

Hẳn là các bạn có nghe nói rất nhiều về nông nghiệp thông minh, nhìn dạo quanh trên thị trường Việt Nam cũng có rất nhiều các startup bắt đầu khởi nghiệp với nông nghiệp thông minh, tập trung chủ yếu các chức năng hiện nay là trồng cây thủy canh, giám sát tưới tiêu, tự điều khiển theo thời gian,... có thể điểm qua như [Hachi](#), [Greenbot](#), [Mimosatek](#)..

Mình chưa có điều kiện để sử dụng cũng như trải nghiệm các sản phẩm của các startup trên, nhưng với một số tham khảo có sẵn từ google cũng như dựa trên số đồ sẵn có của mình thử tự làm một hệ thống nông nghiệp thông minh đơn giản, 1 phần để xem nó hoạt động ra làm sao, IOT là cái gì, cách để quan sát, giám sát và điều khiển từ xa như thế nào? 1 phần để chia sẻ cho anh em yêu thích nghiên cứu, tìm hiểu khoa học có thể tự làm cho mình một hệ thống có thể ứng dụng được vào cuộc sống.

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](#)

Giới thiệu

Xuất phát từ một nhu cầu trong thực tế đó là tự trồng rau sạch, ví dụ như mình thích trồng rau xà lách, thì mình cần có được thông tin về nhiệt độ, độ ẩm của môi trường, độ ẩm của đất. Rồi từ các thông tin này thì mình mới có quyết định chính xác là nên bơm thêm nước hay không ? Quá trình bơm nước có thể được thực hiện tự động để đỡ mất thời gian chăm sóc của mình hoặc chế độ bằng tay khi mình muốn tự chăm sóc cây như một cách để giải tỏa stress sau những ngày làm việc mệt mỏi.

Vậy thì trước khi bắt đầu mình cần xác định một số yếu tố như input/output :

Đầu vào (input) của hệ thống gồm

1. Cảm biến

- Nhiệt độ và độ ẩm môi trường
- Độ ẩm đất
- Ánh sáng

1. Nút nhấn

- Nút điều khiển bơm nước
- Nút điều khiển bơm thuốc trừ sâu

Đầu ra hệ thống gồm

1. Cơ cấu chấp hành

- Relay điều khiển bơm nước
- Relay điều khiển bơm thuốc trừ sâu

1. Thông tin phản hồi

- Trạng thái hoạt động của bơm nước
- Trạng thái hoạt động của bơm thuốc trừ sâu

1. Dữ liệu hiển thị

- Thông tin nhiệt độ, độ ẩm, ánh sáng, độ ẩm đất, trạng thái các bơm(bật/tắt)

1. Dữ liệu lưu trữ

- Thông tin nhiệt độ, độ ẩm, ánh sáng, độ ẩm đất, trạng thái các bơm theo thời gian

Sau khi đã xác định xong thì công việc mình **cần làm** là

- Đọc dữ liệu và hiển thị và điều khiển bằng tay (chế độ offline).
- Hiển thị và lưu trữ thông tin trên cloud (chế độ online/ IOT).
- Ứng dụng vào thực tế, thay đổi thông số phù hợp.
- Điều khiển và giám sát qua smartphone.
- Cải tiến và nâng cấp.

Trong bài viết này mình sẽ trình bày 2 phần 1 và 2 trước, phần 3,4,5 mình sẽ cập nhật sau

Chuẩn bị

Phần cứng

- TIVA Launchpad/ Arduino/ STM32F1 (mình làm với TIVA C trước)
- ESP8266 v1 hoặc v7/v12 (mình dùng v1 có sẵn)
- Cảm biến nhiệt độ, độ ẩm DHT22
- Cảm biến ánh sáng (mình tự chế)
- Modul Relay (2 relay)
- LCD Gphone
- Nút nhấn và LED (mình tận dụng có sẵn trên kit TIVA C)
- Dây nối
- Nguồn 3.3V
- Testboard (Breadboard)



Phần mềm

- Energia IDE
- Cloud [thingspeak](#)
- Thư viện [DHT22](#), [Gphone](#), [Simple Timer](#)

Các bài viết cần xem qua

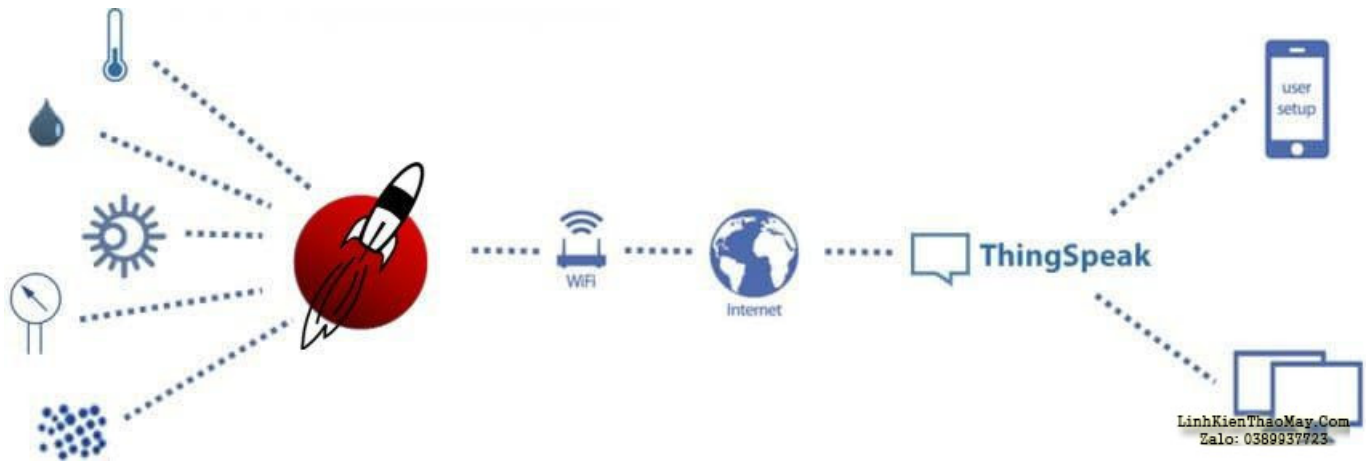
Bài viết sẽ được viết từ dễ tới khó, nếu bạn chưa biết bắt đầu từ đâu hoặc trong quá trình làm việc một số modul bị debug mà không biết kiểm tra thế nào thì có thể tham khảo thêm một số bài viết hướng dẫn từ trước của mình. Mình xin 5 phút quảng cáo dành cho danh sách dưới đây:

- [Học lập trình ARM đơn giản với Energia](#)
- [DHT22, TIVA C và cảm biến độ ẩm đất](#)
- [TIVA C và LCD Gphone/homephone/ST7565](#)
- [ESP8266 cho người không biết gì](#)
- [Hướng dẫn cập nhật firmware AT cho ESP8266](#)
- [Giao tiếp ESP8266 và TIVA C](#)
- [Thingspeak với ESP8266 và TIVA C](#)

Coi như bước chuẩn bị đã ổn, giờ thì mình sẽ đi từng bước một, tranh thủ nói luôn tại sao cần phải có thêm bước này bước kia ở từng phần. Cũng đơn giản nhưng hơi dài nên mọi người chịu khó đọc một chút.

Kết nối IO, lập trình kiểm tra các modul

Mình đăng lại hình minh bạn có thể là dễ hiểu nhất cho hệ thống này



DHT22

Như đã hướng dẫn đọc dữ liệu ở bài trước, mình nói qua về cảm biến này một chút. DHT22 là loại cảm biến để đọc nhiệt độ và độ ẩm của môi trường khá thông dụng hiện nay. Theo [datasheet](#) thì ngưỡng điện áp hoạt động của nó từ 3.3V đến 5V (một số lên tới 6V), nhiệt độ từ -40oC tới +80oC (một số datasheet là 125oC) với độ chính xác +/-0.5oC cho nhiệt độ và +/-2% cho độ ẩm. Chu kỳ lấy dữ liệu trung bình là 2s một lần.

Về chân cắm thì DHT22 có 4 chân, hầu như loại DHT nào cũng có 4 chân và thứ tự chân là giống nhau

- VCC (3-5V)
- Data out
- NC (not connect)
- Ground (GND)

Thông thường thì phải có điện trở 10K kết nối giữa 2 chân Data và VCC, chân output (data out) sẽ được kết nối với chân của vi điều khiển, ở đây là chân PD0 của TIVA C

Tìm hiểu sơ bộ thế thôi, giờ thì mình đưa chương trình đọc dữ liệu từ DHT ở bài [DHT22, TIVA C và cảm biến độ ẩm đất](#) tại đây cho các bạn để quan sát

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

#include "DHT.h"

#define DHTPIN PD_0 // Chân DATA nối với PD0

// 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      | TIVA C
//-----
// VCC(1)   | 3.3V
// DATA(2) | PD0
// NC(3)    | x
// GND(4)   | GND

// Kết nối chân 1 của DHT với 3.3V
// Chân 2 kết nối với các chân nào của TIVA C
// Chân 4 nối với GND
// 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!");
    // Bắt 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;
    }

    // Tính chỉ số nhiệt độ F (mặc định)
    // float hif = dht.computeHeatIndex(f, h);
    // Tính chỉ số nhiệt độ C (isFahreheit = false)
    // float hic = dht.computeHeatIndex(t, h, false);

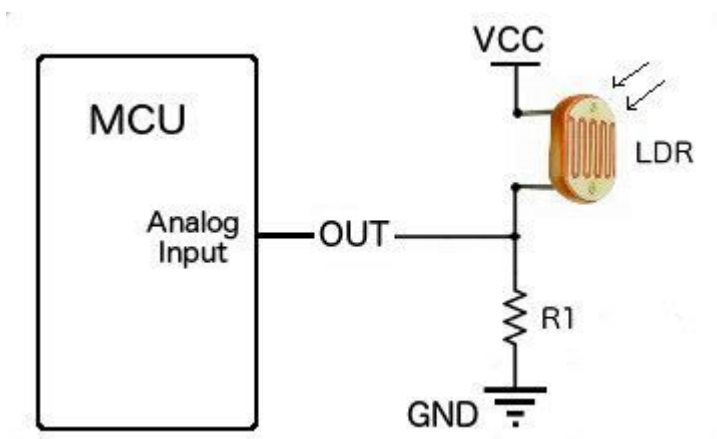
    // IN thông tin ra màn hình
```

```

Serial.print("Do am: ");
Serial.print(h);
Serial.print(" %t");
Serial.print("Nhiệt độ: ");
Serial.print(t);
Serial.println(" *C ");
// Serial.print(f);
// Serial.print(" *Ft");
// Serial.print("Heat index: ");
// Serial.print(hic);
// Serial.print(" *C ");
// Serial.print(hif);
// Serial.println(" *F");
}
    
```

Cảm biến ánh sáng

Sau khi đã kiểm tra hoạt động của DHT xong thì tiếp tục tới cảm biến ánh sáng. Ở đây mình dùng loại cảm biến ánh sáng tự chế từ quang trở, cái này do đơn giản dễ độ chế nên làm luôn cho nhanh.



Quang trở LDR là có điện trở thay đổi khi ánh sáng thay đổi, do đó mình có thể dựa trên chân Analog của TIVA C để và cách đọc giá trị ADC để lấy giá trị này.

Như hình mình có thể quan sát được cảm biến này sẽ có 3 chân

- VCC nguồn cấp sử dụng 3.3V hoặc 5V (mình dùng 3.3V)
- GND
- OUT được kết nối với chân PE3 của TIVA

Mình viết một hàm để đọc giá trị trung bình của cảm biến ánh sáng với ngưỡng giá trị ADC là từ 0-4095, sẽ được 'map' với cường độ ánh sáng là từ 0 - 100%, nó giống như hình thức chuyển đổi tương ứng như sau:

- "Trời tối hẳn": Giá trị ADC: 4095 => 0%
- "Trời sáng hẳn": Giá trị ADC: 0 => 100%

Hàm đọc cảm biến ánh sáng int getLumen(int anaPin)

Kết hợp với DHT22, ta thêm hàm đọc cảm biến ánh sáng, ta lấy 3 thông số là nhiệt độ, độ ẩm, ánh sáng, hiển thị debug ra màn hình máy tính xem thế nào

```
// Chương trình đọc nhiệt độ, độ ẩm từ cảm biến DHT
// Thêm chức năng đọc cảm biến ánh sáng
// Written by ladyada, public domain
// Chuyển sửa cho TIVA C bởi hocARM.org

#include "DHT.h"

#define DHTPIN PD_0 // Chân DATA nối với PD0
#define LDR_PIN PE_3 // Chân PE3 nối với chân OUT cảm biến as
// 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 | TIVA C
//-----
// VCC(1) | 3.3V
// DATA(2) | PD0
// NC(3) | x
// GND(4) | GND

// Kết nối chân 1 của DHT với 3.3V
// Chân 2 kết nối với các chân nào của TIVA C
// Chân 4 nối với GND
// 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!");
    // Bắt đầu đọc dữ liệu
    dht.begin();
}

void loop() {
    // Đọc chuyển đổi dữ liệu khoảng 2s
```

```
delay(2000);

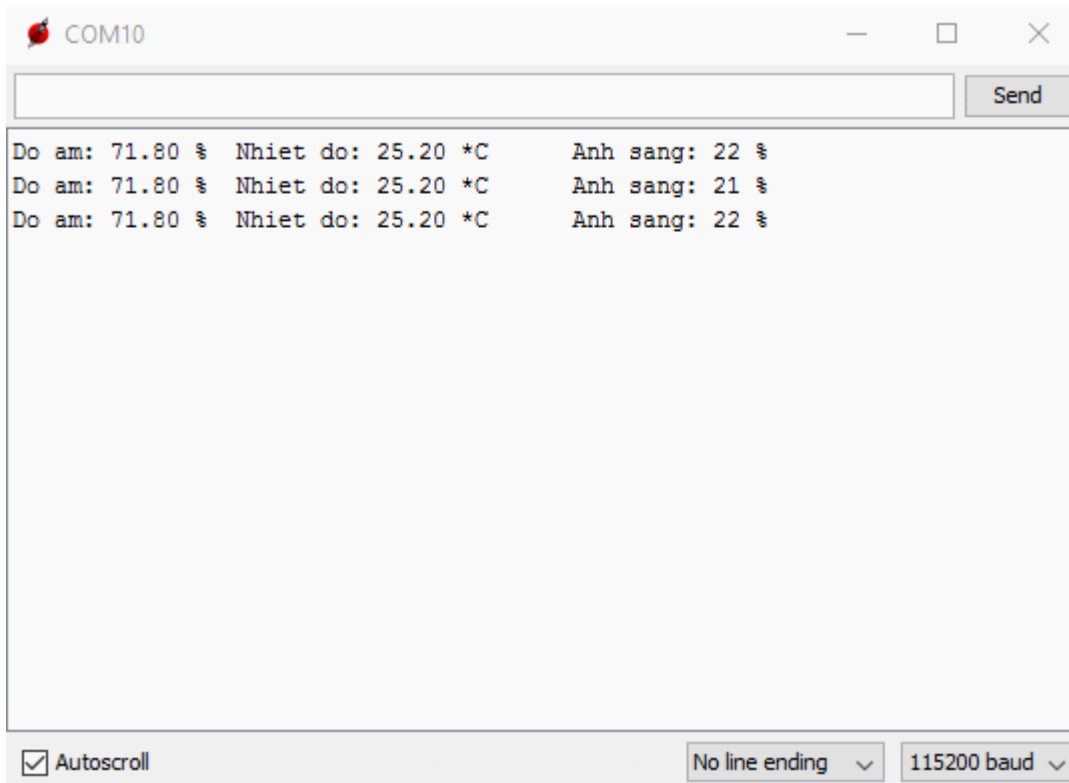
float h = dht.readHumidity();
// Đọc giá trị nhiệt độ C (mặc định)
float t = dht.readTemperature();
// Đọc giá trị nhiệt độ F (isFahrenheit = true)
float f = dht.readTemperature(true);

// 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;
}

// Tính chỉ số nhiệt độ F (mặc định)
// float hif = dht.computeHeatIndex(f, h);
// Tính chỉ số nhiệt độ C (isFahreheit = false)
// float hic = dht.computeHeatIndex(t, h, false);
// Đọc dữ liệu cảm biến ánh sáng
int lumen = getLumen(LDR_PIN);
// IN thông tin ra màn hình
Serial.print("Do am: ");
Serial.print(h);
Serial.print(" %t");
Serial.print("Nhiệt độ: ");
Serial.print(t);
Serial.print(" *Ct");
Serial.print("Ánh sáng: ");
Serial.print(lumen);
Serial.println(" %");
// Serial.print(f);
// Serial.print(" *Ft");
// Serial.print("Heat index: ");
// Serial.print(hic);
// Serial.print(" *C ");
// Serial.print(hif);
// Serial.println(" *F");
}
int getLumen(int anaPin)
{
    int anaValue = 0;
    for (int i = 0; i < 10; i++) // Đọc giá trị cảm biến 10 lần và lấy
    giá trị trung bình
    {
        anaValue += analogRead(anaPin);
        delay(50);
    }
}
```

```
}  
anaValue = anaValue / 10;  
anaValue = map(anaValue, 4095, 0, 0, 100); //T[i]:0 ==> Sáng 100%  
  
return anaValue;  
}
```

Kết quả



Cảm biến độ ẩm đất

Coi như xong 2 bạn cảm biến, giờ tới cảm biến độ ẩm đất, cách đọc dữ liệu của nó cũng khá giống với cảm biến ánh sáng, thuận tiện ở đây là modul thiết kế khá ổn rồi nên chỉ việc lấy tín hiệu về để xác định là nhiều nước hay ít nước/ đất khô để tác động bơm nước thêm thôi.

Chân cổng của cảm biến độ ẩm đất:

- VCC (3.3V - 5V)
- GND
- D0 đầu ra tín hiệu số (ở đây mình không dùng)
- A0 đầu ra analog kết nối với chân PE4 của TIVA C

Mình viết một hàm để đọc giá trị trung bình của cảm biến độ ẩm đất tương tự như cảm biến ánh sáng với ngưỡng giá trị ADC là từ 0-4095, sẽ được 'map' với ngưỡng độ ẩm đất là từ 0 - 100% (nhiều nước giá trị thấp, ít nước thì giá trị cao), nó giống như hình thức chuyển đổi tương ứng như sau:

- “Ít nước”: Giá trị ADC: 4095 => 0%
- “Nhiều nước”: Giá trị ADC: 0 =>100%

Hàm đọc cảm biến độ ẩm đất `intgetSoilMoist()`

Giờ thì thêm bạn cảm biến độ ẩm này kết hợp với DHT22 và cảm biến ánh sáng ta sẽ có được khá khá dữ liệu rồi, mình gom hết tất cả vào một hàm đọc cảm biến. Hàm đọc tất cả cảm biến `voidreadSensors(void)`

Mình sẽ lấy tất cả các thông tin từ cảm biến và in ra màn hình với 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 độ ẩm đất
// Written by ladyada, public domain
// Chuyển sã cho TIVA C bởi hocARM.org
// Kết nối
// DHT      | TIVA C
//-----
// VCC(1)   | 3.3V
// DATA(2) | PD0
// NC(3)    | x
// GND(4)   | GND

// Kết nối chân 1 của DHT với 3.3V
// Chân 2 kết nối với các chân nào của TIVA C
// Chân 4 nối với GND
// Nối trở 10k giữa chân 1 và chân 2
#include "DHT.h"

#define DHTPIN PD_0      // Chân DATA nối với PD0
#define LDR_PIN PE_3     // Chân PE3 nối với chân OUT cảm biến as
#define SOIL_MOIST_1_PIN PE_4 // Chân PE4 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() {
```

```
// Khai 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("Độ ẩm đất: ");
  Serial.print(soilMoist);
  Serial.println(" %");
}

int getLumen(int anaPin)
{
  int anaValue = 0;
  for (int i = 0; i < 10; i++) // Đọc giá trị cảm biến 10 lần và lấy
  giá trị trung bình
  {
    anaValue += analogRead(anaPin);
    delay(50);
  }

  anaValue = anaValue / 10;
  anaValue = map(anaValue, 4095, 0, 0, 100); //Tỉ lệ:0 ==> Sáng 100%

  return anaValue;
}

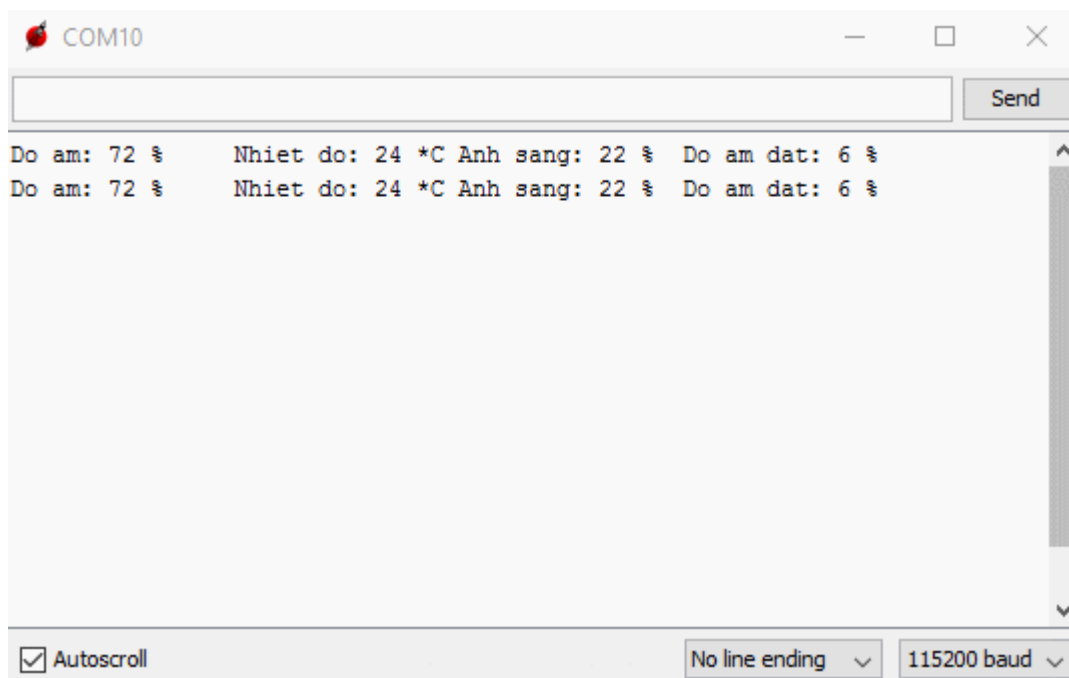
int getSoilMoist()
{
  int i = 0;
  int anaValue = 0;
  for (i = 0; i < 10; i++) //
  {
    anaValue += analogRead(SOIL_MOIST_1_PIN); //Đọc giá trị cảm biến
    độ ẩm đất
  }
}
```

```

        delay(50); // Đợi đọc giá trị ADC
    }
    anaValue = anaValue / (i);
    anaValue = map(anaValue, 4095, 0, 0, 100); //Ít nước:0% ==> Nhi u
nước 100%
    return anaValue;
}
void readSensors(void)
{
    tempDHT = dht.readTemperature(); //Đọc nhiệt độ DHT22
    humDHT = dht.readHumidity(); //Đọc độ  m DHT22
    lumen = getLumen(LDR_PIN); //Đọc ánh sáng
    soilMoist = getSoilMoist(); //Đọc c m bi n độ  m đ t
}

```

Kết quả



The screenshot shows a serial monitor window titled 'COM10'. The output displays two lines of sensor data: 'Do am: 72 %', 'Nhiet do: 24 *C', 'Anh sang: 22 %', and 'Do am dat: 6 %'. The window includes a 'Send' button, an 'Autoscroll' checkbox, and settings for 'No line ending' and '115200 baud'.

Thêm LCD

Sau khi đã lấy được thông tin và hiển thị lên máy tính ổn rồi thì có một vấn đề phát sinh là muốn kiểm tra dữ liệu cảm biến thì lúc nào cũng phải kè kè theo 1 cái máy tính, mà cái máy tính của mình thì nó còn dùng cho nhiều việc khác nữa, giờ nếu muốn cho bạn bè xem nhiệt độ, độ ẩm đang là bao nhiêu không lẽ lại bảo “Để đi khởi động máy tính mới xem được”, giải pháp để khắc phục được tình trạng này là dùng thêm một LCD để hiển thị thông tin, bạn có thể lựa chọn các LCD nào, ban đầu mình có chọn LCD 16×2 nhưng chỉ hiển thị được có 2 dòng, không quan sát được hết, nên phải chuyển qua xài cái LCD Gphone cho thoải mái.

Trong bài [TIVA C và LCD Gphone/homephone/ST7565](http://linhkienthaomay.com) mình có hướng dẫn về cách kết nối và kiểm tra LCD, mình vẫn giữ nguyên kết nối như cũ, mình chỉ có chút lưu ý là LCD này có 10 chân, và hoạt động với điện áp 3.3V, 2 chân LED K+ và A được dùng để bật đèn nền cho

LCD

- GND
- RST
- SCK
- A0(C/D)
- LED K+
- LED A
- SDA
- CS
- VCC
- GND

Vì quá trình hiển thị thông tin cần khá nhiều trong chương trình nên mình sẽ viết một hàm để hiển thị các thông tin này để thuận tiện cho quá trình tái sử dụng `void showDataLCD(void)`. Có hàm hiển thị LCD rồi thì giờ vừa hiển thị trên LCD, vừa hiển thị debug dữ liệu từ cảm biến gửi lên và đối chiếu xem có giống nhau không

```
/* 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
HocARM NDTR BOT by hocARM.org
```

```
-----
Kết nối
```

```
DHT      | TIVA C
```

```
-----
VCC(1)   | 3.3V
```

```
DATA(2)  | PD0
```

```
NC(3)    | x
```

```
GND(4)   | GND
```

```
Kết nối chân 1 của DHT với 3.3V
```

```
Chân 2 kết nối với các chân nào của TIVA C
```

```
Chân 4 nối với GND
```

```
Nối trở 10k giữa chân 1 và chân 2
```

```
-----
So thu tu GLCD và kết nối với TIVA
```

```
// Pin | Chức năng | TIVA C
```

```
//-----
```

```
// 1 | GND | GND
```

```
// 2 | RST | PE1
```

```
// 3 | SCK | PD2
```

```
// 4 | A0(C/D) | PD3
```

```
// 5 | LED K+ (3.3V) | 3.3V
```

```
// 6 | LED A (GND) | GND
```

```
// 7 | SDA | PD1
```

```
// 8 | CS | PE2
```


0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0xC0, 0xC0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x07, 0x1F,
0xFF, 0xF3, 0x87, 0x87, 0xC7,
0xC6, 0xEE, 0xFC, 0xFC, 0xFE, 0xFF, 0x3F, 0x18, 0x19, 0x3B, 0x37,
0x3E, 0xBC, 0xF8, 0xF8, 0xF8,
0xF0, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0xF0, 0xFF, 0xBF, 0xC3, 0xE7, 0x7E, 0x7C, 0x7C, 0x7C, 0xFC, 0xCE,
0x87, 0xFF, 0xFF, 0xFD, 0x9E,
0x87, 0xF9, 0xFF, 0x1F, 0x80, 0xC0, 0xE0, 0x70, 0xF8, 0xFE, 0x7F,
0x1F, 0x07, 0x01, 0x07, 0x3F,
0xFC, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFF, 0xFF, 0xFF, 0xDE, 0xC0,
0xC0, 0xC0, 0xC0, 0xDF, 0xFF, 0xFF, 0xFF, 0x00, 0xC0, 0xE0, 0xF0,
0x70, 0x38, 0x38, 0x38, 0x78,
0xF0, 0xE0, 0xC0, 0x00, 0xE0, 0xF0, 0xF0, 0x78, 0x38, 0x38, 0x78,
0x70, 0x70, 0x00, 0x00, 0x80,
0xE0, 0xF8, 0xFE, 0x3F, 0x3F, 0xFF, 0xFC, 0xE0, 0x80, 0x00, 0x00,
0x00, 0xFF, 0xFF, 0xFF, 0x87,
0x87, 0x87, 0xCF, 0xCE, 0xFE, 0xFC, 0x78, 0x00, 0x00, 0xFF, 0xFF,
0xFF, 0x7F, 0xFC, 0xF0, 0xC0,
0x80, 0xE0, 0xF8, 0xFE, 0x3F, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x1F,
0xFF, 0xF1, 0x81, 0x00, 0x80, 0xC0, 0xF0, 0x78, 0x1E, 0x8F, 0xC3,
0xC7, 0xEE, 0x7F, 0xFF, 0xFF,
0xFF, 0x3E, 0x37, 0xF3, 0xF1, 0x3C, 0x3F, 0x37, 0x31, 0x30, 0x30,
0x30, 0x30, 0x30, 0x30, 0x31,
0x3F, 0x3F, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F,
0x3F, 0x3F, 0x3D, 0x01, 0x01,
0x01, 0x01, 0x3D, 0x3F, 0x3F, 0x3F, 0x00, 0x07, 0x0F, 0x1F, 0x3C,
0x38, 0x38, 0x38, 0x3C, 0x1F,
0x1F, 0x07, 0x00, 0x0F, 0x1F, 0x1F, 0x38, 0x38, 0x38, 0x38, 0x1C,
0x1C, 0x30, 0x3E, 0x3F, 0x1F,
0x07, 0x07, 0x07, 0x07, 0x07, 0x07, 0x1F, 0x3F, 0x3E, 0x30, 0x00,
0x3F, 0x3F, 0x3F, 0x03, 0x03,


```
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
};
```

```
int humDHT;
int tempDHT;
int lumen;
int soilMoist;
// Khai tạo cảm biến
DHT dht(DHTPIN, DHTTYPE);
// Khai tạo LCD
homephone lcd (PD_1, PD_2, PD_3, PE_1, PE_2);
void setup() {
    // Khai tạo cổng serial baud 115200
    Serial.begin(115200);
    Serial.println("HocARM NDTR Bot!");
    // Bật đầu đọc dữ liệu
    dht.begin();
    lcd.begin(); //Khai tạo LCD
    lcd.setContrast(0x0A); //Đặt độ tương phản
    lcd.clear(); //Xóa màn hình
    lcd.drawBitmap(0, 0, 127, 63, ha_lg, black); // Hiện thị hình bitmap
    lcd.display();
}
```

```
void loop() {
    delay(2000);
    readSensors();
    printData();
    showDataLCD();
}
int getLumen(int anaPin)
{
    int anaValue = 0;
    for (int i = 0; i < 10; i++) // Đọc giá trị cảm biến 10 lần và lấy
    giá trị trung bình
    {
        anaValue += analogRead(anaPin);
        delay(50);
    }

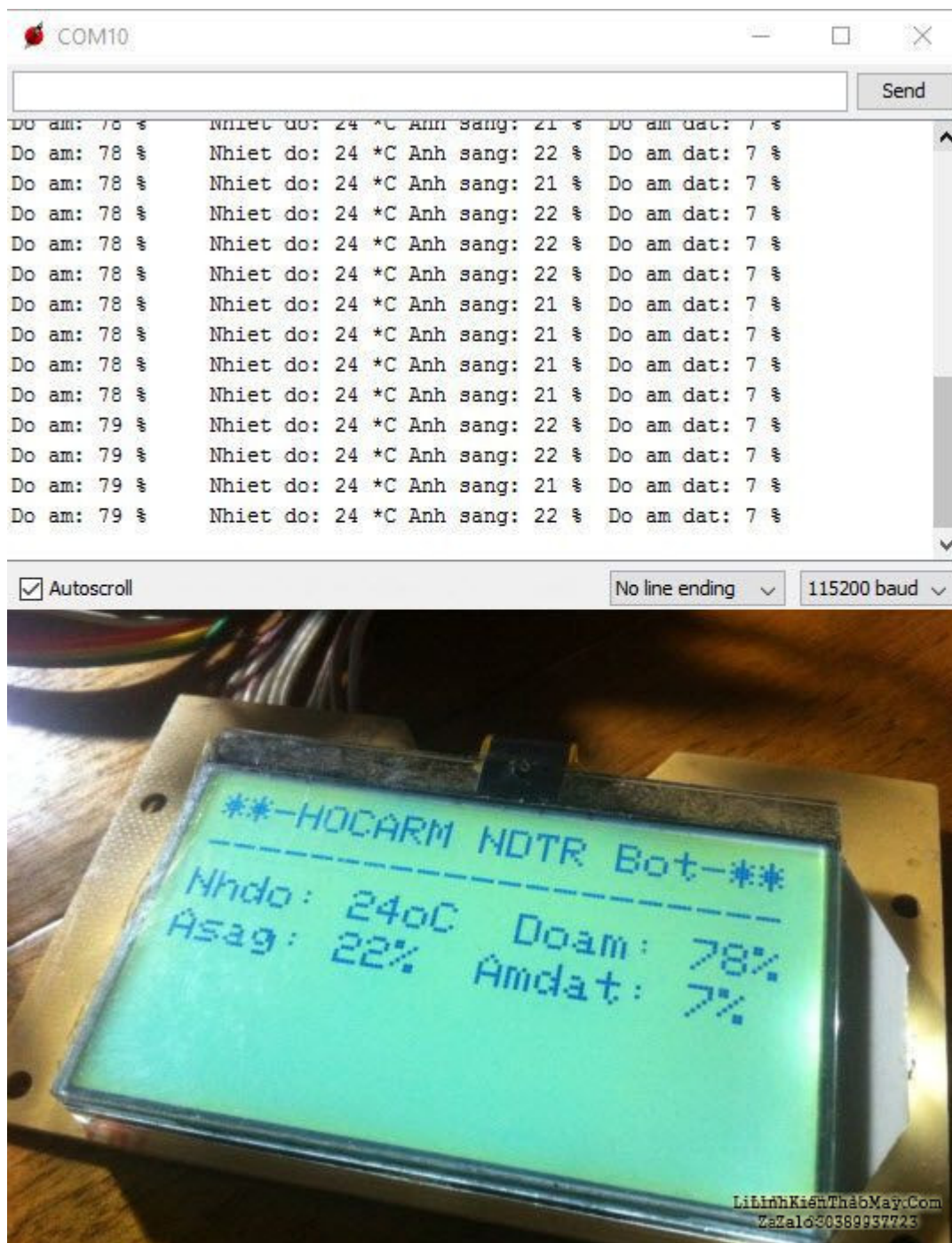
    anaValue = anaValue / 10;
    anaValue = map(anaValue, 4095, 0, 0, 100); //Tỉ lệ:0 ==> Sáng 100%

    return anaValue;
}
```

```
}
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, 4095, 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 showDataLCD(void)
{
    lcd.clear(); //Xóa màn hình
    lcd.setTextColor(black, white);
    lcd.setCursor(0, 0); //đặt con trỏ vào tọa độ (0,0)
    lcd.print("**-HOCARM NDTR Bot-**");
    lcd.setCursor (0, 10);
    lcd.print("-----");
    lcd.setCursor (0, 20);
    lcd.print("Nhdo: ");
    lcd.print(tempDHT);
    lcd.print("oC Doam: ");
    lcd.print(humDHT);
    lcd.print("% ");
    lcd.setCursor (0, 30);
    lcd.print("Asag: ");
    lcd.print(lumen);
    lcd.print("%");
    lcd.print(" Amdat: ");
    lcd.print(soilMoist);
    lcd.print("%");
    lcd.display();
}
```

```
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(" %");
}
```

Kết quả



Thêm Relay và nút nhấn

Vậy là mình đã đọc được dữ liệu từ các cảm biến và hiển thị được thông tin lên serial và LCD. Coi như các thông số đã nằm trong tay mình, giờ mình muốn thêm chức năng điều khiển bơm tưới nước và bơm trừ sâu bằng tay thì phải làm thế nào ? Thôi thì cải tiến thêm chút nữa.

Ở đây mình sử dụng modul 2 Relay và 2 nút nhấn có sẵn trên kit để điều khiển bơm nước và bơm thuốc trừ sâu

Nói qua về Modul relay này có 4 chân cần quan tâm là 2 chân nguồn (5V) và 2 chân cho việc điều khiển là IN1, IN2

- GND
- IN1
- IN2
- VCC

6 chân output của modul dành cho 2 relay, mỗi relay 3 chân là NO(Normal Open),Ref(Reference) và NC(Normanl Closed), mà ta hay quen gọi là tiếp điểm thường đóng, thường mở. Thông thường sử dụng NO và Ref để nối với nguồn 220V. Ngoài ra trên modul còn có 2 LED báo hiệu hoạt động của relay.Vì mình không có bơm nước cũng như bơm thuốc sâu nên phần output chỉ là điều khiển đóng/ cắt relay thôi.

Để điều khiển được 2 bơm hoạt động một cách thủ công thì mình cần có 2 nút nhấn để điều khiển, 2 nút nhấn này sẽ được hoạt động dựa trên trạng thái của bơm. Nghĩa là như thế nào ?

- Giả sử ban đầu bơm nước chưa hoạt động, trạng thái của bơm sẽ là mức 0, khi có người nhấn nút sẽ kiểm tra trạng thái này, nếu đúng là 0 thì sẽ thực hiện “BẬT”
- Ngược lại sẽ thực hiện “TẮT”

Mỗi khi mình chạy vòng loop() thì hàm readLocalCmd() sẽ được thực hiện kiểm tra xem nút nhấn có được nhấn không, sau đó cập nhật trạng thái của các bơm (2 biến là pumpWaterStatus và pumpPesStatus). Hàm debounce(pin) được dùng thay thế hàm digitalRead(pin) nhằm mục đích hạn chế việc đọc sai từ nút nhấn.

Khi nút nhấn được ấn, thì một hàm khác được gọi là hàm applyCmd(). Hàm này sẽ dựa trên trạng thái của bơm hiện có để thực hiện BẬT hoặc TẮT các bơm.

Mình thử chương trình ấn nút điều khiển bơm trước khi ghép vào chung với chương trình đọc cảm biến, ở đây mình tận dụng sẵn 2 nút bấm PF0 và PF4 có trên kit TIVA C luôn cho đỡ tốn dây nối ?

```
/* 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
```

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

```
-----  
Kit nối
```

```
DHT      | TIVA C
```

```
-----  
VCC(1)   | 3.3V
```

```
DATA(2)  | PD0
```

```
NC(3)    | x
```

```
GND(4)   | GND
```

```
Kit nối chân 1 của DHT với 3.3V
```

```
Chân 2 kit nối với các chân nào của TIVA C
```

```
Chân 4 nối với GND
```



```
0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
};
// Biến lưu các giá trị cảm biến
int humDHT;
int tempDHT;
int lumen;
int soilMoist;
// Biến lưu trạng thái bơm
boolean pumpWaterStatus = 0;
boolean pumpPesStatus = 0;
int timePumpOn = 10; // Thời gian bật bơm nước

// Khởi tạo cảm biến
DHT dht(DHTPIN, DHTTYPE);
// Khởi tạo LCD
homephone lcd (PD_1, PD_2, PD_3, PE_1, PE_2);
void setup() {
    pinMode(PUMPW_PIN, OUTPUT);
    pinMode(PUMPS_PIN, OUTPUT);
    pinMode(PUMPW_ON, INPUT_PULLUP);
    pinMode(PUMPS_ON, INPUT_PULLUP);
    // Khởi tạo cổng serial baud 115200
    Serial.begin(115200);
    Serial.println("HocARM NDTR Bot!");
    // Bật đọc dữ liệu
```

```
dht.begin();
lcd.begin(); //Khởi tạo LCD
lcd.setContrast(0x0A); //Đặt độ tương phản
lcd.clear(); //Xóa màn hình
lcd.drawBitmap(0, 0, 127, 63, ha_lg, black); // Hiển thị hình bitmap
lcd.display();
}

void loop() {
  readSensors();
  // printData();
  readLocalCmd();
  showDataLCD();
}

int getLumen(int anaPin)
{
  int anaValue = 0;
  for (int i = 0; i < 10; i++) // Đọc giá trị cảm biến 10 lần và lấy
giá trị trung bình
  {
    anaValue += analogRead(anaPin);
    delay(50);
  }

  anaValue = anaValue / 10;
  anaValue = map(anaValue, 4095, 0, 0, 100); //Tỉ lệ:0 ==> Sáng 100%

  return anaValue;
}

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, 4095, 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 showDataLCD(void)
{
    lcd.clear();                  //Xóa màn hình
    lcd.setTextColor(black, white);
    lcd.setCursor(0, 0);         //đặt con trỏ vào tọa độ (0,0)
    lcd.print("***-HOCARM NDTR Bot-***");
    lcd.setCursor (0, 10);
    lcd.print("-----");
    lcd.setCursor (0, 20);
    lcd.print("Nhdo: ");
    lcd.print(tempDHT);
    lcd.print("oC Doam: ");
    lcd.print(humDHT);
    lcd.print("% ");
    lcd.setCursor (0, 30);
    lcd.print("Asag: ");
    lcd.print(lumen);
    lcd.print("%");
    lcd.print(" Amdat: ");
    lcd.print(soilMoist);
    lcd.print("%");
    lcd.setCursor (0, 40);
    lcd.print("BomN: ");
    lcd.print(pumpWaterStatus);
    lcd.print(" BomS: ");
    lcd.print(pumpPesStatus);
    lcd.display();
    lcd.setCursor (0, 0);
    lcd.display();
}
void printData(void)
{
    // IN thông tin ra màn hình
    Serial.print("Do am: ");
    Serial.print(humDHT);
    Serial.print(" %t");
    Serial.print("Nhiet do: ");
    Serial.print(tempDHT);
    Serial.print(" *Ct");
    Serial.print("Anh sang: ");
    Serial.print(lumen);
}
```

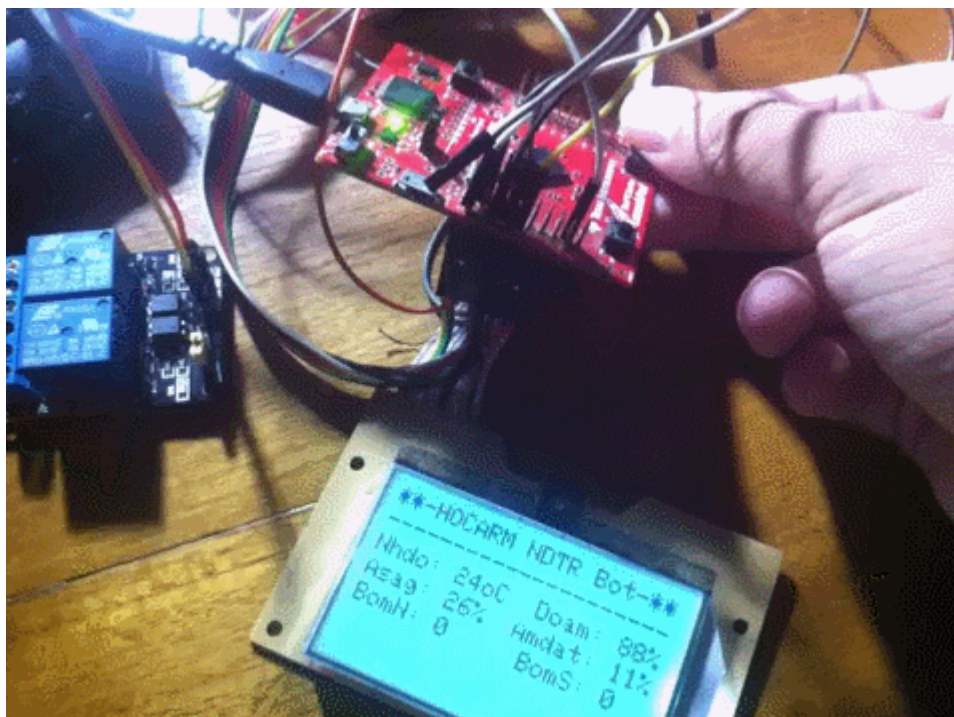
```
Serial.print(" %t");
Serial.print("Do am dat: ");
Serial.print(soilMoist);
Serial.println(" %");
}
/*****
Hàm bật bơm nước
*****/
void turnPumpOn()
{
    digitalWrite(PUMPW_PIN, HIGH);
    pumpWaterStatus = 1;
    showDataLCD();
    delay (timePumpOn * 1000);
    digitalWrite(PUMPW_PIN, LOW);
    pumpWaterStatus = 0;
    showDataLCD();
}
/*****
Hàm đọc trạng thái bơm và kiểm tra nút nhàn
(Nút nhàn mặc định là mức "CAO"):
*****/
void readLocalCmd()
{
    int digiValue = debounce(PUMPW_ON);
    if (!digiValue)
    {
        pumpWaterStatus = !pumpWaterStatus;
        showDataLCD();
        aplyCmd();
    }
    digiValue = debounce(PUMPS_ON);
    if (!digiValue)
    {
        pumpPesStatus = !pumpPesStatus;
        showDataLCD();
        aplyCmd();
    }
}
/*****
Thực hiện điểu khiển các bơm
*****/
void aplyCmd()
{
    if (pumpWaterStatus == 1) digitalWrite(PUMPW_PIN, HIGH);
    if (pumpWaterStatus == 0) digitalWrite(PUMPW_PIN, LOW);
}
```

```
if (pumpPesStatus == 1) digitalWrite(PUMPS_PIN, HIGH);
if (pumpPesStatus == 0) digitalWrite(PUMPS_PIN, LOW);
}
/*****
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;
}
```

Kết quả

Ở đây mình dùng 2 đèn báo hiệu có sẵn trên kit mô phỏng cho 2 relay hoạt động, các bạn có thể thấy mình ấn nút khá lâu thì mới nhận, mình sẽ khắc phục cái này ở chương trình chạy offline thủ công



Hệ thống chạy offline thủ công

Kiểm tra từng cái hơi vất vả, chúc mừng bạn nếu đã cố gắng được tới đây, phần trên mình đã làm được 3 nhiệm vụ “lớn”

- Đọc được tất cả các cảm biến
- Điều khiển được các bơm thủ công thông qua nút nhấn
- Hiển thị thông tin lên LCD và máy tính

Với các tín hiệu như nhiệt độ, độ ẩm, thì thường phải đợi khoảng 2s, nếu thời gian có tăng lên 1 phút thì cũng không vấn đề gì lắm, ngoài ra tín hiệu về ánh sáng, độ ẩm đất cũng không yêu cầu quá khắt khe về thời gian. Nhưng với các bơm thì cần phải được điều khiển cũng như đọc trạng thái nhanh chóng, ví dụ như nếu đọc trạng thái không đúng thì nhiều lúc tưởng đã bơm rồi nhưng thực ra chưa bơm -> cây hư, hoặc tưởng đã tắt bơm nhưng chưa tắt, dẫn tới bơm quá nhiều nước -> ngập úng -> cây hư.

Do đó ở đây mình sẽ sử dụng hàm `millis()`, hàm này có chức năng tương tự như timer để định thời gian bao nhiêu lâu thì làm lại một nhiệm vụ.

Ở đây trong vòng `loop()` mình sẽ luôn kiểm tra xem nút nhấn có được ấn hay không, và thực hiện 50s lấy dữ liệu từ các cảm biến 1 lần. Bằng cách này có thể khắc phục được phần nào hiện tượng ấn phím một lúc mới nhận mình đã nói ở trên

Chương trình để thực hiện lấy dữ liệu 5s/lần.

Giờ là lúc ghép nối toàn bộ những gì đã làm bên trên thành một và xem kết quả để lấy động lực chiến đấu tiếp.

Lưu ý:

Các bạn lưu ý là mình có một số thay đổi mức ở hàm aplyCmd() từ mức LOW đổi thành HIGH và ngược lại, sở dĩ phải đổi vì modul relay đang dùng hoạt động khi kích mức LOW

Ngoài ra mình đổi chân PUMPW_PIN và PUMPS_PIN về PC6 và chân PC7

```
/* 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
```

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

```
-----
```

Kết nối

```
DHT      | TIVA C
```

```
-----
```

VCC(1) | 3.3V

DATA(2) | PD0

NC(3) | x

GND(4) | GND

Kết nối chân 1 của DHT với 3.3V

Chân 2 kết nối với các chân nào của TIVA C

Chân 4 nối với GND

Nối trở 10k giữa chân 1 và chân 2

```
-----
```

Sơ đồ kết nối GLCD và kết nối với TIVA

```
// Pin | Chức năng | TIVA C
```

```
//-----
```

```
// 1 | GND | GND
```

```
// 2 | RST | PE1
```

```
// 3 | SCK | PD2
```

```
// 4 | A0(C/D) | PD3
```

```
// 5 | LED K+ (3.3V) | 3.3V
```

```
// 6 | LED A (GND) | GND
```

```
// 7 | SDA | PD1
```

```
// 8 | CS | PE2
```

```
// 9 | VCC | 3.3V
```

```
// 10 | GND | GND
```

Sử dụng 2 nút nhấn có sẵn trên TIVA (PF4 và PF0)

Sử dụng 2 chân PC6 và PC7 nối với Modul 2 Relay

Lưu ý GND Relay nối với GND của TIVA C

```
*/
```

```
#include "DHT.h"
```

```
#include "homephone.h"
```

```
#define DHTPIN PD_0 // Chân DATA nối với PD0
```


0xFF, 0xF3, 0x87, 0x87, 0xC7,
0xC6, 0xEE, 0xFC, 0xFC, 0xFE, 0xFF, 0x3F, 0x18, 0x19, 0x3B, 0x37,
0x3E, 0xBC, 0xF8, 0xF8, 0xF8,
0xF0, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0xF0, 0xFF, 0xBF, 0xC3, 0xE7, 0x7E, 0x7C, 0x7C, 0x7C, 0xFC, 0xCE,
0x87, 0xFF, 0xFF, 0xFD, 0x9E,
0x87, 0xF9, 0xFF, 0x1F, 0x80, 0xC0, 0xE0, 0x70, 0xF8, 0xFE, 0x7F,
0x1F, 0x07, 0x01, 0x07, 0x3F,
0xFC, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFF, 0xFF, 0xFF, 0xDE, 0xC0,
0xC0, 0xC0, 0xC0, 0xDF, 0xFF, 0xFF, 0xFF, 0x00, 0xC0, 0xE0, 0xF0,
0x70, 0x38, 0x38, 0x38, 0x78,
0xF0, 0xE0, 0xC0, 0x00, 0xE0, 0xF0, 0xF0, 0x78, 0x38, 0x38, 0x78,
0x70, 0x70, 0x00, 0x00, 0x80,
0xE0, 0xF8, 0xFE, 0x3F, 0x3F, 0xFF, 0xFC, 0xE0, 0x80, 0x00, 0x00,
0x00, 0xFF, 0xFF, 0xFF, 0x87,
0x87, 0x87, 0xCF, 0xCE, 0xFE, 0xFC, 0x78, 0x00, 0x00, 0xFF, 0xFF,
0xFF, 0x7F, 0xFC, 0xF0, 0xC0,
0x80, 0xE0, 0xF8, 0xFE, 0x3F, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x1F,
0xFF, 0xF1, 0x81, 0x00, 0x80, 0xC0, 0xF0, 0x78, 0x1E, 0x8F, 0xC3,
0xC7, 0xEE, 0x7F, 0xFF, 0xFF,
0xFF, 0x3E, 0x37, 0xF3, 0xF1, 0x3C, 0x3F, 0x37, 0x31, 0x30, 0x30,
0x30, 0x30, 0x30, 0x30, 0x31,
0x3F, 0x3F, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F,
0x3F, 0x3F, 0x3D, 0x01, 0x01,
0x01, 0x01, 0x3D, 0x3F, 0x3F, 0x3F, 0x00, 0x07, 0x0F, 0x1F, 0x3C,
0x38, 0x38, 0x38, 0x3C, 0x1F,
0x1F, 0x07, 0x00, 0x0F, 0x1F, 0x1F, 0x38, 0x38, 0x38, 0x38, 0x1C,
0x1C, 0x30, 0x3E, 0x3F, 0x1F,
0x07, 0x07, 0x07, 0x07, 0x07, 0x07, 0x1F, 0x3F, 0x3E, 0x30, 0x00,
0x3F, 0x3F, 0x3F, 0x03, 0x03,
0x07, 0x0F, 0x1F, 0x3F, 0x38, 0x30, 0x00, 0x00, 0x3F, 0x3F, 0x3F,
0x00, 0x01, 0x07, 0x1F, 0x1F,
0x0F, 0x03, 0x00, 0x00, 0x3F, 0x3F, 0x3F, 0x00, 0x00, 0x00, 0x00,


```
};  
// Biện lưu các giá trị cảm biến  
int humDHT;  
int tempDHT;  
int lumen;  
int soilMoist;  
// Biện lưu trạng thái bơm  
boolean pumpWaterStatus = 0;  
boolean pumpPesStatus = 0;  
// Biện cho timer  
long sampleTimingSeconds = 50; // ==> Thời gian đọc cảm biến (s)  
long startTiming = 0;  
long elapsedTime = 0;  
// Khai tạo cảm biến  
DHT dht(DHTPIN, DHTTYPE);  
// Khai tạo LCD  
homephone lcd (PD_1, PD_2, PD_3, PE_1, PE_2);  
void setup() {  
    pinMode(PUMPW_PIN, OUTPUT);  
    pinMode(PUMPS_PIN, OUTPUT);  
    pinMode(PUMPW_ON, INPUT_PULLUP);  
    pinMode(PUMPS_ON, INPUT_PULLUP);  
    // Khai tạo cổng serial baud 115200  
    Serial.begin(115200);  
    Serial.println("HocARM NDTR Bot!");  
    // Bắt đầu đọc dữ liệu  
    dht.begin();  
    lcd.begin(); //Khai tạo LCD  
    lcd.setContrast(0x0A); //Đặt độ tương phản  
    lcd.clear(); //Xóa màn hình  
    lcd.drawBitmap(0, 0, 127, 63, ha_lg, black); // Hiện thị hình bitmap  
    lcd.display();  
    readSensors(); // Khai tạo đọc cảm biến  
    startTiming = millis(); // Bắt đầu đếm thời gian  
    lcd.clear();  
}  
  
void loop() {  
    // Khai tạo timer  
    elapsedTime = millis() - startTiming;  
    readLocalCmd();  
    showDataLCD();  
    if (elapsedTime > (sampleTimingSeconds * 1000))  
    {  
        readSensors();  
        printData();  
    }  
}
```

```
    startTiming = millis();
  }
}
int getLumen(int anaPin)
{
  int anaValue = 0;
  for (int i = 0; i < 10; i++) // Đọc giá trị cảm biến 10 lần và lấy
  giá trị trung bình
  {
    anaValue += analogRead(anaPin);
    delay(50);
  }

  anaValue = anaValue / 10;
  anaValue = map(anaValue, 4095, 0, 0, 100); //Tỉ lệ:0 ==> Sáng 100%

  return anaValue;
}
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, 4095, 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 showDataLCD(void)
{
  lcd.setTextColor(black, white);
  lcd.setCursor(0, 0); //đặt con trỏ vào tọa độ (0,0)
  lcd.print("***-HOCARM NDTR Bot-***");
  lcd.setCursor (0, 10);
```

```

    lcd.print("-----");
    lcd.setCursor (0, 20);
    lcd.print("Nhdo: ");
    lcd.print(tempDHT);
    lcd.print("oC Doam: ");
    lcd.print(humDHT);
    lcd.print("% ");
    lcd.setCursor (0, 30);
    lcd.print("Asag: ");
    lcd.print(lumen);
    lcd.print("%");
    lcd.print(" Amdat: ");
    lcd.print(soilMoist);
    lcd.print("%");
    lcd.setCursor (0, 40);
    lcd.print("BomN: ");
    lcd.print(pumpWaterStatus);
    lcd.print("      BomS: ");
    lcd.print(pumpPesStatus);
    lcd.display();
    lcd.setCursor (0, 0);
    lcd.display();
}
void printData(void)
{
    // IN thông tin ra màn hình
    Serial.print("Do am: ");
    Serial.print(humDHT);
    Serial.print(" %t");
    Serial.print("Nhiet do: ");
    Serial.print(tempDHT);
    Serial.print(" *Ct");
    Serial.print("Anh sang: ");
    Serial.print(lumen);
    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 "CAO"):
    *****/
void readLocalCmd()
{
    int digiValue = debounce(PUMPW_ON);

```

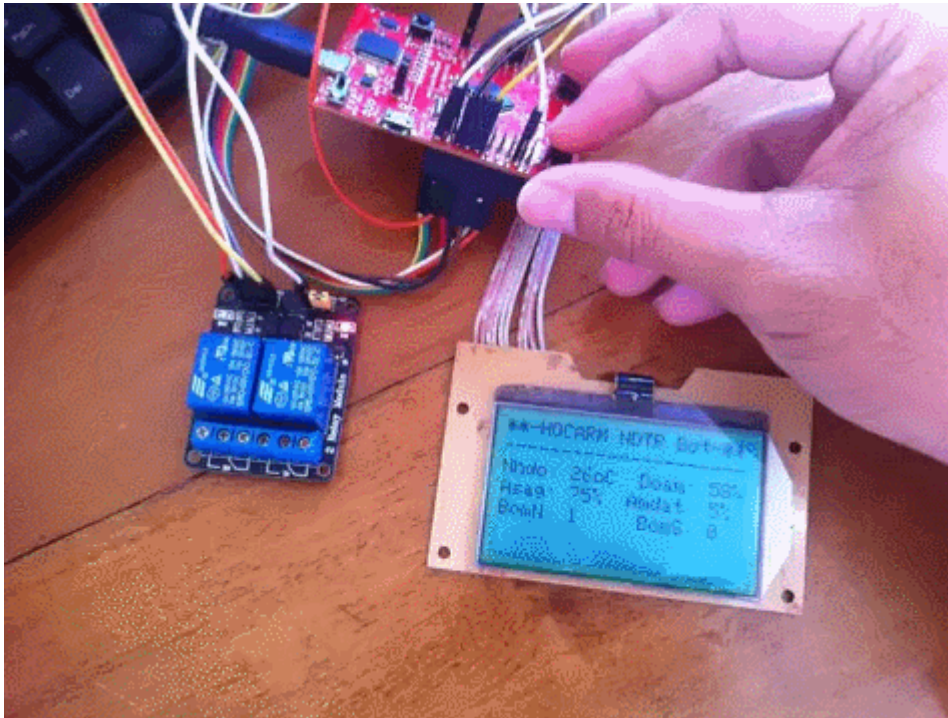
```
if (!digiValue)
{
    pumpWaterStatus = !pumpWaterStatus;
    showDataLCD();
    aplyCmd();
}
digiValue = debounce(PUMPS_ON);
if (!digiValue)
{
    pumpPesStatus = !pumpPesStatus;
    showDataLCD();
    aplyCmd();
}
}
/*****
Thực hiện đi□u khi□n các bơm
*****/
void aplyCmd()
{
    if (pumpWaterStatus == 1) digitalWrite(PUMPW_PIN, LOW);
    if (pumpWaterStatus == 0) digitalWrite(PUMPW_PIN, HIGH);

    if (pumpPesStatus == 1) digitalWrite(PUMPS_PIN, LOW);
    if (pumpPesStatus == 0) digitalWrite(PUMPS_PIN, HIGH);
}
/*****
Hàm ki□m tra trạng thái phím b□m
*****/
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;
}
```

}

Kết quả



Hệ thống chạy offline tự động

Khi đã đọc được dữ liệu cảm biến và điều khiển tủ công bơm được rồi thì một yêu cầu mới lại đặt ra là tại sao lại phải bấm nút điều khiển bơm bằng tay nhỉ ? Không lẽ lại ngồi soi từng thông số, kiểm tra các thông số rồi lại ngồi ấn nút để bơm thì đúng là quá vất vả, thế thì chắc vút luôn cái này đi cho rồi.

Đằng nào cũng đã cố tới đây rồi thì ráng làm cho con Bot này thông minh hơn một chút, ta tận dụng thông tin dữ liệu thấy để bắt hệ thống tự tưới tiêu khi đất khô quá, khi nóng quá, tự tưới nước hoặc bón phân vào buổi sáng chẳng hạn,.. Dưới đây là một số thông số cơ bản để có thể dựa vào (thực tế thì thông số này có thể thay đổi)

Cảm biến độ ẩm đất

- Ẩm ướt : trên 60%
- Ổn định: 40% - 60%
- Khô: Dưới 40% (cần phải tưới nước)

Nhiệt độ

- Lạnh: Dưới 15oC
- Trung bình: 20oC - 25oC
- Nóng: 25oC trở lên

Ánh sáng

- Tối: dưới 40% (không bơm nước tưới)
- Sáng: trên 40%

Lưu ý:

Các thông số mình đưa ra dựa trên quan sát môi trường thực tế của mình, chưa cần sử dụng thông số về nhiệt độ. Nếu cây trồng của bạn yêu cầu đòi hỏi phát triển cần có nhiệt độ thì ko nên bỏ qua cái này.

Muốn thông số điều khiển chính xác thì cần thử trong thời gian dài để lấy ra được giá trị chính xác

Có các thông số rồi mình sẽ lưu nó vào các biến trong chương trình của mình như sau

Với cảm biến DHT

- int tempDHT;
- int HOT_TEMP = 25;
- int COLD_TEMP = 15;

Với cảm biến ánh sáng

- int lumen;
- int DARK_LIGHT = 40;

Với cảm biến độ ẩm đất

- int soilMoist;
- int DRY_SOIL = 40;
- int WET_SOIL = 60;

Dựa trên các định nghĩa trên thì sẽ có ban đầu sẽ có một số cách điều khiển đơn giản dựa trên độ ẩm đất và ánh sáng (phần nhiệt độ cứ để đó dự phòng)

- Nếu khô và trời sáng -> bơm nước
- Nếu khô và trời tối -> không bơm nước

Mình viết thêm một hàm để thực hiện nhiệm vụ tưới tiêu tự động là `autoControlPlantation()` hàm này có nhiệm vụ là điều khiển bơm tưới nước dựa trên độ ẩm và ánh sáng

Ở trong chương trình có hàm `turnPumpOn()` có tác dụng bật tắt bơm, ở đây mình điều khiển bật tắt theo thời gian định sẵn, nghĩa là nếu đất khô quá thì mình sẽ bơm nước trong 1 phút rồi tắt (sau 1ph bơm nước nếu độ ẩm vẫn chưa đủ thì tiếp tục bơm), thời gian bơm này có thể thay đổi bằng cách đổi thông số `timePumpOn`, đồng thời mình sẽ cho hiển thị thông tin này lên LCD luôn.

Gom hết lại toàn bộ tiếp tục thử xem kết quả thế nào

```
/* 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
  Thêm chức năng điều khiển tự động bơm khi thiếu nước
  HocARM NDTR BOT by hocARM.org
```

Kết nối

DHT | TIVA C

VCC(1) | 3.3V

DATA(2) | PD0

NC(3) | x

GND(4) | GND

Kết nối chân 1 của DHT với 3.3V

Chân 2 kết nối với các chân nào của TIVA C

Chân 4 nối với GND

Nối trở 10k giữa chân 1 và chân 2

Sơ đồ kết nối GLCD và kết nối với TIVA

// Pin | Chức năng | TIVA C

//-----

// 1 | GND | GND

// 2 | RST | PE1

// 3 | SCK | PD2

// 4 | A0(C/D) | PD3

// 5 | LED K+ (3.3V) | 3.3V

// 6 | LED A (GND) | GND

// 7 | SDA | PD1

// 8 | CS | PE2

// 9 | VCC | 3.3V

// 10 | GND | GND

Sử dụng 2 nút nhấn có sẵn trên TIVA (PF4 và PF0)

Sử dụng 2 chân PC6 và PC7 nối với Modul 2 Relay

Lưu ý GND Relay nối với GND của TIVA C

*/

```
#include "DHT.h"
```

```
#include "homephone.h"
```

```
#define DHTPIN PD_0 // Chân DATA nối với PD0
```

```
#define LDR_PIN PE_3 // Chân PE3 nối với chân OUT cảm biến as
```

```
#define SOIL_MOIST_1_PIN PE_4 // Chân PE4 nối với cảm biến độ ẩm
```

```
// Relay, nút nhấn
```


0xF0, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0xF0, 0xFF, 0xBF, 0xC3, 0xE7, 0x7E, 0x7C, 0x7C, 0x7C, 0xFC, 0xCE,
0x87, 0xFF, 0xFF, 0xFD, 0x9E,
0x87, 0xF9, 0xFF, 0x1F, 0x80, 0xC0, 0xE0, 0x70, 0xF8, 0xFE, 0x7F,
0x1F, 0x07, 0x01, 0x07, 0x3F,
0xFC, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFF, 0xFF, 0xFF, 0xDE, 0xC0,
0xC0, 0xC0, 0xC0, 0xDF, 0xFF, 0xFF, 0xFF, 0x00, 0xC0, 0xE0, 0xF0,
0x70, 0x38, 0x38, 0x38, 0x78,
0xF0, 0xE0, 0xC0, 0x00, 0xE0, 0xF0, 0xF0, 0x78, 0x38, 0x38, 0x78,
0x70, 0x70, 0x00, 0x00, 0x80,
0xE0, 0xF8, 0xFE, 0x3F, 0x3F, 0xFF, 0xFC, 0xE0, 0x80, 0x00, 0x00,
0x00, 0xFF, 0xFF, 0xFF, 0x87,
0x87, 0x87, 0xCF, 0xCE, 0xFE, 0xFC, 0x78, 0x00, 0x00, 0xFF, 0xFF,
0xFF, 0x7F, 0xFC, 0xF0, 0xC0,
0x80, 0xE0, 0xF8, 0xFE, 0x3F, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x1F,
0xFF, 0xF1, 0x81, 0x00, 0x80, 0xC0, 0xF0, 0x78, 0x1E, 0x8F, 0xC3,
0xC7, 0xEE, 0x7F, 0xFF, 0xFF,
0xFF, 0x3E, 0x37, 0xF3, 0xF1, 0x3C, 0x3F, 0x37, 0x31, 0x30, 0x30,
0x30, 0x30, 0x30, 0x30, 0x31,
0x3F, 0x3F, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F,
0x3F, 0x3F, 0x3D, 0x01, 0x01,
0x01, 0x01, 0x3D, 0x3F, 0x3F, 0x3F, 0x00, 0x07, 0x0F, 0x1F, 0x3C,
0x38, 0x38, 0x38, 0x3C, 0x1F,
0x1F, 0x07, 0x00, 0x0F, 0x1F, 0x1F, 0x38, 0x38, 0x38, 0x38, 0x1C,
0x1C, 0x30, 0x3E, 0x3F, 0x1F,
0x07, 0x07, 0x07, 0x07, 0x07, 0x07, 0x1F, 0x3F, 0x3E, 0x30, 0x00,
0x3F, 0x3F, 0x3F, 0x03, 0x03,
0x07, 0x0F, 0x1F, 0x3F, 0x38, 0x30, 0x00, 0x00, 0x3F, 0x3F, 0x3F,
0x00, 0x01, 0x07, 0x1F, 0x1F,
0x0F, 0x03, 0x00, 0x00, 0x3F, 0x3F, 0x3F, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x1D,
0x3F, 0x3F, 0x3E, 0x3F, 0x3F, 0x36, 0x37, 0x33, 0x31, 0x71, 0x70,
0x60, 0x60, 0xE1, 0xFF, 0xFF,


```
int tempDHT;

int lumen;
int DARK_LIGHT = 40;

int soilMoist;
int DRY_SOIL = 40;
int WET_SOIL = 60;

// Biến lưu trạng thái bơm
boolean pumpWaterStatus = 0;
boolean pumpPesStatus = 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);
// Khởi tạo LCD
homephone lcd (PD_1, PD_2, PD_3, PE_1, PE_2);
void setup() {
    pinMode(PUMPW_PIN, OUTPUT);
    pinMode(PUMPS_PIN, OUTPUT);
    pinMode(PUMPW_ON, INPUT_PULLUP); // Button
    pinMode(PUMPS_ON, INPUT_PULLUP); // Button
    // Khởi tạo cổng serial baud 115200
    Serial.begin(115200);
    Serial.println("HocARM NDTR Bot!");
    // Bật đầu đọc dữ liệu
    dht.begin();
    lcd.begin(); //Khởi tạo LCD
    lcd.setContrast(0x0A); //Đặt độ tương phản
    lcd.clear(); //Xóa màn hình
    lcd.drawBitmap(0, 0, 127, 63, ha_lg, black); // Hiện thị hình bitmap
    lcd.display();
    readSensors(); // Khởi tạo đọc cảm biến
    startTiming = millis(); // Bật đầu đếm thời gian
    lcd.clear();
}

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

```
if (elapsedTime > (sampleTimingSeconds * 1000))
{
    readSensors();
    printData();
    autoControlPlantation();
    startTiming = millis();
}
}
int getLumen(int anaPin)
{
    int anaValue = 0;
    for (int i = 0; i < 10; i++) // Đọc giá trị cảm biến 10 lần và lấy
    giá trị trung bình
    {
        anaValue += analogRead(anaPin);
        delay(50);
    }

    anaValue = anaValue / 10;
    anaValue = map(anaValue, 4095, 0, 0, 100); //T=0 ==> Sáng 100%

    return anaValue;
}
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, 4095, 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 showDataLCD(void)
```

```
{
  lcd.setTextColor(black, white);
  lcd.setCursor(0, 0);    //đặt con trỏ vào tọa độ (0,0)
  lcd.print("**-HOCARM NDTR Bot-**");
  lcd.setCursor (0, 10);
  lcd.print("-----");
  lcd.setCursor (0, 20);
  lcd.print("Nhdo: ");
  lcd.print(tempDHT);
  lcd.print("oC Doam: ");
  lcd.print(humDHT);
  lcd.print("% ");
  lcd.setCursor (0, 30);
  lcd.print("Asag: ");
  lcd.print(lumen);
  lcd.print("%");
  lcd.print(" Amdat: ");
  lcd.print(soilMoist);
  lcd.print("%");
  lcd.setCursor (0, 40);
  lcd.print("BomN: ");
  lcd.print(pumpWaterStatus);
  lcd.print("      BomS: ");
  lcd.print(pumpPesStatus);
  lcd.display();
  lcd.setCursor (0, 0);
  lcd.display();
}
void printData(void)
{
  // IN thông tin ra màn hình
  Serial.print("Do am: ");
  Serial.print(humDHT);
  Serial.print(" %t");
  Serial.print("Nhiet do: ");
  Serial.print(tempDHT);
  Serial.print(" *Ct");
  Serial.print("Anh sang: ");
  Serial.print(lumen);
  Serial.print(" %t");
  Serial.print("Do am dat: ");
  Serial.print(soilMoist);
  Serial.println(" %");
}
/*****
Hàm bật bơm nước
```

```
*****/
void turnPumpOn()
{
    digitalWrite(PUMPW_PIN, LOW);
    pumpWaterStatus = 1;
    showDataLCD();
    delay (timePumpOn * 1000);
    digitalWrite(PUMPW_PIN, HIGH);
    pumpWaterStatus = 0;
    showDataLCD();
}
/*****
    Hàm đọc trạng thái bơm và kiểm tra nút nhàn
    (Nút nhàn mặc định là mức "CAO"):
*****/
void readLocalCmd()
{
    int digiValue = debounce(PUMPW_ON);
    if (!digiValue)
    {
        pumpWaterStatus = !pumpWaterStatus;
        showDataLCD();
        aplyCmd();
    }
    digiValue = debounce(PUMPS_ON);
    if (!digiValue)
    {
        pumpPesStatus = !pumpPesStatus;
        showDataLCD();
        aplyCmd();
    }
}
/*****
    Thực hiện điểu khiển các bơm
*****/
void aplyCmd()
{
    if (pumpWaterStatus == 1) digitalWrite(PUMPW_PIN, LOW);
    if (pumpWaterStatus == 0) digitalWrite(PUMPW_PIN, HIGH);

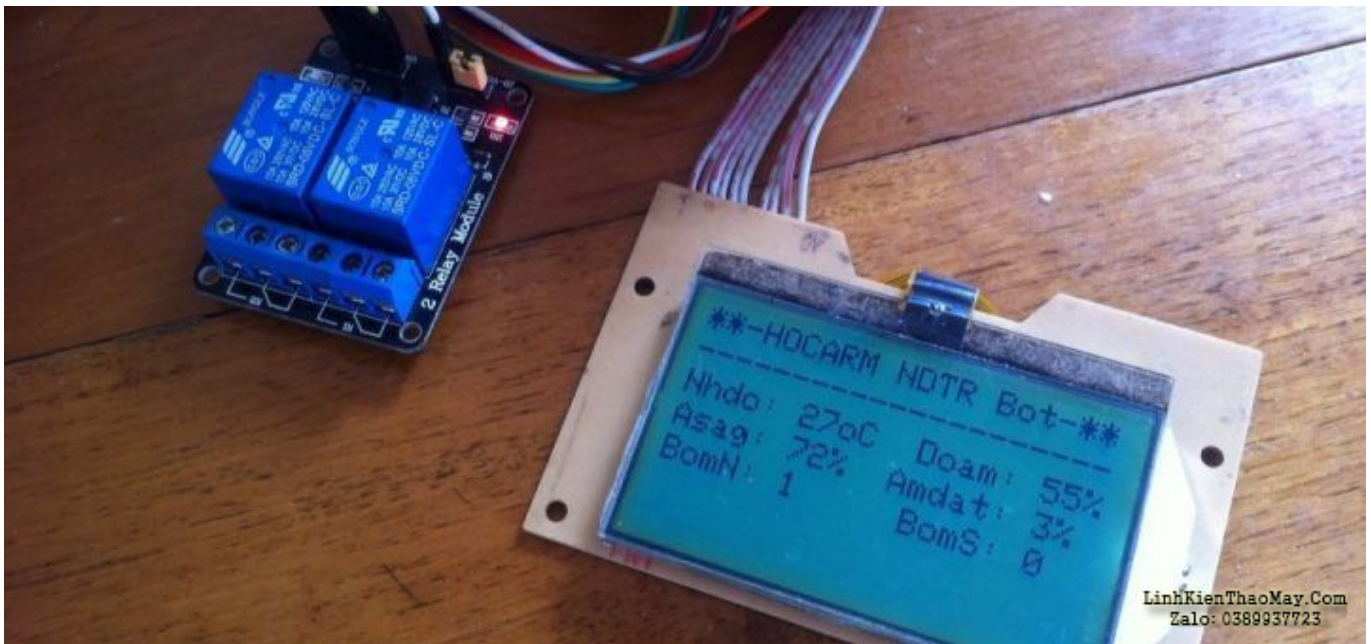
    if (pumpPesStatus == 1) digitalWrite(PUMPS_PIN, LOW);
    if (pumpPesStatus == 0) digitalWrite(PUMPS_PIN, HIGH);
}
/*****
    Hàm kiểm tra trạng thái phím bơm
*****/
```

```
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;
}
/*****
* Chức năng tự động tưới tiêu
*****/
void autoControlPlantation()
{
    //----- BƠM NƯỚC -----//
    if (soilMoist < DRY_SOIL && lumen > DARK_LIGHT)
    {
        turnPumpOn();
    }
}
```

Kết quả

Hiện tại hên là đất của mình quá thiếu nước nên sau 50s kiểm tra thì hệ thống lại tự bật relay để tưới nước, bạn có thể nhìn qua LED báo hiệu trên relay ở hình



Gửi dữ liệu lên cloud

Phù, thế là xong phần offline, ta đã có được một hệ thống có thể nắm bắt được tình hình môi trường, đất đai thông qua các thông số và cho các bác nông dân có thể biết được các số liệu có liên quan về nhiệt độ độ ẩm, ngoài ra còn có thể tự điều khiển tưới tiêu đáp ứng được một số yêu cầu cơ bản rồi.

Thế nhưng vừa xử xong được cái này thì lại đề ra một vấn đề mới, đó là bác nông dân đi uống cà phê sáng với bạn bè, muốn kiểm tra xem tình trạng cây trồng ở nhà như thế nào, tranh thủ khoe luôn với anh em xem hệ thống vườn sau siêu sạch của mình, nhiều lúc bác cũng muốn tác động điều khiển bơm nước và bơm thuốc trừ sâu mà ở xa quá, chạy về thì mất bao nhiêu công. Ước gì giải quyết được vấn đề này thì bác an nhàn đi khắp mọi miền đất nước mà không phải lo lắng gì cả.

Vậy giải pháp ở đây là gì ? Có thể cho con BOT này kết nối với internet, xong đẩy dữ liệu lên đâu đó, kiểu như 1 trang web giống giống facebook, rồi chỉ với điện thoại/ laptop ta có thể cập nhật được mọi thông tin, nó khác facebook ta cập nhật tin tức từ bạn bè, còn trang này thì cập nhật tin tức về tình hình cây trồng ở nhà. May quá có anh thingspeak đây rồi.

Nói về thingspeak một chút thì đây là một server chuyên dùng để lưu dữ liệu, được nhất là nó miễn phí, biểu thị data theo thời gian cũng tốt, và dễ dùng. Nếu bạn có thắc mắc là mình tự build một server được không ? thì câu trả lời là hoàn toàn có thể, nếu bạn làm hệ thống lớn hơn và cần server riêng thì có thể tự build một cái, lưu ý là chi phí nuôi server cùng các vấn đề liên quan cũng khá tốn kém nhé :D. Cái của mình làm thì nhỏ và tận dụng những gì có sẵn thì thingspeak đã phù hợp rồi ?

Tạm thời nêu chút vấn đề tại sao lại cho lên cloud, quá trình này được mọi người hay râm rần trên mạng là IOT, thuật ngữ gây tranh cãi và làm tốn bao nhiêu status cũng như tranh cãi của mọi người trên các diễn đàn.

Theo mình thì hiểu đơn giản IOT = Internet of Things, hiểu luôn tiếng việt là mọi vật kết nối

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

internet, vậy thì như con BOT của mình mà kết nối được internet thì chắc được gọi là IOT rồi, hehe

Thôi tạm dừng chém, giờ làm thế nào để con BOT có thể kết nối và gửi dữ liệu lên thingspeak ? Mình có viết một bài về cách gửi dữ liệu thingspeak các bạn có thể tham khảo để kiểm tra [Thingspeak với ESP8266 và TIVA C](#)

Giờ thì mình gom mấy hàm đã viết vào chương trình chính để gửi toàn bộ dữ liệu lên thingspeak luôn, có một số lệnh mình gom lại thành hàm để thuận tiện sử dụng lại, gồm có 4 hàm để thực hiện quá trình cập nhật dữ liệu lên thingspeak là

- startThingSpeakCmd() để thiết lập kết nối với server thingspeak
- sendThingSpeakCmd() để gửi độ dài data
- updateDataThingSpeak() để gửi thông số về cảm biến, trạng thái
- updateCmdThingSpeak() để cập nhật trạng thái bơm

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 KIẾN 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

Hàm viết đã xong, giờ gom tất cả vào một, thêm chức năng gửi dữ liệu lên thingspeak nữa là kết thúc thôi :D. Trước khi gom thì cần thêm hàm updateCmdThingSpeak() vào trong hàm turnPumpOn() và hàm readLocalCmd(), để mỗi khi có sự thay đổi trạng thái bơm thì cập nhật ngay lên thingspeak.

Thời gian bật bơm nước, lấy dữ liệu mình tăng lên 1 ít, để cho nó giống với thực tế chút.

```
Lưu ý cần thay một số thông số sau cho phù hợp với wifi và thingspeak của bạn
sendATcommand("AT+CWJAP="ten_wifi","mat_khau","OK", 3000);
//Thay ten_wifi và mat_khau của bạn
char msg[] = "GET /update?key=xxxxxxxx"; //Thay xxxxxxxx
bằng key của bạn và mục API Write key
```

```

/* 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
Thêm chức năng điều khiển tự động bơm khi thiếu nước
Thêm chức năng cập nhật dữ liệu qua cloud thingspeak
HocARM NDTR BOT by hocARM.org

```

Kết nối

DHT | TIVA C

VCC(1) | 3.3V

DATA(2) | PD0

NC(3) | x

GND(4) | GND

Nối trở 10k giữa chân 1 và chân 2

Sơ đồ kết nối GLCD và kết nối với TIVA

// Pin | Chức năng | TIVA C

//-----

// 1 | GND | GND

// 2 | RST | PE1

// 3 | SCK | PD2

// 4 | A0(C/D) | PD3

// 5 | LED K+ (3.3V) | 3.3V

// 6 | LED A (GND) | GND

// 7 | SDA | PD1

// 8 | CS | PE2

// 9 | VCC | 3.3V

// 10 | GND | GND

Sử dụng 2 nút nhấn có sẵn trên TIVA (PF4 và PF0)

Sử dụng 2 chân PC6 và PC7 nối với Modul 2 Relay

Lưu ý GND Relay nối với GND của TIVA C

Kết nối ESP8266 TIVA

ESP8266 | TIVA | Nguồn ngoài

VCC | x | 3.3v

GND | GND | GND

CH_PD | x | 3.3v

RX | PB1 | x

TX | PB0 | x

*/

#include "DHT.h"

#include "homephone.h"

#define DHTPIN PD_0 // Chân DATA nối với PD0

0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0xC0, 0xC0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x07, 0x1F,
0xFF, 0xF3, 0x87, 0x87, 0xC7,
0xC6, 0xEE, 0xFC, 0xFC, 0xFE, 0xFF, 0x3F, 0x18, 0x19, 0x3B, 0x37,
0x3E, 0xBC, 0xF8, 0xF8, 0xF8,
0xF0, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
0xF0, 0xFF, 0xBF, 0xC3, 0xE7, 0x7E, 0x7C, 0x7C, 0x7C, 0xFC, 0xCE,
0x87, 0xFF, 0xFF, 0xFD, 0x9E,
0x87, 0xF9, 0xFF, 0x1F, 0x80, 0xC0, 0xE0, 0x70, 0xF8, 0xFE, 0x7F,
0x1F, 0x07, 0x01, 0x07, 0x3F,
0xFC, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFF, 0xFF, 0xFF, 0xDE, 0xC0,
0xC0, 0xC0, 0xC0, 0xDF, 0xFF, 0xFF, 0xFF, 0x00, 0xC0, 0xE0, 0xF0,
0x70, 0x38, 0x38, 0x38, 0x78,
0xF0, 0xE0, 0xC0, 0x00, 0xE0, 0xF0, 0xF0, 0x78, 0x38, 0x38, 0x78,
0x70, 0x70, 0x00, 0x00, 0x80,
0xE0, 0xF8, 0xFE, 0x3F, 0x3F, 0xFF, 0xFC, 0xE0, 0x80, 0x00, 0x00,
0x00, 0xFF, 0xFF, 0xFF, 0x87,
0x87, 0x87, 0xCF, 0xCE, 0xFE, 0xFC, 0x78, 0x00, 0x00, 0xFF, 0xFF,
0xFF, 0x7F, 0xFC, 0xF0, 0xC0,
0x80, 0xE0, 0xF8, 0xFE, 0x3F, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x1F,
0xFF, 0xF1, 0x81, 0x00, 0x80, 0xC0, 0xF0, 0x78, 0x1E, 0x8F, 0xC3,
0xC7, 0xEE, 0x7F, 0xFF, 0xFF,
0xFF, 0x3E, 0x37, 0xF3, 0xF1, 0x3C, 0x3F, 0x37, 0x31, 0x30, 0x30,
0x30, 0x30, 0x30, 0x30, 0x31,
0x3F, 0x3F, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x3F, 0x3F, 0x3D, 0x01, 0x01,
0x01, 0x01, 0x3D, 0x3F, 0x3F, 0x3F, 0x00, 0x07, 0x0F, 0x1F, 0x3C,
0x38, 0x38, 0x38, 0x3C, 0x1F,
0x1F, 0x07, 0x00, 0x0F, 0x1F, 0x1F, 0x38, 0x38, 0x38, 0x38, 0x1C,
0x1C, 0x30, 0x3E, 0x3F, 0x1F,
0x07, 0x07, 0x07, 0x07, 0x07, 0x07, 0x1F, 0x3F, 0x3E, 0x30, 0x00,


```
0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
};
// Biện lưu các giá trị cảm biến
int humDHT;
int tempDHT;

int lumen;
int DARK_LIGHT = 40;

int soilMoist;
int DRY_SOIL = 40;
int WET_SOIL = 60;

// Biện lưu trạng thái bơm
boolean pumpWaterStatus = 0;
boolean pumpPesStatus = 0;
int timePumpOn = 60; // Thời gian bật bơm nước
// Biện cho timer
long sampleTimingSeconds = 120; // ==> Thời gian đọc cảm biến (s)
long startTiming = 0;
long elapsedTime = 0;

char msg[] = "GET /update?key=xxxxxxxx"; //Thay xxxxxxxxx bằng key
của bạn ở mục API Write key
char cmd[100];
char aux_str[100];
int legth;

// Khi tạo cảm biến
DHT dht(DHTPIN, DHTTYPE);
// Khi tạo LCD
homephone lcd (PD_1, PD_2, PD_3, PE_1, PE_2);
void setup() {
    pinMode(PUMPW_PIN, OUTPUT);
    pinMode(PUMPS_PIN, OUTPUT);
    pinMode(PUMPW_ON, INPUT_PULLUP); // Nút nhấn
    pinMode(PUMPS_ON, INPUT_PULLUP); // Nút nhấn
    // Khi tạo cổng serial baud 115200
    Serial.begin(115200);
    Serial1.begin(9600);
    Serial.println("HocARM NDTR Bot!");
    // Bắt đầu đọc dữ liệu
    dht.begin();
```

```
lcd.begin(); //Khởi tạo LCD
lcd.setContrast(0x0A); //Đặt độ tương phản
lcd.clear(); //Xóa màn hình
lcd.drawBitmap(0, 0, 127, 63, ha_lg, black); // Hiện thị hình bitmap
lcd.display();
connectWiFi();
readSensors(); // Khởi tạo đọc cảm biến
startTiming = millis(); // Bật đầu đếm thời gian
lcd.clear();
}

void loop() {
// Khởi tạo timer
elapsedTime = millis() - startTiming;
readLocalCmd();
showDataLCD();
if (elapsedTime > (sampleTimingSeconds * 1000))
{
readSensors();
printData();
autoControlPlantation();
updateDataThingSpeak();
startTiming = millis();
}
}
int getLumen(int anaPin)
{
int anaValue = 0;
for (int i = 0; i < 10; i++) // Đọc giá trị cảm biến 10 lần và lấy
giá trị trung bình
{
anaValue += analogRead(anaPin);
delay(50);
}

anaValue = anaValue / 10;
anaValue = map(anaValue, 4095, 0, 0, 100); //Tỉ lệ:0 ==> Sáng 100%

return anaValue;
}
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, 4095, 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 showDataLCD(void)
{
    lcd.setTextColor(black, white);
    lcd.setCursor(0, 0); //đặt con trỏ vào tọa độ (0,0)
    lcd.print("***-HOCARM NDTR Bot-***");
    lcd.setCursor (0, 10);
    lcd.print("-----");
    lcd.setCursor (0, 20);
    lcd.print("Nhdo: ");
    lcd.print(tempDHT);
    lcd.print("oC Doam: ");
    lcd.print(humDHT);
    lcd.print("% ");
    lcd.setCursor (0, 30);
    lcd.print("Asag: ");
    lcd.print(lumen);
    lcd.print("%");
    lcd.print(" Amdat: ");
    lcd.print(soilMoist);
    lcd.print("%");
    lcd.setCursor (0, 40);
    lcd.print("BomN: ");
    lcd.print(pumpWaterStatus);
    lcd.print(" BomS: ");
    lcd.print(pumpPesStatus);
    lcd.display();
    lcd.setCursor (0, 0);
    lcd.display();
}
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 bật bơm nước
*****/
void turnPumpOn()
{
digitalWrite(PUMPW_PIN, LOW);
pumpWaterStatus = 1;
showDataLCD();
updateCmdThingSpeak();
delay (timePumpOn * 1000);
digitalWrite(PUMPW_PIN, HIGH);
pumpWaterStatus = 0;
showDataLCD();
updateCmdThingSpeak();
}
/*****
Hàm đọc trạng thái bơm và kiểm tra nút nhàn
(Nút nhàn mặc định là mức "CAO"):
*****/
void readLocalCmd()
{
int digiValue = debounce(PUMPW_ON);
if (!digiValue)
{
pumpWaterStatus = !pumpWaterStatus;
showDataLCD();
aplyCmd();
updateCmdThingSpeak();
}
digiValue = debounce(PUMPS_ON);
if (!digiValue)
```

```
{
    pumpPesStatus = !pumpPesStatus;
    showDataLCD();
    aplyCmd();
    updateCmdThingSpeak();
}
}
/*****
Thực hiện đi□u khi□n các bơm
*****/
void aplyCmd()
{
    if (pumpWaterStatus == 1) digitalWrite(PUMPW_PIN, LOW);
    if (pumpWaterStatus == 0) digitalWrite(PUMPW_PIN, HIGH);

    if (pumpPesStatus == 1) digitalWrite(PUMPS_PIN, LOW);
    if (pumpPesStatus == 0) digitalWrite(PUMPS_PIN, HIGH);
}
/*****
Hàm ki□m tra trạng thái phím b□m
*****/
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;
}
/*****
Chức năng tự động tưới tiêu
*****/
void autoControlPlantation()
{
    //----- BƠM NƯỚC -----//
```

```
if (soilMoist < DRY_SOIL && lumen > DARK_LIGHT)
{
    turnPumpOn();
}
}
// Hàm gửi lệnh AT
int8_t sendATcommand(char* ATcommand, char* expected_answer, unsigned
int timeout) {

    uint8_t x = 0, answer = 0;
    char response[100];
    unsigned long previous;

    memset(response, '', 100);    // xóa buffer

    delay(100);

    while ( Serial1.available() > 0) Serial1.read();    // đọc input

    Serial1.println(ATcommand);    // Gửi lệnh AT

    x = 0;
    previous = millis();

    // Chờ phản hồi
    do {
        if (Serial1.available() != 0) {
            // Nếu có dữ liệu trong buffer UART, đọc và kiểm tra nó với
            expected_answer
            response[x] = Serial1.read();
            x++;
            // Nếu đúng thì trả kết quả answer = 1, thoát hàm
            if (strstr(response, expected_answer) != NULL)
            {
                answer = 1;
            }
        }
    } while ((answer == 0) && ((millis() - previous) < timeout)); // Nếu
    sai thì tiếp tục thử lại cho tới hết thời gian timeout
    Serial.println(response);    // In giá trị thấy để debug
    return answer;
}
// Hàm gửi lệnh AT 2 để gửi dữ liệu
int8_t sendATcommand2(char* ATcommand, char* expected_answer1,
char* expected_answer2, unsigned int timeout) {
```

```
uint8_t x = 0, answer = 0;
char response[100];
unsigned long previous;

memset(response, '', 100); // Khởi tạo lại chuỗi về 0

delay(100);

while ( Serial1.available() > 0) Serial1.read(); // Xóa buffer

Serial1.println(ATcommand); // Gửi lệnh AT

x = 0;
previous = millis();

// Chờ phản hồi
do {
  // Nếu có dữ liệu từ UART thì đọc và kiểm tra
  if (Serial1.available() != 0) {
    response[x] = Serial1.read();
    x++;
    // Kiểm tra giá trị 1 nếu thấy expected_answer1
    if (strstr(response, expected_answer1) != NULL)
    {
      answer = 1;
    }
    // Kiểm tra giá trị 2 nếu thấy expected_answer2
    else if (strstr(response, expected_answer2) != NULL)
    {
      answer = 2;
    }
  }
}
// Đợi time out
while ((answer == 0) && ((millis() - previous) < timeout));
Serial.println(response); // In giá trị thấy để debug
return answer;
}
/*****
  Kết nối wifi
*****/
void connectWiFi(void)
{
  sendATcommand("AT", "OK", 5000); //Kiểm tra kết nối
  sendATcommand("AT+CWMODE=1", "OK", 5000); //Cấu hình chế độ
station
```

```

sendATcommand("AT+CWJAP="ten_wifi","mat_khau", "OK", 5000);
sendATcommand("AT+CIPMUX=1", "OK", 5000); // Bật chế độ đa kết
nối
sendATcommand("AT+CIFSR", "OK", 5000); // Hiện thị ip
Serial.println("ESP8266 Connected");
}

/*****
Kết nối với ThingsSpeak.com
*****/
void startThingSpeakCmd(void)
{
memset(aux_str, '', 100);
sprintf(aux_str, sizeof(aux_str), "AT+CIPSTART=1,\"TCP\",\"%s\",80",
IP);
if (sendATcommand2(aux_str, "OK", "ERROR", 20000) == 1)
{
Serial.println("OK Connected Thingspeak");
}
}

/*****
Gửi data lên channel ThingsSpeak.com
*****/
void sendThingSpeakCmd(void)
{
memset(aux_str, '', 100);
sprintf(aux_str, "AT+CIPSEND=1,%d", legth);
if (sendATcommand2(aux_str, ">", "ERROR", 20000) == 1)
{
sendATcommand2(cmd, "SEND OK", "ERROR", 10000);
}
}

/*****
Truyền tất cả dữ liệu lên thingspeak.com và đóng kết nối
*****/
void updateDataThingSpeak(void)
{
startThingSpeakCmd();
// Gửi toàn bộ thông tin cảm biến cũng như trạng thái lên thingspeak
sprintf(cmd,
"%s&field1=%d&field2=%d&field3=%d&field4=%d&field5=%d&field6=%d", msg,
tempDHT, humDHT, lumen, soilMoist, pumpWaterStatus, pumpPesStatus);
legth = strlen(cmd) + 2;
sendThingSpeakCmd();
sendATcommand("AT+CIPCLOSE=1", "OK", 5000);
}

```

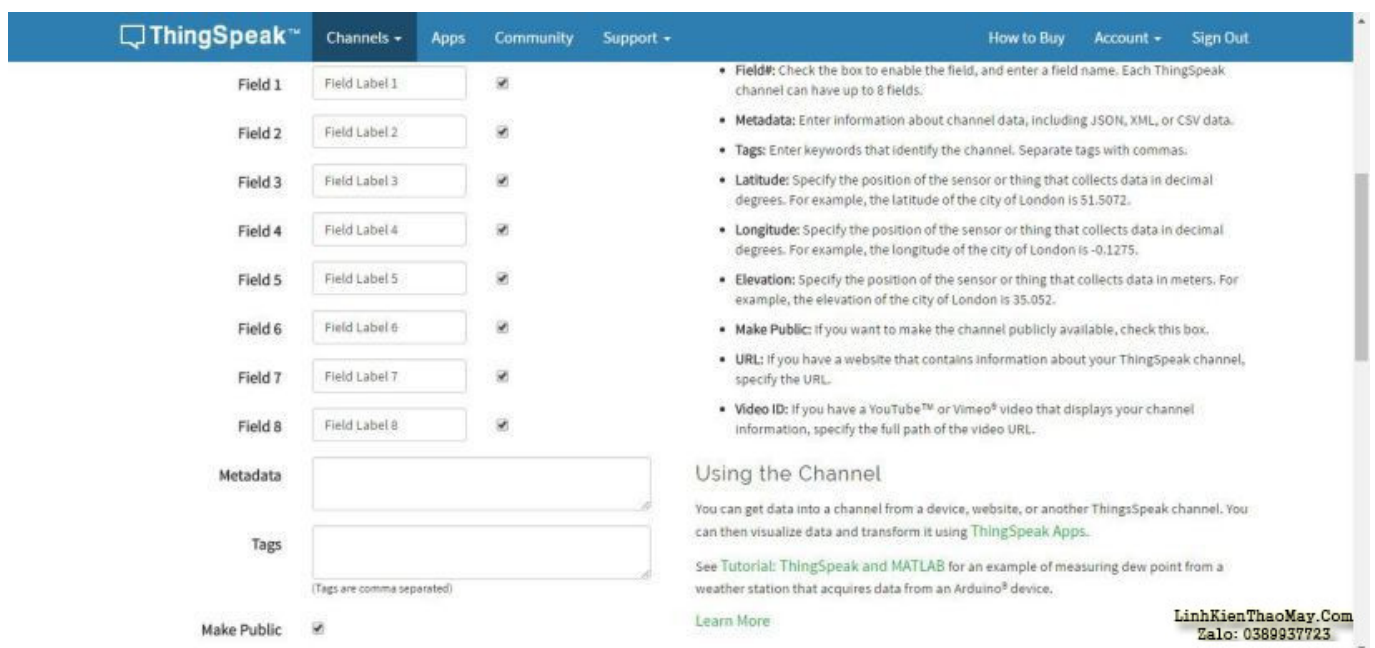
```

/*****
Cập nhật trạng thái bơm lên thingspeak.com
*****/
void updateCmdThingSpeak(void)
{
    digitalWrite(GREEN_LED, HIGH); // Bật đèn xanh báo quá trình truy
    n b
    for (int i = 0; i < 2; i++) // Thực hiện 2 l
    {
        startThingSpeakCmd ();
        // Cập nhật trạng thái bơm
        sprintf(cmd, "%s&field5=%d&field6=%d", msg, pumpWaterStatus,
        pumpPesStatus);
        legth = strlen(cmd) + 2;
        sendThingSpeakCmd();
        sendATcommand("AT+CIPCLOSE=1", "OK", 5000);
    }
    digitalWrite(GREEN_LED, LOW); // K
}

```

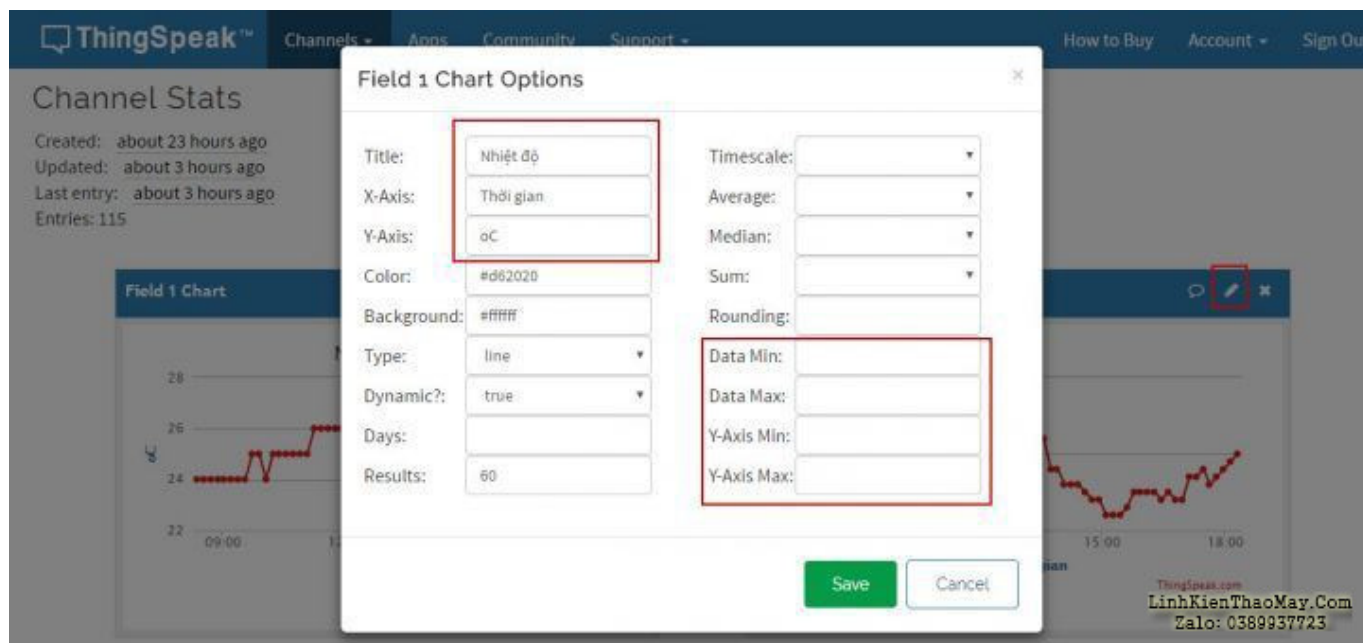
Mình quên mất, phải chỉnh chút cấu hình trên thingspeak với chỉnh sửa lại tên các trục để biết biểu đồ nào, hiển thị thông số gì nữa chứ.

Các bạn vào tài khoản thingspeak của mình, thẻ Channel setting bật hết 8 field lên, mình dùng chỉ 6 field thôi, 2 cái để dành :D, nếu muốn public channel của mình cho mọi người quan sát được thì tick chọn vào Make public là ổn.



The screenshot shows the 'Channel Settings' page on ThingSpeak. It includes a navigation bar with 'Channels', 'Apps', 'Community', and 'Support'. The main content area has a table with 8 fields, each with a label and a checked checkbox. Below the table are sections for 'Metadata', 'Tags', and 'Make Public' (checked). There is also a 'Using the Channel' section with a 'Learn More' link. The footer contains the contact information for LinhKienThaoMay.Com.

Vào lại thẻ Private View chỉnh lại mô tả về field để dễ biết đâu là biểu đồ nhiệt độ, đâu là biểu đồ độ ẩm,...



Kết quả

Kết quả có thể theo dõi được trên thingspeak sẽ có dạng như hình dưới đây, khá là trực quan và rõ ràng



Các bạn có thể xem channel mình public tại <https://thingspeak.com/channels/222622/>

Toàn bộ chương trình cho ae tham khảo tại đây nhé

hocarm/hocarmNDTRbot

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





Tạm kết

Tới đây là thở phào thoải mái rồi, viết mãi mới xong, bài hướng dẫn trên chỉ là bước khởi đầu để bạn có thể biết được từ ý tưởng tới một phiên bản prototype để đem đi thử là như thế nào, đồng thời cũng biết được thêm xiu xiu về cái IOT và cách mạng công nghiệp lần thứ 4 mà mọi người vẫn đang truyền miệng khắp nơi. Tất nhiên nó chỉ ở mức độ cơ bản, chỉ phục vụ cho nhu cầu cá nhân, vẫn còn rất nhiều vấn đề phải làm và cải tiến thêm như đem ra thử, lấy số liệu, thông số để điều khiển, giám sát qua điện thoại, ... các phiên bản với Arduino, STM32 thì chắc mình xin hẹn bài viết sau thôi ?

Nếu bạn muốn thương mại một sản phẩm gọi là công nghệ cao trong nông nghiệp thì cần phải giải quyết thêm vô vàn bài toán về những vấn đề về kinh phí, về phần cứng, về dữ liệu, bảo mật, về làm app riêng trên điện thoại, về các điều kiện thổ nhượng đất đai để cây trồng phát triển, vv và vv, thậm chí cả những vấn đề chẳng liên quan chút nào tới kỹ thuật. Cái này hơi vĩ mô chút nên thôi mình xin phép không chém ?

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 chuyên la the nay mình moi nhan dc của khách nhỏ sửa chữa cho dan am thanh kenwood chạy radio và đọc đĩa CD nhưng máy khách mang tôi trong tình trạng chấp phân công suất .DIA và RADIO vẫn hoạt động bình thường nhưng bị chấp CÔNG SUẤT nên kẹp loa vào dinh u nếu để lâu loa sẽ bị cháy .hien gio mình vẫn chưa dám làm gì cả mọi kiểm tra so bo thì thay chạy con STK4150 mình nhìn mà đã thay chuối rồi vì hàng xách tay mà lại thay con STK4150 mình chưa thay gặp con này bao giờ vì vậy nhờ các bạn góp ý và giúp mình xem trên thị trường có con này không vậy?tro giới liệu có ko các bạn nhi?ban nào đã từng làm qua ban này xin giúp đỡ mình một tay.thank các bạn nhiều.](#)
3. [Hệ thống nông nghiệp thông minh phiên bản Arduino](#)
4. [Hệ thống nông nghiệp thông minh phiên bản ESP8266](#)
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 sạch 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 sức xịt ngâm bằng nito phin e thay mới, cấp e cân xuống 170 khi nạp ga sờ đầu cũng nóng đường hồi động sương mặc dù cho ít ga ạ.](#)
9. [tủ lạnh đông tuyết. \(tủ bảo ôn\) - bục giàn. hết ga, mình đã lên 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.](#)
10. [tủ lạnh hitachi 154l nội địa - xì dàn nóng., cấp 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. [tulanh vtb quá gio vn - tu bị thung dan nong,minh đa thay dan moi,khi bom ga thay dan nong, nong nua dan con nua dan ve phin kg nong.đầu dàn lạnh có ít tuyết bám.mình đã kiểm tra 2dàn nonglạnh thay kg bị tuc.](#)
12. [tuyển 5 thợ sửa chữa laptop chuyên nghiệp . 0904677396. Mr : hưng - sửa chữa phần cứng laptop chuyên nghiệp](#)