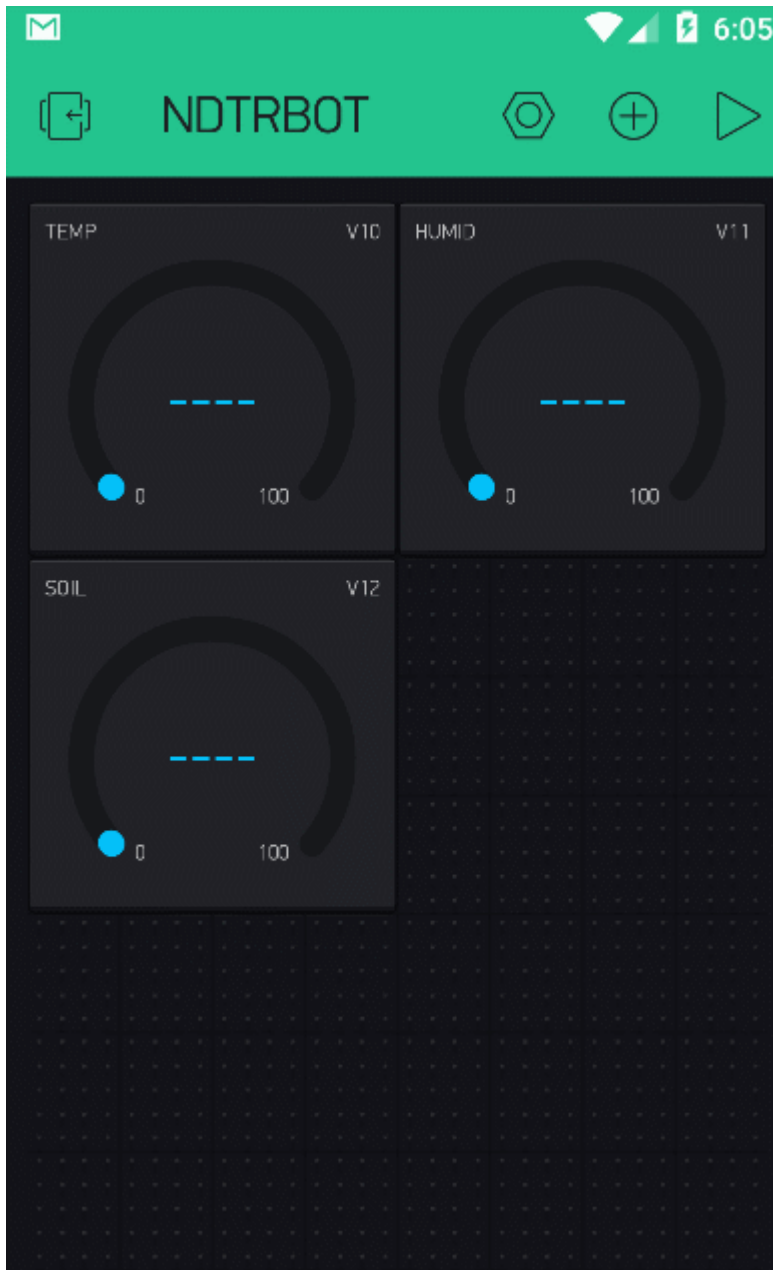
Bước 3: Chỉnh thông số độ ẩm HUMID

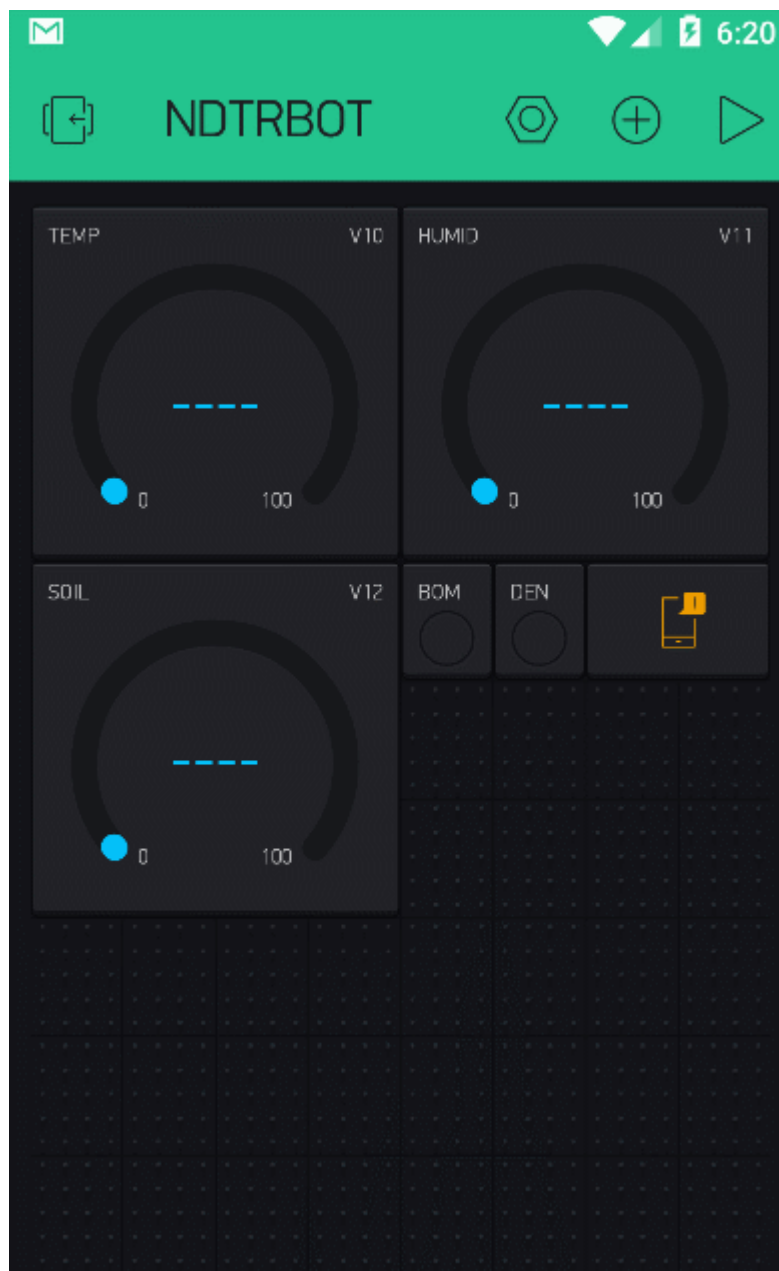


Bước 4: Điều chỉnh thông số cảm biến độ ẩm đất SOIL

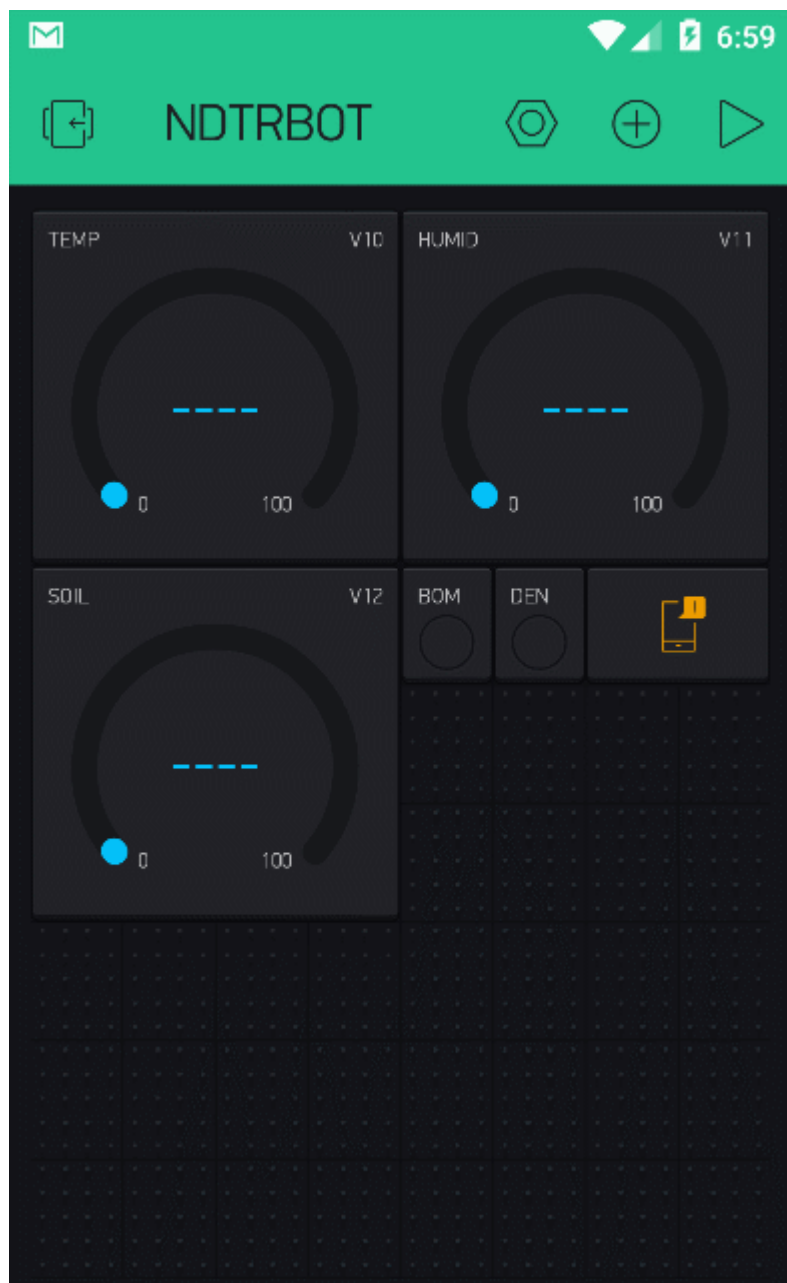


Bước 5: Thêm đèn LED trạng thái và cảnh báo điện thoại

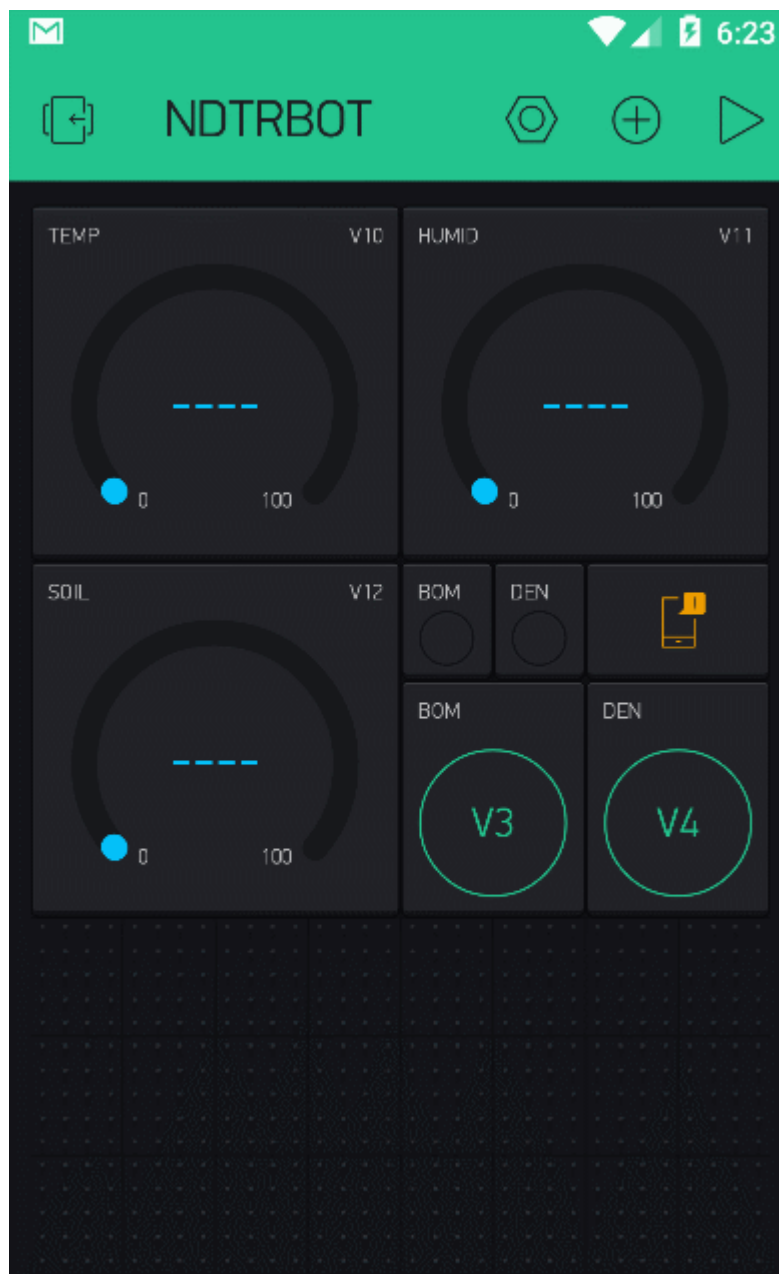




Bước 6: Thêm nút nhấn điều khiển



Bước 7: Thêm đồ thị hiển thị thông tin lưu trữ



Bước 8: Lấy token



Chương trình

```

/* Chương trình đọc nhiệt độ, độ ẩm từ cảm biến DHT
Thêm chức năng đọc cảm biến ánh sáng
Thêm chức năng đọc cảm biến độ ẩm đất
Thêm hiển thị LCD
Thêm chức năng điều khiển tưới tiêu bằng tay 2 bơm
HocARM NDTR BOT for ESP8266 by hocARM.org
// Kết nối
// DHT      | ESP8266
//-----
// VCC(1)   | 3.3V
// DATA(2) | 2
// NC(3)    | x
// GND(4)   | GND
    
```

```
// Nối trỏ 10k giữa chân 1 và chân 2
//-----
//Cảm biến độ ẩm | ESP8266
//-----
// VCC(1)      | 5V
// GND(2)      | GND
// D0(3)       | x
// A0(4)       | A0
*/
/* ESP & Blynk */
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
WidgetLED PUMP(V0); // Đèn trạng thái bơm
WidgetLED LAMP(V1); // Đèn trạng thái đèn sưởi

#include "DHT.h"

/* TIMER */
#include <SimpleTimer.h>

#define DHTPIN D3 // Chân DATA nối với D3

#define SOIL_MOIST_1_PIN A0 // Chân PE4 nối với cảm biến độ ẩm
// Relay, nút nhấn
#define PUMP_ON_BUTTON D0 //Nút điều khiển bằng tay bơm
#define LAMP_ON_BUTTON D1 //Nút điều khiển đèn bằng tay
#define SENSORS_READ_BUTTON D4 //Nút lấy dữ liệu tức thời

#define PUMP_PIN D6 //Bơm
#define LAMP_PIN D7 //Đèn
// Uncomment loại cảm biến bạn sử dụng, nếu DHT11 thì uncomment DHT11
// và comment DHT22
// #define DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
// #define DHTTYPE DHT21 // DHT 21 (AM2301)

/* Thông số cho chế độ tự động */
#define DRY_SOIL 66
#define WET_SOIL 85
#define COLD_TEMP 12
#define HOT_TEMP 22
#define TIME_PUMP_ON 15
#define TIME_LAMP_ON 15
```

```
/* TIMER */
#define READ_BUTTONS_TM 1L // Tương ứng với giây
#define READ_SOIL_HUM_TM 10L //Đọc cảm biến ẩm đất
#define READ_AIR_DATA_TM 2L //Đọc DHT
#define DISPLAY_DATA_TM 10L //Gửi dữ liệu lên terminal
#define SEND_UP_DATA_TM 10L //Gửi dữ liệu lên blynk
#define AUTO_CTRL_TM 60L //Chế độ tự động
//Token Blynk và wifi
char auth[] = "api_token_blynk"; // Blynk token
char ssid[] = "ten_wifi"; //Tên wifi
char pass[] = "password"; //Mật khẩu

// Biến lưu các giá trị cảm biến
float humDHT = 0;
float tempDHT = 0;
//int lumen;
int soilMoist = 0;
// Biến lưu trạng thái bơm
boolean pumpStatus = 0;
boolean lampStatus = 0;

int timePumpOn = 10; // Thời gian bật bơm nước
// Biến cho timer
long sampleTimingSeconds = 50; // ==> Thời gian đọc cảm biến (s)
long startTiming = 0;
long elapsedTime = 0;
// Khởi tạo timer
SimpleTimer timer;

// Khởi tạo cảm biến
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  pinMode(PUMP_PIN, OUTPUT);
  pinMode(LAMP_PIN, OUTPUT);
  pinMode(PUMP_ON_BUTTON, INPUT_PULLUP);
  pinMode(LAMP_ON_BUTTON, INPUT_PULLUP);
  pinMode(SENSORS_READ_BUTTON, INPUT_PULLUP);
  aplyCmd();
  // Khởi tạo cổng serial baud 115200
  Serial.begin(115200);
  Serial.println("HocARM NDTR Bot!");

  dht.begin(); // Bắt đầu đọc dữ liệu
  Blynk.begin(auth, ssid, pass);
  PUMP.off();
```

```
LAMP.off();
startTimers();
}

void loop() {
    timer.run(); // Bật đồng hồ SimpleTimer
    Blynk.run();
}
/*****
* Hàm điều khiển nhận tín hiệu từ blynk
*****/
BLYNK_WRITE(3) // Điều khiển bơm
{
    int i = param.asInt();
    if (i == 1)
    {
        pumpStatus = !pumpStatus;
        applyCmd();
    }
}

BLYNK_WRITE(4) // Điều khiển đèn
{
    int i = param.asInt();
    if (i == 1)
    {
        lampStatus = !lampStatus;
        applyCmd();
    }
}

void getSoilMoist(void)
{
    int i = 0;
    soilMoist = 0;
    for (i = 0; i < 10; i++) //
    {
        soilMoist += analogRead(SOIL_MOIST_1_PIN); //Đọc giá trị cảm biến
        độ ẩm đất
        delay(50); // Đợi đọc giá trị ADC
    }
    soilMoist = soilMoist / (i);
    soilMoist = map(soilMoist, 1023, 0, 0, 100); //Ít nước:0% ==> Nhiều
    nước 100%
}
```

```
void getDhtData(void)
{
    tempDHT = dht.readTemperature();
    humDHT = dht.readHumidity();
    if (isnan(humDHT) || isnan(tempDHT))    // Kiểm tra kết quả lỗi thì
thông báo.
    {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }
}

void printData(void)
{
    // IN thông tin ra màn hình
    Serial.print("Do am: ");
    Serial.print(humDHT);
    Serial.print(" %t");
    Serial.print("Nhiệt độ: ");
    Serial.print(tempDHT);
    Serial.print(" *Ct");

    Serial.print(" %t");
    Serial.print("Do ẩm đất: ");
    Serial.print(soilMoist);
    Serial.println(" %");
}

/*****
Hàm đọc trạng thái bơm và kiểm tra nút nhón
(Nút nhón mặc định là mức "CAO"):
*****/
void readLocalCmd()
{
    boolean digiValue = debounce(PUMP_ON_BUTTON);
    if (!digiValue)
    {
        pumpStatus = !pumpStatus;
        aplyCmd();
    }

    digiValue = debounce(LAMP_ON_BUTTON);
    if (!digiValue)
    {
        lampStatus = !lampStatus;
        aplyCmd();
    }
}
```

```
}

digiValue = debounce(SENSORS_READ_BUTTON);
if (!digiValue)
{
    getDhtData();
    getSoilMoist();
    printData();
}
}
/*****
Thực hiện điểu khiển các bơm
*****/
void aplyCmd()
{
    if (pumpStatus == 1)
    {
        Blynk.notify("NDTRBOT: Canh bao ==>> BOM ON");
        digitalWrite(PUMP_PIN, LOW);
        PUMP.on();
    }

    else {
        digitalWrite(PUMP_PIN, HIGH);
        PUMP.off();
    }

    if (lampStatus == 1)
    {
        Blynk.notify("NDTRBOT: Canh bao ==>> DEN ON");
        digitalWrite(LAMP_PIN, LOW);
        LAMP.on();
    }
    else
    {
        digitalWrite(LAMP_PIN, HIGH);
        LAMP.off();
    }
}
/*****
Hàm kiểm tra trạng thái phím bấm
*****/
boolean debounce(int pin)
{
    boolean state;
    boolean previousState;
```

```
const int debounceDelay = 60;

previousState = digitalRead(pin);
for (int counter = 0; counter < debounceDelay; counter++)
{
    delay(1);
    state = digitalRead(pin);
    if (state != previousState)
    {
        counter = 0;
        previousState = state;
    }
}
return state;
}
/*****
* Ch□ độ tự động dựa trên thông s□ c□m bi□n
*****/
void autoControlPlantation(void)
{
    if (soilMoist < DRY_SOIL)
    {
        turnPumpOn();
    }

    if (tempDHT < COLD_TEMP)
    {
        turnLampOn();
    }
}
/*****
* Bật bơm trong thời gian định s□n
*****/
void turnPumpOn()
{
    pumpStatus = 1;
    aplyCmd();
    delay (TIME_PUMP_ON * 1000);
    pumpStatus = 0;
    aplyCmd();
}

/*****
* Bật đèn trong thời gian định s□n
*****/
void turnLampOn()
```

```
{
  lampStatus = 1;
  aplyCmd();
  delay (TIME_LAMP_ON * 1000);
  lampStatus = 0;
  aplyCmd();
}

/*****
  Khởi động Timers
*****/
void startTimers(void)
{
  timer.setInterval(READ_BUTTONS_TM * 1000, readLocalCmd);
  timer.setInterval(READ_AIR_DATA_TM * 1000, getDhtData);
  timer.setInterval(READ_SOIL_HUM_TM * 1000, getSoilMoist);
  timer.setInterval(SEND_UP_DATA_TM * 1000, sendUptime);
  timer.setInterval(AUTO_CTRL_TM * 1000, autoControlPlantation);
//  timer.setInterval(DISPLAY_DATA_TM * 1000, printData);
}
/*****
  * Gửi dữ liệu lên Blynk
*****/
void sendUptime()
{
  Blynk.virtualWrite(10, tempDHT); //Nhiệt độ với pin V10
  Blynk.virtualWrite(11, humDHT); // Độ ẩm với pin V11
  Blynk.virtualWrite(12, soilMoist); // Độ ẩm đất với V12
}
```

Kết quả sau khi điều chỉnh và làm đẹp thêm 1 chút



Toàn bộ chương trình các bạn có thể tải tại

[hocarm/hocarmNDTRbot](#)

hocARM NDTR Bot - Open Smart agriculture bot. Contribute to [hocarm/hocarmNDTRbot](#) development by creating an account on GitHub.



[hocarmGitHub](#)



Tạm kết

TRUNG TÂM SỬA CHỮA ĐIỆN TỬ QUẢNG BÌNH

MR. XÔ - 0901.679.359 - 80 Võ Thị Sáu, Phường Quảng Thuận, tx Ba Đồn, tỉnh Quảng Bình

GIÁ RẺ

NHANH CHÓNG

LINH KIỆN CHÍNH HÃNG



TRUNG TÂM SỬA CHỮA ĐIỆN TỬ
XÔ NGUYỄN

- Dịch vụ sửa chữa điện tử tại nhà
- Cung cấp linh kiện điện tử
- Tư vấn lắp đặt nhà thông minh

Đc: Quảng Thuận, tx Ba Đồn,
tỉnh Quảng Bình - 0901.679.359

Thế là xong được anh bạn điều khiển qua điện thoại với ESP8266, vẫn còn rất nhiều nhược điểm cần khắc phục như tình trạng delay, trễ quá nhiều khi ấn nút điều khiển, phản hồi hơi chậm với chế độ điều khiển bằng tay. Tuy nhiên nếu sử dụng ở mức cơ bản vẫn đáp ứng được với nhu cầu nghiên cứu và tìm hiểu ứng dụng được ở nhà. Hi vọng sẽ có ích và giúp bạn có bước đầu để phát triển hệ thống tốt hơn lớn hơn.

Tham khảo [1]

Các bài viết tương tự:

1. [Cách khắc phục lỗi. 1:không stand by được 2:stand by sau 2-5s thì quay trở lại màn hình log 3:Mờ biểu tượng stand by. 4: mục system trong windows task manager chêm dụng trên 20% cpu khiến hiệu suất hoạt động máy giảm sút \(lỗi 2 là nguyên nhân cơ bản\) khi mắc lỗi này máy sẽ gặp vấn đề về stand by-computer sleep. - Ai đang gặp những lỗi trên vui lòng liên hệ với tiny mino trên facebook <http://facebook.com/tiny.mino.3> hoặc LH: 016577082380 để được hướng dẫn.](#)
2. [chao cac ban. - dan am thanh KENWOOD rat mong cac ban gop y giúp mình chuyen la the nay mình moi nhan dc cua khách nho sua chua cho dan am thanh kenwood chay radio va doc dia CD nhưngmay khách mang toi trong tình trạng chap phan cong suat .DIA va RADIO van hoat dong binh thuong nhưng bi chap CONG SUAT nen kep loa vao dinh u neu de lau loa se bi chay .hien gio mình van chua dam lam gi ca moikiem tra so bo thi thay chay con STK4150 mình nhìn ma da thay chuoi roi vi hang xach tay ma lai thay con STK4150 mình chua thay gap con nay bao gio vi vay nho cac ban gop y va giúp mình xem tren thi truong co con nay khong vay?tro gioi lieu co ko cac ban nhi?ban nao da tung lam qua ban nay xin giúp do mình mot tay.thank cac ban nhieu.](#)
3. [Hệ thống nông nghiệp thông minh phiên bản Arduino](#)
4. [Hệ thống nông nghiệp thông minh phiên bản ARM TIVA](#)
5. [Main PC-g31 b - Mong tất cả các đồng nghiệp giúp đỡ mình.hiện em nó khởi động không lên màn hình.led báo cây đang hoạt động kg sáng.quạt cpu vẫn quay.cpu và chip bắc,nam vẫn nóng.ram bình thường.các bạn cho mình hướng để sửa chữa em nó nhé.cây này của mình.nên mình muốn tự sửa và đi sâu vào main.mình chuyên tivi.](#)
6. [máy lạnh daikin.... - các bạn cho mình hỏi. ví dụ như 1 máy lạnh đang chạy bình thường nhưng ví 1 lí do nào đó mà lock chạy cao dòng nên thermit ngắt lock. như vậy là điện cấp ra dàn nóng vẫn còn đúng ko các bạn. chỉ có thermit ngắt thôi fai ko hay là boar ngắt điện ra dàn nóng luôn](#)
7. [máy tính của m dùng 2 cây ram 128mb, cpu chắc của intel lúc còn dùng FDD, máy bị lỗi như thế này: + khi cắm dây nguồn vào bộ nguồn, máy tự chạy + nguồn, quạt cpu chạy mà màn hình không lên + đèn bàn phím chớp cái mất luôn \(màn hình ok không bị hư gì hết, mình đem qua màn hình và bàn phím khác thử rồi\) mà nó vẫn không lên màn hình + bàn phím không cháy + trên main không có hiện tượng bị phù tụ gì hết, mình đã tháo 2 cây ram, cpu ra vệ sinh sách sẽ và cắm dây cáp vào thật chắc rồi v=> vậy máy tính m bị gì vậy các bạn, mong các bạn giúp đỡ, mặc dù đầu năm , nhưng m mong sẽ có bạn online cảm ơn cả nhà - bật máy nguồn và quạt chíp chạy](#)
8. [Nạp ga tủ lạnh Sharp - Tủ dùng được 1 năm bảo hành đến thay block.Ban đầu đến dàn nóng. phin. block nóng dữ dội tuyết bám đầu dàn.Em đã kiểm tra block khỏe.Thông súc xịt ngâm bằng nito phin e thay mới, cấp e cân xuống 170 khi nạp ga sò đầu cũng nóng đường hồi động sương mặc dù cho ít ga ạ.](#)
9. [tủ lạnh đông tuyết. \(tủ bảo ôn\) - bực giận, hết ga, mình đã lén giàn nóng riêng. và đã biết giàn nóng bị thủng. giờ mình muốn kiểm tra nguyên giàn lạnh xem có bị thủng ko mà mình ko nghĩ ra cách nào. vì mình mới vào nghề chưa am hiểu và chưa có kinh nghiệm j cả. vì cos một thợ trước đến nhà khách kiểm tra cái tủ này. ong thợ kia phán với chủ nhà là thủng giàn lạnh. giờ mình mới kiểm tra đc mỗi giàn nóng.](#)

Tài liệu này được tải từ website: <http://linhkienthaomay.com>. Zalo hỗ trợ: 0389937723

10. [tủ lạnh hitachi 154l nội địa - xì dàn nóng,, cặp dàn nóng rồi mà bây giờ sao lâu đông đá quá, dàn nóng nóng dữ dội, block thì nóng không chạm tay vào được luôn, phin lọc cũng nóng luôn.](#)
11. [tủ lạnh vtb quá gio vn - tu bị thủng dàn nóng,minh đã thay dàn mới,khi bơm ga thay dàn nóng, nóng nua dàn con nua dàn vệ phin kg nóng.đau dàn lạnh có ít tuyết bám.minh đã kiểm tra 2dàn nónglạnh thay kg bị tuc.](#)
12. [tuyển 5 thợ sửa chữa laptop chuyên nghiệp . 0904677396. Mr : hưng - sửa chữa phần cứng laptop chuyên nghiệp](#)