**LỜI NÓI ĐẦU**

Ngày nay, ngành nông nghiệp là một cuộc đua. Nông dân bị thúc ép phải trồng nhiều sản phẩm hơn trong khi chất lượng đất ngày một tệ hơn, diện tích ngày một giảm và biến động thời tiết ngày một phức tạp. **IoT trong nông nghiệp** sẽ cho phép nông dân theo dõi sản phẩm và điều kiện của họ trong thời gian thực. Họ nhận được thông tin chi tiết nhanh, có thể dự đoán các vấn đề trước khi chúng xảy ra và đưa ra quyết định sáng suốt về cách phòng tránh chúng. Ngoài ra, các giải pháp IoT trong nông nghiệp cũng cho phép thực hiện quy trình sản xuất tự động, ví dụ, tưới tiêu, bón phân dựa trên nhu cầu và robot thu hoạch tự động.

Các hệ thống sản xuất nông nghiệp thông minh giúp tối ưu hóa việc sử dụng tài nguyên như: nước, năng lượng, đất đai. Không chỉ làm cho nông nghiệp xanh hơn, mà nó còn giúp giảm đáng kể việc sử dụng thuốc trừ sâu và phân bón. Cách tiếp cận này cho phép có được một sản phẩm cuối cùng sạch hơn và hữu cơ hơn so với các phương pháp nông nghiệp truyền thống. Bên cạnh đó sử dụng các hệ thống được kết nối, họ có thể tạo ra những điều kiện tốt nhất, thuân lợi nhất giúp tăng giá trị dinh dưỡng, chất lượng của sản phẩm.

.Trong Đồ án chuyên ngành kỹ thuật máy tính lần này nhóm em xin được trình bày: “**Hệ thống nhà vườn thông minh**” sử dụng vi điều khiển ESP8266 kết hợp với ứng dụng Blynk.

Trong quá trình thực hiện đồ án, do kiến thức còn hạn hẹp cùng với kỹ năng phân tích chưa cao nên còn nhiều thiếu sót, mong thầy cô nhiệt tình đóng góp để đồ án của chúng em thêm hoàn thiện hơn.

Nhóm thực hiện đồ án chúng em xin chân thành cảm ơn thầy Hồ Viết Việt – Chuyên ngành Kỹ thuật Máy tính – Khoa Điện Tử Viễn Thông đã nhiệt tình hướng dẫn và giúp đỡ nhóm em hoàn thành Đồ án này.

**Chúng em xin chân thành cảm ơn!**

**Nhóm thực hiện đề tài: Trần Thị Soa**

**Phạm Hoàng Bảo**

**Lê Quang Tuấn**

**MỤC LỤC**

[**LỜI NÓI ĐẦU** 1](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277451)

[**MỤC LỤC** 2](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277452)

[**CHƯƠNG 1: NGUYÊN LÝ HOẠT ĐỘNG CỦA HỆ THỐNG** 4](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277453)

[**1.1.** **Mục tiêu và sơ đồ khối** 4](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277454)

[1.1.1. Mục tiêu 4](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277455)

[1.1.2. Sơ đồ khối 4](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277456)

[**1.2.** **Nguyên lý hoạt động của hệ thống** 5](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277457)

[1.2.1. Giới thiệu tổng quan 5](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277458)

[1.2.2. Nguyên lý hoạt động của mạch 6](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277459)

[**CHƯƠNG 2: GIỚI THIỆU CÁC LINH KIỆN VÀ PHẦN MỀM TRONG HỆ THỐNG** 7](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277460)

[**2.1.** **Giới thiệu về các linh kiện trong hệ thống** 7](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277462)

[2.1.1. Vi điều khiển ESP8266 7](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277463)

[2.1.2. Module Relay 2-Channel 5v 9](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277464)

[2.1.3. Cảm biến nhiệt độ DHT11 10](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277465)

[2.1.4. Nút nhấn 13](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277466)

[2.1.5**.** Cảm biến độ ẩm đất 13](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277467)

[2.1.6 Điện trở 14](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277469)

[**2.2.** **Các phần mềm trong hệ thống** 17](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277472)

[2.2.1. Arduino IDE 17](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277473)

[2.2.2. Ứng dụng Blynk 18](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277474)

[2.2.3. Thư viện Simple-Timer 19](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277475)

[2.2.4. Thư viện DHT 20](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277476)

**2.3. Tính toán các thông số**……………………………………………………

* 1. 2.3.1. Tính toán công thức chuyển đổi từ giá trị số đọc ADC thành đại lượng vật lý cần đo nhiệt độ…………………………

[**CHƯƠNG 3: MÔ PHỎNG HỆ THỐNG** 22](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277477)

[**3.1.** **Thiết kế mạch ng**u**yên lý** 22](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277478)

[3.1.1. Thuật toán trong chương trình chính 22](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277481)

[3.1.2. Thuật toán khi dùng nút bấm 23](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277482)

[3.1.3. Thuật toán bật tắt máy bơm-đèn 24](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277483)

[3.1.4. Thuật toán hiển thị trên Blynk 25](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277484)

[**3.2.** **Chạy mô phỏng bằng Proteus** 33](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277486)

[**CHƯƠNG 4: CHẾ TẠO MẠCH THỰC TẾ** 34](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277487)

[**4.1.** **Thiết kế mạch in** 34](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277489)

[**4.2.** **Lắp đặt thiết bị và hoàn thiện hệ thống** 34](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277490)

[**4.3.** **Chạy mạch và đánh giá kết quả** 35](file:///C:\Users\ADMIN\Downloads\BAO_CAO.docx#_Toc515277491)

**CHƯƠNG 1: NGUYÊN LÝ HOẠT ĐỘNG CỦA HỆ THỐNG**

**Mục tiêu và sơ đồ khối**

* + 1. **Mục tiêu**

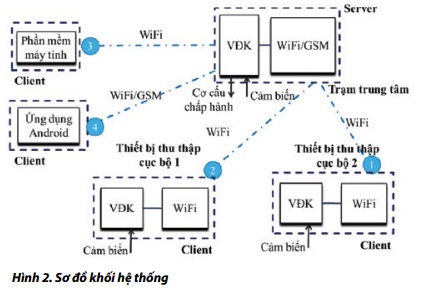
Ngày nay thế giới đang bước sang kỷ nguyên mới , kỷ nguyên của sự phát triển kỹ thuật công nghệ nói chung hay kỹ thuật điều khiển và tự động hóa nói riêng. Các hệ thống điều khiển và tự động hóa có ứng dụng rộng rãi trong các lĩnh vực thực tế như: thiết kế hệ thống đèn tín hiệu giao thông, hệ thống đồng hồ số đo được thời gian thực, các hệ thống đếm sản phẩm trong các nhà máy công nghiệp, ...

Và trong xu thế hiện nay mọi hoạt động hầu hết được tự động hóa bằng các thiết bị điện tử có kết nối Internet để người sử dụng có thể điều khiển từ xa hoặc tự động làm việc . Theo dòng phát triển công nghệ, công nghệ bán dẫn đang phát triển mạnh mẽ, thành tựu của nó là sự ra đời các hệ thống nhúng, từ khi ra đời đến nay hệ thống nhúng đã góp phần quan trọng trong các hệ thống điều khiển và tự động hóa. Bên cạnh đó, việc kết hợp IoT với các thiết bị, hệ thống nhúng được dự báo sẽ tạo ra lượng dữ liệu lớn từ đa dạng nguồn, kéo theo sự cần thiết cho việc kết tập dữ liệu nhanh, gia tăng nhu cầu đánh chỉ mục, lưu trữ, và xử lý các dữ liệu này hiệu quả hơn. Internet Vạn Vật hiện nay là một trong các nền tảng của Thành phố Thông minh, và các Hệ thống Quản lý Năng lượng Thông minh…

Với mục đích để nhằm hiểu rõ và nắm bắt công nghệ của dòng vi điều khiển, vi xử lý, khai thác các ứng dụng của nó trong cuộc sống và để thấy được sự hấp dẫn của lĩnh vực IoT nhóm chúng em xin được trình bày : “**Hệ thống nhà vườn thông minh ”.**

Với mục tiêu thực hiện như sau:

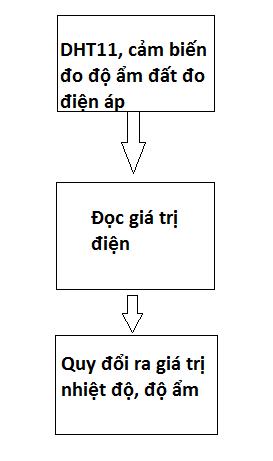
* Phần cứng: Thiết kế mô phỏng được một hệ thống có khả năng đọc nhiệt độ độ ẩm đất, hiển thị cho người dùng xem. Hệ thống hoạt động tốt ở chế độ offline và tự động.
* Phần mềm: Sử dụng thành thạo hợp ngữ để lập trình cho hệ thống có thể truyền dữ liệu lên ứng dụng Blynk để người dùng theo dõi và lập trình để hệ thống tự động làm việc khi thỏa mãn điều kiện cho trước.
  + 1. **Sơ đồ khối**



* 1. Muốn cảm biến gửi tín hiệu đầu tiên vi điều khiển sẽ gửi tín hiệu muốn đo đến DHT11 sau khi xác nhận được . Khối cảm biến sẽ nhận nhiệt độ và độ ẩm từ môi trường bên ngoài để gửi tín hiệu dạng sóng 5 byte dữ liệu và nhiệt độ đo được đến vi điều khiển . Sau đó vi điều khiển sẽ gửi tín hiệu tới Blynk, sau đó số liệu được hiển thị trên màn hình.
  2. Nguồn được dùng để cung cấp điện áp cho vi điều khiển, cảm biến

**Nguyên lý hoạt động của hệ thống**

* + 1. **Giới thiệu tổng quan**
* Khối nguồn: Ở đây ta sử dụng Adapter 5V-2A dùng để cung cấp nguồn cho vi điều khiển.
* Khối hiển thị: Đưa dữ liệu lên và hiển thị trên ứng dụng BLYNK
* Khối cảm biến:
* Ở đây ta dùng cảm biến DHT11 để đo nhiệt độ - độ ẩm không khí, cảm biến đo độ ẩm đất để đo độ ẩm. Muốn hiển thị nhiệt độ ta phải sử dụng thông qua bộ chuyển đổi ADC, DHT11 sẽ chuyển đổi giá trị nhiệt độ đo được thành giá trị điện áp qua bộ chuyển đổi ADC rồi cấp cho vi điều khiển, để đưa lên Blynk
* Ta có sơ đồ như sau:



* 1. Nguyên lý hoạt động của mạch

Khi cấp nguồn 5V cho hệ thống, cảm biến hoạt động, cung cấp giá trị điện áp cho vi điều khiển, sau đó, vi điều khiển sẽ quy đổi các giá trị điện áp đó ra giá trị nhiệt độ-độ ẩm không khí, độ ẩm đất rồi đẩy giữ liệu lên Blynk.

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.

Chúng ta sẽ định nghĩa các thông số này trong chương trình

**CHƯƠNG 2: GIỚI THIỆU VÀ TÍNH TOÁN LINH KIỆN TRONG HỆ THỐNG**

* 1. **Giới thiệu về các linh kiện trong hệ thống**
  2. **2.1.1 ESP8266 NodeMCU**

ESP8266 là một mạch vi điều khiển có thể giúp chúng ta điều khiển các thiết bị điện tử.Thêm vào đó nó được tích hợp wi-fi 2.4GHz có thể dùng cho lập trình.



*Hình 1: ESP8266 NodeMCU*

-WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n

-Điện áp hoạt động: 3.3V

-Điện áp vào: 5V thông qua cổng USB

-Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)

-Số chân Analog Input: 1 (điện áp vào tối đa 3.3V)

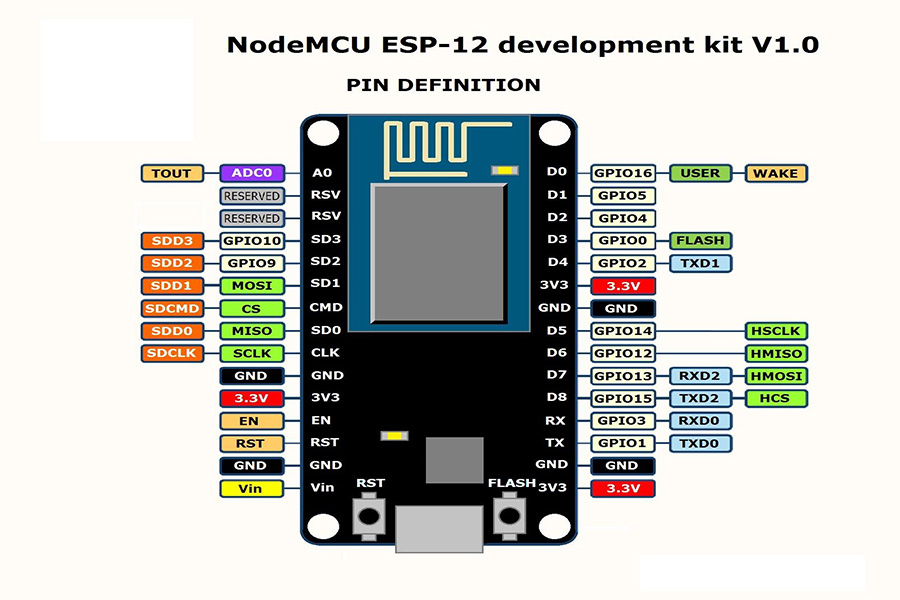
-Bộ nhớ Flash: 4MB

-Giao tiếp: Cable Micro USB ( tương đương cáp sạc điện thoại )

-Hỗ trợ bảo mật: WPA/WPA2

-Tích hợp giao thức TCP/IP

-Lập trình trên các ngôn ngữ: C/C++, Micropython,…



*Hình 2: Sơ đồ chân của NodeMCU*

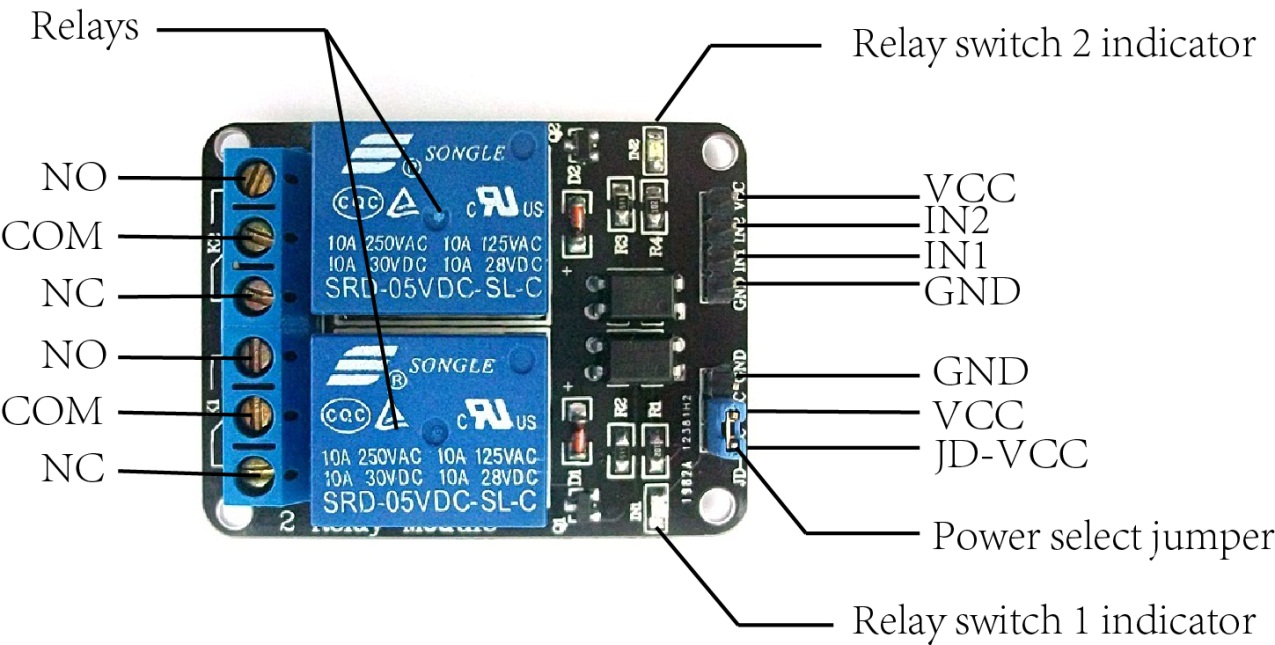
NodeMCU có tổng cộng 13 chân GPIO tuy nhiên một số chân được dùng cho những mục đích quan trọng khác:

* Tất cả các GPIO đều có trở kéo lên nguồn bên trong (ngoại trừ GPIO16 có trở kéo xuống GND). Người dùng có thể cấu hình kích hoạt hoặc không kích hoạt trở kéo này.
* GPIO1 và GPIO3: hai GPIO này được nối với TX và RX của bộ UART0, NodeMCU nạp code thông qua bộ UART này nên tránh sử dụng 2 chân GPIO này.
* GPIO0, GPIO2, GPIO15: đây là các chân có nhiệm vụ cấu hình mode cho ESP8266 điều khiển quá trình nạp code nên bên trong NodeMCU (có tên gọi là strapping pins) có các trở kéo để định sẵn mức logic cho chúng như sau: GPIO0: HIGH, GPIO2: HIGH, GPIO15: LOW. Vì vậy khi muốn sử dụng các chân này ở vai trò GPIO cần phải thiết kế một nguyên lý riêng để tránh xung đột đến quá trình nạp code.
* GPIO9, GPIO10: hai chân này được dùng để giao tiếp với External Flash của ESP8266 vì vậy cũng không thể dùng được.
* Các GPIO còn lại: GPIO 4, 5, 12, 13, 14, 16 có thể sử dụng bình thường

**2.1.2 Module Relay 2-Channel 5v**

Module Relay 2-Channel 5V được dùng nhiều trong các ứng dụng đóng ngắt các thiết bị tiêu thụ dòng điện lớn (<10A).

Module có thể đóng ngắt cùng lúc hai kênh bằng tín hiệu điều khiển ( với mức điện áp 3V3 hoặc 5V) từ các vi điều khiển khác nhau như: Arduino, 8051, AVR, PIC, DSP, ARM, ARM, MSP430, logic TTL, đồng thời module được cách ly bằng optocoupler giúp bảo vệ tốt hơn cho các vi điều khiển.



*Hình 3: Sơ đồ chân của Module Relay*

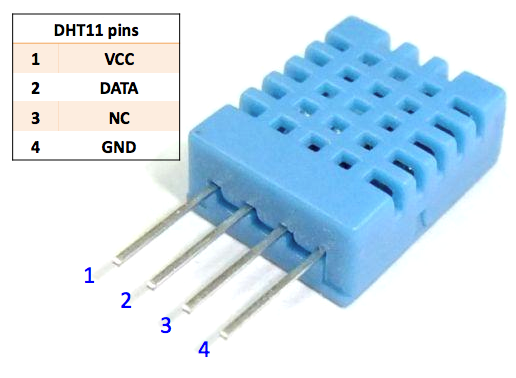
Module được kết nối với các board điều khiển bằng 4 chân header như sau:

* **VCC** cung cấp nguồn cho các opto.
* **GND** kết nối với GND của board điều khiển.
* **IN1** và **IN2** dùng để điều khiển relay 1 và relay 2, tích cực mức thấp

Ngoài ra còn một 3 chân header được dùng để cấp nguồn cho relay, header này sẽ có một jumper dùng để kết nối chân VCC với chân RY\_VCC mục đích dùng chung nguồn VCC (5V) từ header 4 chân cho relay, thông thường jumper được nối lại với nhau. Nếu như muốn cách ly tín hiệu điều khiển với nguồn cấp cho relay thì có thể bỏ jumper này ra và cấp nguồn riêng 5V cho chân RY\_VCC.

* Thông số kỹ thuật
* Đóng ngắt được dòng điện cao: AC250V 10A, DC30V 10A
* 2 led báo trạng thái relay
* Điện áp điều khiển: 5V
* Mạch cách ly bằng opto
* Kích thước: 50x45 mm
  1. **2.1.3 Cảm biến nhiệt độ DHT11**

Cảm biến độ ẩm và nhiệt độ DHT11 Temperature Humidity Sensor là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1 wire (giao tiếp digital 1 dây truyền dữ liệu duy nhất). Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không phải qua bất kỳ tính toán nào



*Hình 4. Cảm biến nhiệt độ DHT11*

### Thông số

- Điện áp hoạt động: 3 --> 5V

- Dải nhiệt độ đo: 0 -> 50°C với độ chính xác là ±2°C

- Dải độ ẩm đo: 20 -> 80% với độ chính xác là 5%

- Kích thước: 15.5mm x 12mm x 5.5mm

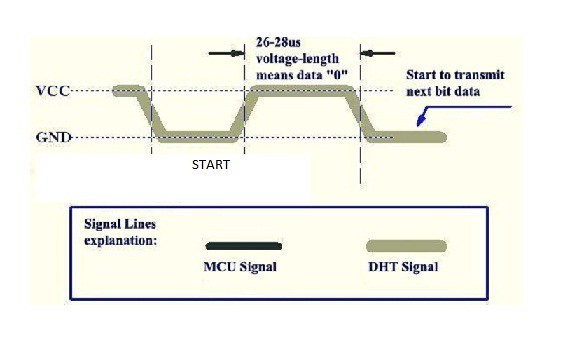
- Tần số lấy mẫu: 1Hz , nghĩa là 1 giây DHT11 lấy mẫu một lần.

- Có 4 chân, khoảng cách chân 0.1''

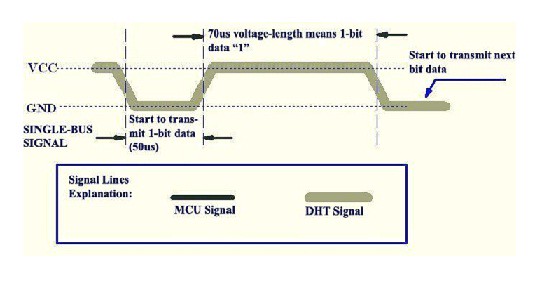
### Cách điều khiển

DHT11 gửi và nhận dữ liệu với một dây tín hiệu **DATA**, với chuẩn dữ liệu truyền 1 dây này, chúng ta phải đảm bảo sao cho ở chế độ chờ (idle) dây DATA có giá trị ở mức cao, nên trong mạch sử dụng DHT11, dây DATA phải được mắc với một trở kéo bên ngoài(thông thường giá trị là 4.7kΩ).

Dữ liệu truyền về của DHT11 gồm 40bit dữ liệu theo thứ tự: 8 bit biểu thị phần nguyên của độ ẩm + 8 bit biểu thị phần thập phân của độ ẩm + 8 bit biểu thị phần nguyên của nhiệt độ + 8 bit biểu thị phần thập phân của nhiệt độ + 8 bit check sum.



*Hình5. Dạng tín hiệu giao tiếp với cảm biến(bit 0)*



*Hình6. Dạng tín hiệu giao tiếp với cảm biến (bit 1)*

* **Ví dụ**: ta nhận được 40 bit dữ liệu như sau:

0011 0101     0000 0000     0001 1000     0000 0000     0100 1101

Tính toán:

8 bit checksum: 0011 0101 + 0000 0000 + 0001 1000 + 0000 0000 = 0100 1101

Độ ẩm: 0011 0101 = 35H = 53% (ở đây do phần thập phân có giá trị 0000 0000, nên ta bỏ qua không tính phần thập phân)

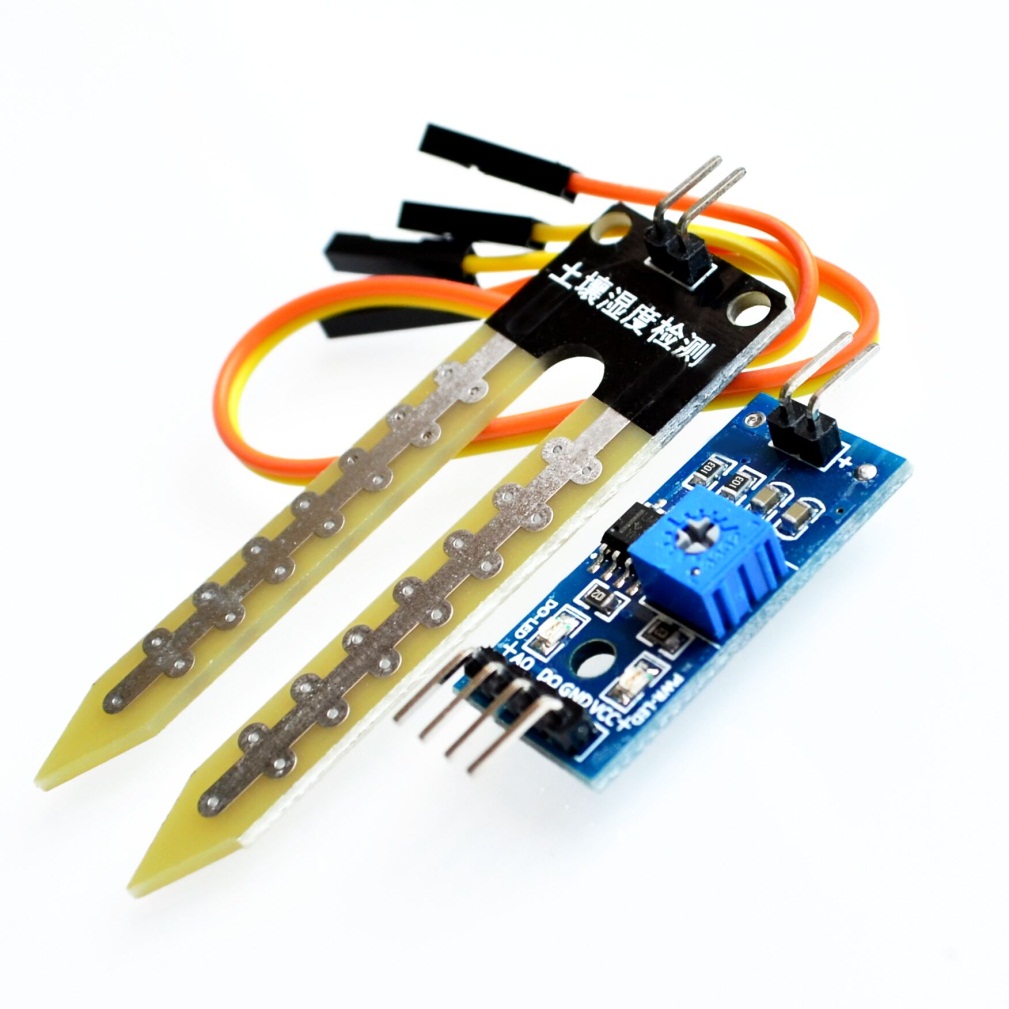
Nhiệt độ: 0001 1000 = 18H = 24°C (ở đây do phần thập phân có giá trị 0000 0000, nên ta bỏ qua không tính phần thập phân)

- 4 chân: VCC( cực (+) nguồn ), DATA(chân tín hiệu), NC, GND(cực (-) nguồn)

* 1. **2.1.4 Nút nhấn**
  2. Trong hệ thống, chúng ta sử dụng nút nhấn để thực hiện reset đưa mạch hoạt động lại ở trạng thái ban đầu nếu cần thiết. Ta sẽ mắc 1 điện trở 10K giữa Vcc và 1 chân của nút bấm (điện trở kéo lên) để xác lập trạng thái là HIGH.
  3. 
  4. *Hình 7. Nút nhấn*
  5. **2.1.5 Cảm biến độ ẩm đất**
* **Giới thiệu**

**Cảm biến độ ẩm đất** trạng thái đầu ra mức thấp (0V), khi đất thiếu nước đầu ra sẽ là mức cao (5V), độ nhạy cao chúng ta có thể điều chỉnh được bằng biến trở. Module **Cảm biến phát hiện độ ẩm đất** có thể sử dung tưới hoa tự động khi không có người quản lý khu vườn của bạn hoặc dùng trong những ứng dụng tương tự như trồng cây.

Độ nhạy của **Cảm biến phát hiện độ ẩm đất** có thể tùy chỉnh được (Bằng cách điều chỉnh chiết áp màu xanh trên board mạch) . Phần đầu đo được cắm vào đất để phát hiện độ ẩm của đất, khi độ ầm của đất đạt ngưỡng thiết lập, đầu ra DO sẽ chuyển trạng thái từ mức thấp lên mức cao



*Hình 8: Cảm biến đo độ ẩm đất*

## ****Thông số kỹ thuật****

**+ Điện áp hoạt động:** 3.3V-5V

**+ Kích thước PCB:** 3cm \* 1.6cm

+ Led đỏ báo nguồn vào, Led xanh báo độ ẩm.

+ **Ic so sánh :** LM393

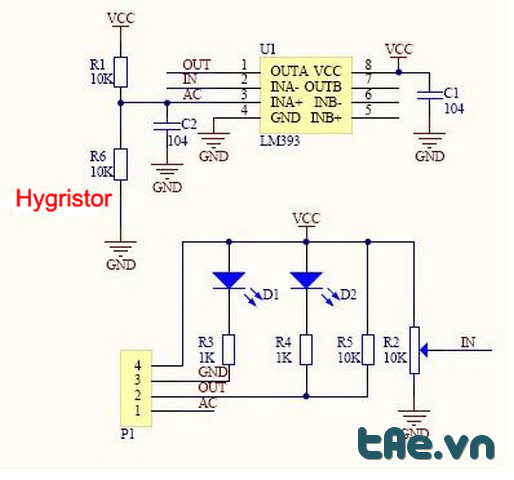
+ **VCC:** 3.3V-5V

+ **GND:** 0V

+ **DO:**Đầu ra tín hiệu số (0 và 1)

+ **AO:** Đầu ra Analog (Tín hiệu tương tự)

## ****Nguyên lý hoạt động****

**

*Hình 9: Sơ đồ nguyên lý hoạt động*

Khi module cảm biến độ ẩm phát hiện, khi đó sẽ có sự thay đổi điện áp ngay tại đầu vào của ic **LM393**. Ic này nhận biết có sự thay đổi nó sẽ đưa ra một tín hiệu 0V để báo hiệu. và thay đổi như thế nào sẽ được tính toán để đọc độ ẩm đất.  
Chi tiết các bạn có thể tham khảo [Datasheet LM393](http://www.ti.com/lit/ds/symlink/lm393-n.pdf)[.](http://www.ti.com.cn/cn/lit/ds/symlink/lm393-n.pdf)

+ Cảm biến độ ẩm đất rất nhạy với độ ẩm môi trường xung quanh

+  Khi độ ẩm đất vượt quá giá trị được thiết lập, ngõ ra của module D0 ở mức giá trị là 0V.

+  Ngõ ra D0 có thể được kết nối trực tiếp với vi điều khiển như (Arduino,PIC,AVR,STM), để phát hiện cao và thấp, và do đó để phát hiện độ ẩm của đất.

+ Đầu ra Analog AO có thể được kết nối với bộ chuyển đổi ADC, bạn có thể nhận được các giá trị chính xác hơn độ ẩm của đất.

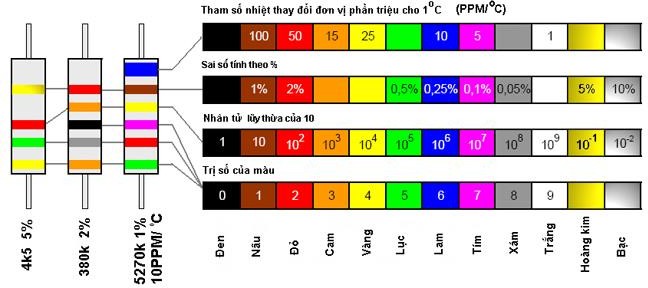
* **Ứng dụng**

+ Đọc độ ẩm đất.

+ Hệ thống tưới tiêu tự động

* 1. **2.1.6 Điện trở**

Điện trở là một [linh kiện điện tử thụ động](http://vi.wikipedia.org/wiki/Linh_ki%E1%BB%87n_%C4%91i%E1%BB%87n_t%E1%BB%AD_th%E1%BB%A5_%C4%91%E1%BB%99ng) trong mạch điện.

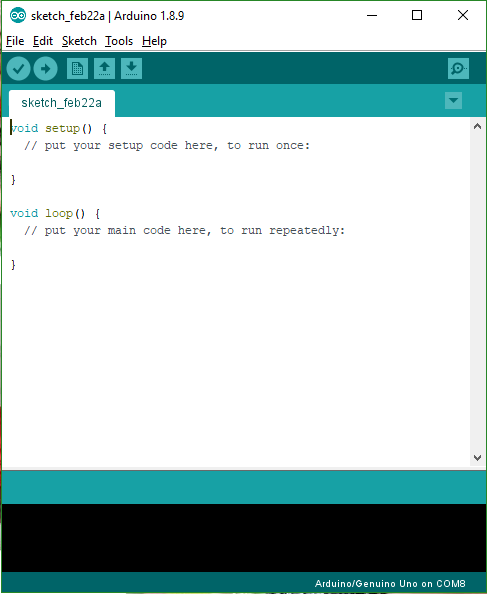


Hình 10. Bảng màu sử dụng để đọc giá trị điện trở.

* 1. **Các phần mềm trong hệ thống**
  2. **2.2.1** Arduino IDE

Arduino IDE được viết tắt  của Arduino Integrated Development Environment là một trình soạn thảo văn bản, giúp bạn viết code để nạp vào bo mạch Arduino.

Arduino IDE được viết bằng ngôn ngữ lập trình Java là ứng dụng đa nền tảng (cross-platform). Ngôn ngữ code cho các chương trình của arduino là bằng C hoặc C++ . Bản thân arduino IDE đã được tích hợp một thư viện phầm mềm thường gọi là "wiring", từ các chương trình "wiring" gốc sẽ giúp bạn thực hiện thao tác code dễ dàng hơn.



*Hình 11. Giao diện chương trình Arduino IDE*

Một chương trình chạy trong arduino được gọi là một sketch, chương trình được định dạng dưới dạng .ino .

* Các giao diện của Arduino IDE

|  |  |  |
| --- | --- | --- |
| https://bizweb.dktcdn.net/100/339/995/files/play-arduino-ide.png?v=1582358879000 | Verify | Kiểm tra lỗi và biên dịch code |
| https://bizweb.dktcdn.net/100/339/995/files/upload-arduino-ide.png?v=1582358899458 | Upload | Dịch và upload code vào bo mạch đã được cài đặt sẵn |
| https://bizweb.dktcdn.net/100/339/995/files/new-arduino-ide.png?v=1582358913291 | New | Tạo sketch mới |
| https://bizweb.dktcdn.net/100/339/995/files/open-arduino-ide.png?v=1582358930179 | Open | Mở một sketch có sẵn |
| https://bizweb.dktcdn.net/100/339/995/files/save-arduino-ide.png?v=1582358947685 | Save | Lưu sketch |
| https://bizweb.dktcdn.net/100/339/995/files/serial-monitor-arduino-ide.png?v=1582358965068 | Serial Monitor | Mở serial monitor. |

Cổng nối tiếp (**Serial port)** là một cổng thông dụng trong các máy tính trong các máy tính truyền thống dùng kết nối các thiết bị ngoại vi với máy tính như: bàn phím, chuột điều khiển, modem, máy quét...Cổng nối tiếp còn có tên gọi khác như: Cổng COM, communication.

Ngày nay, do tốc độ truyền dữ liệu chậm hơn so với các cổng mới ra đời nên các cổng nối tiếp đang dần bị loại bỏ trong các chuẩn máy tính hiện nay, chúng được thay thế bằng các cổng có tốc độ nhanh hơn như: USB, FireWire-Serial Monitor là thành phần của **Arduino IDE**, giúp bo mạch và máy tính có thể gửi và nhận dữ liệu với nhay qua giao tiếp USB.  
-Để mở màn hình Serial Monitor, chúng ta chọn Tool > Serial Monitor

- Để giao tiếp với máy tính, chúng ta cần phải thiết lập tốc độ kết nối qua cổng usb bằng hàm Serial.begin(). Sau đó dùng hàm Serial.print() để hiện thị nội dung ra màn hình Serial Monitor

* Nạp chương trình mới lên Arduino IDE:

­- Khởi động **Arduino IDE**. Vào Tool/Chọn Board mà bạn đang sử dụng

- **Chọn cổng COM**

**-Upload chương trình**

* Cách thêm thư viện :

- Bạn khởi động**Arduino IDE**, click vào Sketch trên thanh công cụ chọn **Include Library > Manage Libraries.**

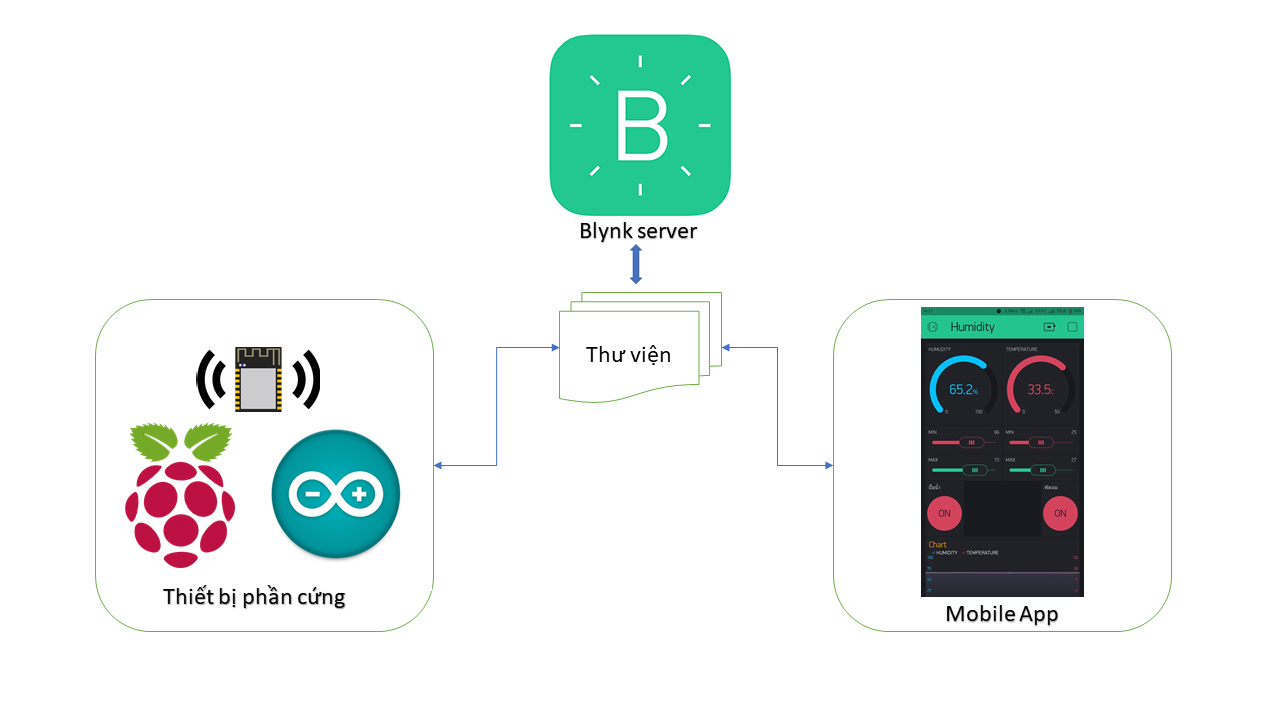
**-** Bạn click vào Filter your seach để tiềm kiếm thư vện hoặc sử dụng **Type** và **Topic** để lọc ra các kết quả tìm kiếm mà bạn đang cần.

**-** Khi đã chọn được thư viện cần, nhấn **Install** để tiến hành cài đặt.

**-**Tiến hành kiểm tra xem thư viện đã được thêm vào chưa: Vào **File** > **Examples**

* 1. **2.2.2** Ứng dụng BLYNK
  2. Blynk là một ứng dụng chạy trên nền tảng iOS và Android để điều khiển và giám sát thiết bị thông qua internet. Blynk không bị ràng buộc với những phần cứng cụ thể nào cả, thay vào đó, nó hỗ trợ phần cứng cho bạn lựa chọn như Arduino, Raspberry Pi, ESP8266 và nhiều module phần cứng phổ biến khác.
* Có ba thành phần chính trong nền tảng:
* **Blynk App** - cho phép tạo giao diện cho sản phẩm của bạn bằng cách kéo thả các widget khác nhau mà nhà cung cấp đã thiết kế sẵn.
* **Blynk Server** - chịu trách nhiệm xử lý dữ liệu trung tâm giữa điện thoại, máy tính bảng và phần cứng. Bạn có thể sử dụng Blynk Cloud của Blynk cung cấp hoặc tự tạo máy chủ Blynk riêng của bạn. Vì đây là mã nguồn mở, nên bạn có thể dễ dàng intergrate vào các thiết bị và thậm chí có thể sử dụng Raspberry Pi làm server của bạn.
* **Library Blynk** – support cho hầu hết tất cả các nền tảng phần cứng phổ biến - cho phép giao tiếp với máy chủ và xử lý tất cả các lệnh đến và đi.

Mỗi khi bạn nhấn một nút trong ứng dụng Blynk, yêu cầu sẽ chuyển đến server của Blynk, server sẽ kết nối đến phần cứng của bạn thông qua library . Tương tự thiết bị phần cứng sẽ truyền dữ liệu ngược lại đến server.

* 1. 
  2. *Hình 11: Sơ đồ trao đổi dữ liệu trên Blynk*

# Tính năng, đặc điểm

* Cung cấp API & giao diện người dùng tương tự cho tất cả các thiết bị và phần cứng được hỗ trợ
* Kết nối với server bằng cách sử dụng:
* Wifi
* Bluetooth và BLE
* Ethernet
* USB (Serial)
* GSM
* …
* Các tiện ích trên giao diện được nhà cung cấp dễ sử dụng
* Thao tác kéo thả trực tiếp giao diện mà không cần viết mã
* Dễ dàng tích hợp và thêm chức năng mới bằng cách sử dụng các cổng kết nối ảo được tích hợp trên blynk app
* Theo dõi lịch sử dữ liệu
* Thông tin liên lạc từ thiết bị đến thiết bị bằng Widget
* Gửi email, tweet, thông báo realtime, v.v.
* Những lý do nên sử dụng Blynk:  
  – Dễ sử dụng: việc cài đặt ứng dụng và đăng ký tài khoản trên điện thoại rất đơn giản cho cả IOS và Android  
  – Chức năng phong phú: Blynk hỗ trợ rất nhiều chức năng với giao diện đẹp và thân thiện, bạn chỉ việc kéo thả đối tượng và sử dụng nó.  
  – Không phải lập trình ứng dụng: nếu bạn không có kiến thức về lập trình app cho Android cũng như IOS thì Blynk là một ứng dụng tuyệt vời để giúp bạn khám phá thế giới IOTs.  
  – Điều khiển, giám sát thiết bị ở bất kì đâu thông qua internet với khả năng đồng bộ hóa trạng thái và thiết bị
* Chức năng
* 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
* 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
* Hiển thị
* 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
  1. **2.2.3** Thư viện [SimpleTimer](https://github.com/schinken/SimpleTimer)
     1. Thư viện [DHT](https://github.com/adafruit/DHT-sensor-library)
  2. **2.3 Tính toán các thông số**
  3. **2.3.1** Tính toán công thức chuyển đổi từ giá trị số đọc ADC thành đại lượng vật lý cần đo nhiệt độ.

Nhiệt độ sẽ được đo bằng cách đo điện áp đầu ra của cảm biến DHT11 hay nói một cách khác là cảm biến DHT11 cho ra một giá trị điện áp nhất định ở chân thứ 2 và điện áp đó ứng với một giá trị nhiệt độ xác định.

Việc đo nhiệt độ thường được thực hiện bằng cách : DHT11 => ADC => Vi điều khiển . Có nghĩa là cảm biến nhiệt độ sẽ đo nhiệt độ , ứng với nhiệt độ đó sẽ cho ra một giá trị điện áp tương ứng , điện áp này được chuyển thành giá trị ADC giúp vi điều khiển hiểu được và thực hiện công việc xuất dữ liệu ra màn hình điện thoại

Chế độ ADC trong ESP8266 là ADC 10 bit nên sẽ nhận kết quả từ 0 ÷ 1023 ( tương ứng từ 0V ÷ 5V trong cảm biến DHT11).

Ta có :

* U là điện áp đầu ra của DHT11
* T là nhiệt độ đo được
* K là hệ số theo nhiệt độ của DHT11, cứ mỗi 10mV là 1
* Giả sử điện áp đầu ra của DHT11 là 5V = 5000mV mà cứ mỗi 10mV là 1 , suy ra 5000mV sẽ tương ứng với nhiệt độ 500.

Mà 500 ( hay 5000mV ) sẽ ứng với giá trị số đọc ADC sẽ là 1023. Gọi ADC là giá trị số đọc tương ứng với nhiệt độ đo được và ta có :

500 ==> 1023

Suy ra : T ==> ADC

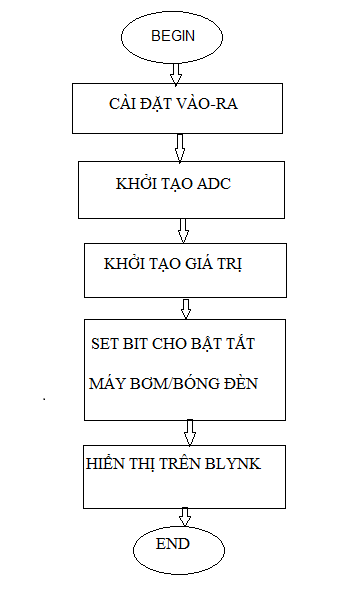
Vậy công thức chuyển đổi giá trị từ giá trị số đọc ADC sang nhiệt độ cần tính là :

T = ADC x 500/1023 ()

**CHƯƠNG 3: MÔ PHỎNG HỆ THỐNG**

* 1. **Thiết kế mạch ng**u**yên lý**

1. 1. 1. Thuật toán trong chương trình chính

****

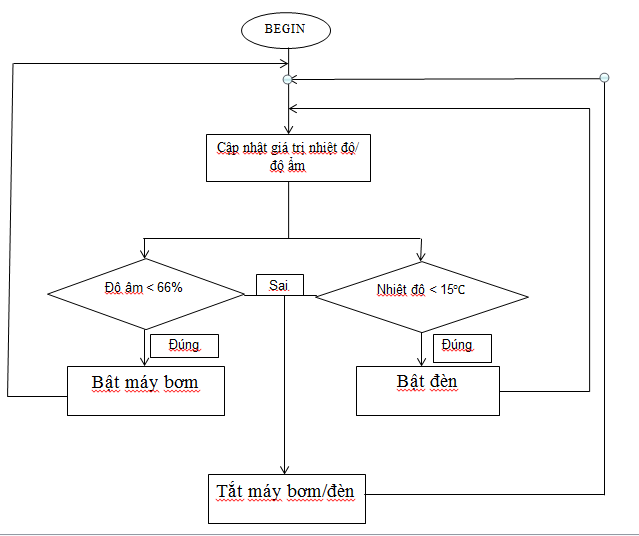
* + 1. Thuật toán điều khiển nút nhấn

Nút 1 (D0) điều khiển bơm (D6)

Nút 2 (D1) điều khiển đèn (D7)

Nút 3 (D4) dùng để bắt wifi mới ( không có sẵn)

* + 1. Thuật toán tự động bật máy bơm/đèn



* + 1. Thuật toán hiển thị trên Blynk