BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC XÂY DỰNG**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN NĂM HỌC 2016 - 2017**

**NGHIÊN CỨU PHÁT TRIỂN HỆ THỐNG VƯỜN THÔNG MINH**

**Mã số đề tài: CNTT-2017-02**

**Sinh viên thực hiện, mã số sinh viên, lớp:**

**Nguyễn Mạnh Hưng – 1504360 - 60PM2**

**Nguyễn Thạch Hưng – 1537460 - 60PM2**

**Lê Quang Tuấn Anh – 1537260 - 60PM2**

**Giáo viên hướng dẫn** ThS. Lê Đức Quang

MỤC LỤC

[MỞ ĐẦU 4](#_Toc479864794)

[1.Đặt vấn đề 4](#_Toc479864795)

[2. Mục tiêu đề tài 4](#_Toc479864796)

[3. Phương pháp nghiên cứu 4](#_Toc479864797)

[4. Khảo sát các hệ thống hiện có 4](#_Toc479864798)

[4.1 Dự Án Ở Việt Nam : Trồng rau bằng Smartphone của nhóm Hachi 4](#_Toc479864799)

[4.2 Các Dự Án Quốc Tế Hiện Nay: 5](#_Toc479864800)

[CHƯƠNG I : ARDUINO VÀ CÁC LINH KIỆN 7](#_Toc479864801)

[1. Giới thiệu về Arduino 7](#_Toc479864802)

[1.1 Lịch sử ra đời 7](#_Toc479864803)

[1.2.Cấu trúc: 7](#_Toc479864804)

[2. Các linh kiện 9](#_Toc479864805)

[2.1.RF24 9](#_Toc479864806)

[2.2. Cảm biến lưu lượng 11](#_Toc479864807)

[2.3. Cảm biến độ ẩm đất TH-50K 12](#_Toc479864808)

[2.4. Màn hình LCD 12](#_Toc479864809)

[2.5. Van 15](#_Toc479864810)

[2.6 Máy bơm MB385 16](#_Toc479864811)

[CHƯƠNG II : LOGIC MỜ 17](#_Toc479864812)

[1. Giới thiệu về Logic mờ 17](#_Toc479864813)

[2. Điều khiển mờ 18](#_Toc479864814)

[2.1. Định nghĩa tập mờ 18](#_Toc479864815)

[2.2 Các thuật ngữ trong logic mờ 18](#_Toc479864816)

[2.3 Biến ngôn ngữ 19](#_Toc479864817)

[2.4 Giải mờ 21](#_Toc479864818)

[CHƯƠNG III : TRIỂN KHAI LẮP RÁP HỆ THỐNG 23](#_Toc479864819)

[1. Ý tưởng và chức năng hệ thống 23](#_Toc479864820)

[2. Cách thức tính toán lượng nước cần bơm 23](#_Toc479864821)

[2.1. Mờ hóa 23](#_Toc479864822)

[2.2. Luật hợp thành mờ 25](#_Toc479864823)

[2.3. Giải mờ 25](#_Toc479864824)

[3.Sơ đồ hệ thống 26](#_Toc479864825)

[4. Lắp ráp các thành phần 26](#_Toc479864826)

[CHƯƠNG IV : THỬ NGHIỆM SẢN PHẨM 38](#_Toc479864827)

[1. Điều kiện thử nghiệm: 38](#_Toc479864828)

[2. Kết quả 40](#_Toc479864829)

[3. Phân tích và kết luận 41](#_Toc479864830)

[CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 42](#_Toc479864831)

[1. Kết luận 42](#_Toc479864832)

[2. Phát triển 42](#_Toc479864833)

**DANH SÁCH HÌNH VẼ**

[Hình 1. Arduino UNO 11](#_Toc479866010)

[Hình 2. Arduino NANO 11](#_Toc479866011)

[Hình 3. Cấu trúc RF24 12](#_Toc479866012)

[Hình 4. Sơ đồ RF24 13](file:///C:\Users\Vaio\Desktop\Chương%201.docx#_Toc479866013)

[Hình 5. Tập mờ độ ẩm 26](#_Toc479866014)

[Hình 6. Tập mờ lượng nước cần bơm 27](#_Toc479866015)

[Hình 7. Sơ đồ hệ thống 28](file:///C:\Users\Vaio\Desktop\Chương%201.docx#_Toc479866016)

[Hình 8. Bộ đo đạc và phát tín hiệu 40](#_Toc479866017)

[Hình 9. Bộ thu và điều khiển 41](#_Toc479866018)

[Hình 10. Thiết bị thí nghiệm 41](#_Toc479866019)

# MỞ ĐẦU

## 1.Đặt vấn đề

Ngày nay, công nghệ đã trở thành một phần quen thuộc với đời sống chúng ta. Từ đứa trẻ cho đến những người cao tuổi cũng có thể hiểu và sử dụng đôi chút những thiết bị điện tử. Việc ứng dụng công nghệ vào đời sống cũng không còn là một khái niệm quá xa lạ trong thời đại Internet of Things ( IoT ) bây giờ.

Hưởng ứng phong trào này, cùng cảm hứng được gợi từ việc trồng rau tại gia đang trở thành một xu hướng tại Việt Nam, chúng em nghiên cứu và tìm cách sử dụng Arduino trong công việc tự động hóa tưới nước cho cây trong môi trường hộ gia đình. Với hệ thống này , bạn chỉ việc chỉ hẹn giờ, thiết lập số lượng nước phù hợp và cây sẽ được tưới đúng với yêu cầu.

## 2. Mục tiêu đề tài

- Tìm hiểu các thiết bị phần cứng như Arduino & các loại cảm biến khác

- Nghiên cứu , phát triển hệ thống tưới tự động quy mô nhỏ dành cho hộ gia đình.

## 3. Phương pháp nghiên cứu

- Nghiên cứu các thiết bị phần cứng dựa trên các hướng dẫn và tài liệu trên mạng Internet và thử nghiệm các thiết bị này để nắm được cách sử dụng cũng như các đặc điểm kỹ thuật

- Tìm hiểu các lý thuyết liên quan đến tưới cây như độ ẩm, lượng nước từ đó xây dựng phương pháp tưới tự động dựa trên việc kết hợp các thiết bị phần cứng

## 4. Khảo sát các hệ thống hiện có

### 4.1 Dự Án Ở Việt Nam : Trồng rau bằng Smartphone của nhóm Hachi

**+ Chức Năng**: Đây là ý tưởng của một nhóm bạn trẻ từ 20 – 25 tuổi, với mong muốn ứng dụng công nghệ thông tin vào nông nghiệp giúp người dân được sử dụng thực phẩm sạch. Tiến hành thử nghiệm giải pháp ứng dụng công nghệ IoT (Internet kết nối vạn vật) trên hệ thống thủy canh thông minh, đạt kết quả khả quan. Việc trồng rau trong nhà tưởng chừng khó khăn lại trở nên đơn giản và dễ dàng hơn bao giờ hết.. Toàn bộ dữ liệu sẽ được gửi lên server của Hachi (tên nhóm khởi nghiệp) và cập nhật trong ứng dụng di động Hachi của người dùng. Từ đó người dùng có thể dễ dàng theo dõi tình trạng môi trường và đưa ra những giải pháp hữu hiệu như thiết lập thời gian tưới, thời gian chiếu sáng và các mức độ cảnh báo thông qua smartphone. Hệ thống sử dụng đèn led chiếu sáng nhân tạo nên có thể đặt trong nhà. Việc ứng dụng giải pháp này có thể giúp tăng 30 – 50% tốc độ sinh trưởng của cây, đảm bảo sạch, cách ly hoàn toàn với môi trường sâu bệnh bên ngoài và trồng được những loại cây trái vụ, khó trồng ở điều kiện tự nhiên.

**+Phạm Vi Ứng Dụng**: nhắm đến đối tượng khách cư dân đô thị – những người quen sử dụng Internet, sử dụng smartphone và đang có xu hướng trồng rau sạch tại nhà cũng như chủ vườn, chủ trang trại quan tâm đến những giải pháp mới giúp tăng năng suất, chất lượng nông sản.

**+Giá Thành**: Do sử dụng công nghệ đèn LED nông nghiệp, nên hệ thống của Hachi rất tiết kiệm điện. Đối với giàn tiêu chuẩn 2 tầng, dự kiến mỗi tháng khoảng 50.000 VNĐ chi phí cho việc chiếu sáng.Chi phí hạt giống tuỳ thuộc vào loại cây giống, hạt giống và nhu cầu của quý khách hàng.

**+Ưu Điểm**: Ưu điểm vượt trội của sản phẩm này là sử dụng các loại cảm biến, tự động theo dõi, giám sát điều kiện môi trường như: nhiệt độ, độ ẩm, ánh sáng thông qua ứng dụng trên smartphone. Không cần đất, không cần ánh sáng, người dùng chỉ cần bổ sung nước 2 lần/tuần và thay hạt giống cây sau khi thu hoạch cây cũ.

**+Nhược Điểm**: khó khăn cho những người không quen tiếp xúc với công nghệ đặc biệt là smartphone

### 4.2 Các Dự Án Quốc Tế Hiện Nay:

**⮚ Công Nghệ Tưới Nhỏ Giọt của Israel:** Tưới cây, một việc tưởng chừng đơn giản đã được làm từ thuở nào, nhưng trước nguy cơ cạn kiệt nước và phải đảm bảo gia tăng năng suất với điều kiện tiết kiệm nhất về phân bón, nước tưới, đảm bảo chu kỳ sinh trưởng cây trồng, tiết kiệm nhân lực… đã tạo cơ hội cho công nghệ tưới nhỏ ngày càng hoàn thiện và phát triển. Tưới nhỏ giọt là kỹ thuật tưới cung cấp nước vào đất dưới dạng các giọt nước nhỏ ra đều đều từ công cụ hay thiết bị tạo giọt đặt tại một số điểm trên mặt đất gần gốc cây. Tưới nhỏ giọt đã có từ thời kỳ cổ đại với các bình đất sét đục lỗ cho nước thấm qua hay hệ thống ống dẫn đục lỗ tại Đức (năm 1920) nhưng phải đến năm 1959, nhờ công sức của hai cha con Simcha và Yeshayahu Blass người Israel, phương pháp này mới được hoàn thiện.

**+ Chức Năng:** Hệ thống tưới nhỏ giọt đơn giản bao gồm bồn chứa nước, hệ thống ống dẫn và đầu tưới nhỏ giọt hay dây nhỏ giọt. Phần điều khiển tự động bao gồm van điện điều khiển khu vực tưới, bộ lọc, bộ điều khiển số lần và thời gian tưới trong ngày. Hệ thống tưới nhỏ giọt có thể kết hợp với bộ châm phân tự động, cung cấp phân bón khi tưới tiêu, cách này được gọi là tưới bón. Đặc biệt hơn, còn có them đầu cảm ứng cắm vào đất và lắp đặt chung với hệ thống tưới nhỏ giọt. Đầu này có thể cảm ứng được độ ẩm của đất và điều khiển quá trình tưới dựa trên nguyên tắc thông minh của con người "đất khô thì tưới, đất ẩm thì ngưng". Các hệ thống tưới nhỏ giọt sẽ được thiết kế, chế tạo dựa trên một nguyên lý chung, thế nhưng mỗi hệ thống đều mang vẻ độc đáo riêng tuỳ thuộc loại cây trồng, thổ nhưỡng, địa hình khi vực, quy mô mà hệ thống tưới…

**+Phạm Vi Ứng Dụng:** Tưới nhỏ giọt được áp dụng nhiều trong các nông trang, nhà kính và vườn gia đình cũng như thích hợp nhất với các loại cạy như dừa, nho, chuối, cam quýt, dâu tây, mía, bong, ngô, cà chua và một số cây công nghiệp khác.

**+Giá Thành:** Đến nay, hệ thống tưới nhỏ giọt là biện pháp tưới tiêu tiết kiệm nước nhất, giảm đến 30-60% nước so với phương pháp tưới truyền thống

**+Ưu Điểm:**. Nông dân có thể mang nước, phân bón đến đúng địa chỉ với liều lượng vừa đủ dùng thông qua hệ thống van, đường ống, máy bơm và hiện đại hơn là kết nối với hệ thống máy tính kiểm soát

**+Nhược Điểm:** nhà sản xuất hệ thống tưới nhỏ giọt phải có kinh nghiệm về chế tạo cơ khí, điện, điện tử, tự động hoá cả lập trình trên máy tính… và còn phải am hiểu "tính nết" của cây trồng và các loại phân bón có thể được sử dụng. Có thể nói sản xuất một hệ thống tưới nhỏ giọt tương tự như tạo hoá tạo ra một con người, trên một cơ chế giống nhau, nhưng hàng triệu người thì khác biệt nhau vô cùng.

# CHƯƠNG I : ARDUINO VÀ CÁC LINH KIỆN

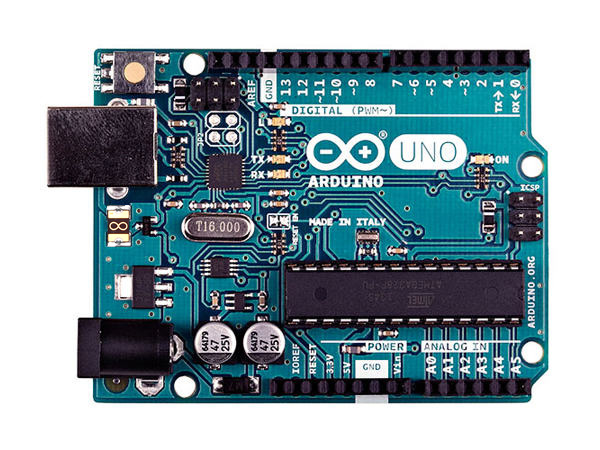
## 1. Giới thiệu về Arduino

### 1.1 Lịch sử ra đời

* Tiền thân của Arduino là Basic Stamps*.* Basic Stamps là 1 bộ vi điều khiển ( microcontroller ) sử dụng bộ thông dịch chuyên dụng BASIC ( PBASIC ) xây dựng nên ROM. Giá của Basic Stamps rất cao , lên đến 100$.
* Arduino được khởi động vào năm 2005 như là một dự án dành cho sinh viên Interaction Design Institute Ivrea (Viện thiết kế tương tác Ivrea) tại Ivrea, Italy. Dự án được đứng đầu bởi Massimo Banzi, một giảng viên tại Ivrea.
* Lý thuyết phần cứng được đóng góp bởi một sinh viên Hernando Barragan người Colombia. Sau khi nền tảng Wiring hoàn thành, các nhà nghiên cứu đã không làm việc tiếp trên nền tảng này. Thay vì vậy, họ sao chép mã nguồn của Wiring để bắt đầu 1 dự án khác với cái tên là Arduino. Dự án này sau đó đã giúp nó nhẹ hơn, rẻ hơn, và khả dụng đối với cộng đồng mã nguồn mở.
* Cái tên "Arduino" đến từ một quán bar tại Ivrea, nơi một vài nhà sáng lập của dự án này thường xuyên gặp mặt. Bản thân quán bar này được lấy tên là Arduino, theo tên của Bá tước của Ivrea, và là vua của Italy từ năm 1002 đến 1014.
* Viện Ivrea cuối cùng bị đóng cửa, vì vậy các nhà nghiên cứu, một trong số đó là David Cuarlielles, đã phổ biến ý tưởng này.

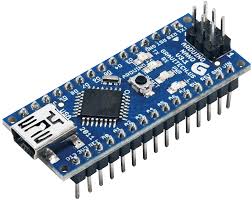
### 1.2.Cấu trúc

* Một mạch Arduino bao gồm **một vi điều khiển AVR** với nhiều linh kiện bổ sung giúp dễ dàng lập trình và có thể mở rộng với các mạch khác.
* Arduino cho phép người dùng kết nối với CPU của board với các module thêm vào có thể dễ dàng chuyển đổi, được gọi là *shield*. Vài shield truyền thông với board Arduino trực tiếp thông qua các chân khác nhau, nhưng nhiều shield được định địa chỉ thông qua **serial bus I²C** (nhiều shield có thể được xếp chồng và sử dụng dưới dạng song song).
* Arduino chính thức thường sử dụng các dòng **chip megaAVR** (ATmega8, ATmega168, ATmega328, ATmega1280, và ATmega2560).
* Hầu hết các mạch gồm **một bộ điều chỉnh tuyến tính 5V** và **một thạch anh dao động 16 MHz** (hoặc **bộ cộng hưởng ceramic** trong một vài biến thể), mặc dù một vài thiết kế như LilyPad chạy tại 8 MHz và bỏ qua bộ điều chỉnh điện áp onboard do hạn chế về kích cỡ thiết bị.
* Theo nguyên tắc, khi sử dụng ngăn xếp phần mềm Arduino, tất cả các board được lập trình thông qua một kết nối RS-232, nhưng cách thức thực hiện lại tùy thuộc vào đời phần cứng. Các board Serial Arduino có chứa **một mạch chuyển đổi giữa RS232 sang TTL**. Các board Arduino hiện tại được lập trình thông qua cổng USB, thực hiện thông qua chip chuyển đổi USB-to-serial như là FTDI FT232.
* Board Arduino sẽ đưa ra hầu hết **các chân I/O của vi điều khiển** để sử dụng cho những mạch ngoài. Những chân này được thiết kế nằm phía trên mặt board, thông qua các header cái 0.10-inch (2.5 mm). Các board Arduino Nano, và Arduino-compatible Bare Bones Board và Boarduino có thể cung cấp các chân header đực ở mặt trên của board dùng để cắm vào các breadboard.
* Có nhiều biến thể như Arduino-compatible và Arduino-derived. Một vài trong số đó có chức năng tương đương với Arduino và có thể sử dụng để thay thế qua lại. Nhiều mở rộng cho Arduino được thực thiện bằng cách thêm vào các driver đầu ra, thường sử dụng trong các trường học để đơn giản hóa các cấu trúc của các 'con rệp' và các robot nhỏ. Những board khác thường tương đương về điện nhưng có thay đổi về hình dạng-đôi khi còn duy trì độ tương thích với các shield, đôi khi không. Vài biến thể sử dụng bộ vi xử lý hoàn toàn khác biệt, với các mức độ tương thích khác nhau.
* Mạch Arduino bao gồm:
* 54 chân digital (15 có thể được sử dụng như các chân PWM)
* 16 đầu vào analog
* 4 UARTs (cổng nối tiếp phần cứng)
* 1 thạch anh 16 MHz
* 1 cổng kết nối USB
* 1 jack cắm điện
* 1 đầu ICSP
* 1 nút reset

****

Hình . Arduino UNO

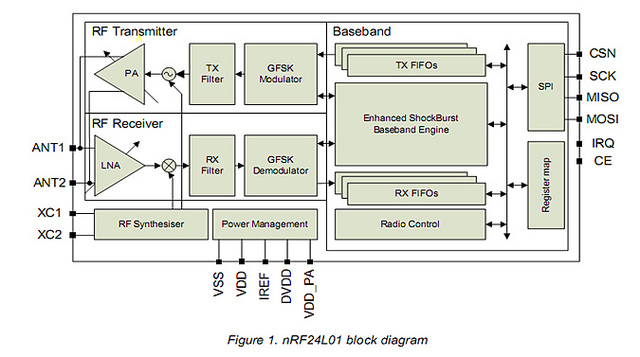
Nó chứa **tất cả** mọi thứ cần thiết để hỗ trợ các vi điều khiển. Arduino Mega2560 khác với tất cả các vi xử lý trước giờ vì không sử dụng FTDI chip điều khiển chuyển tín hiệu từ USB để xử lý. Thay vào đó, nó sử dụng ATmega16U2 lập trình như là một công cụ chuyển đổi tín hiệu từ USB. Ngoài ra, Arduino Mega2560 cơ bản vẫn giống Arduino Uno R3, chỉ khác số lượng chân và nhiều tính năng mạnh mẽ hơn, nên các bạn vẫn có thể lập trình cho con vi điều khiển này bằng chương trình lập trình cho Arduino Uno R3.

****

Hình 2. Arduino NANO

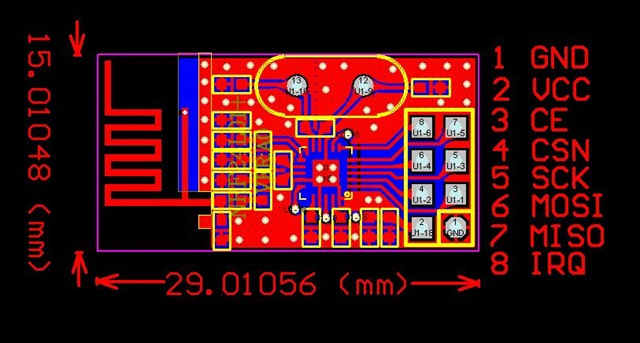
## 2. Các linh kiện

### 2.1.RF24

2.1.1.**Thông số kỹ thuật:**  
- Radio  
o Hoạt động ở giải tần 2.4G  
o Có 126 kênh  
o Truyền và nhận dữ liệu  
o Truyền tốc độ cao 1Mbps hoặc 2Mbps.  
- Công suất phát:  
o Có thể cài đặt được 4 công suất nguồn phát: 0,-6,-12,-18dBm.  
- Thu:  
o Có bộ lọc nhiễu tại đầu thu  
o Kếch đại bị ảnh hưởng bởi nhiễu thấp (LNA)  
- Nguồn cấp:  
o Hoạt động từ 1.9-3.6V.  
o Các chân IO chạy được cả 3.3 lẫn 5V.  
- Giao tiếp:  
o 4 pin SPI  
o Tốc độ tối đa 8Mbps  
o 3-32 bytes trên 1 khung truyền nhận  


Hình 3. Cấu trúc RF24

-**Phân tích:**  
o Modul nRF24L01 hoạt động ở tần số sóng ngắn 2.4G nên Modul này khả năng truyền dữ liệu tốc độ cao và truyền nhận dữ liệu trong điều kiện môi trường có vật cản  
o Modul nRF24L01 có 126 kênh truyền. Điều này giúp ta có thể truyền nhận dữ liệu trên nhiều kênh khác nhau.  
o Modul khả năng thay đổi công suất phát bằng chương trình, điều này giúp nó có thể hoạt động trong chế độ tiết kiệm năng lượng.  
o Chú ý: Điện áp cung cấp cho là 1.93.6V. Điện áp thường cung cấp là 3.3V. Nhưng các chân IO tương thích với chuẩn 5V. Điều này giúp nó giao tiếp rộng dãi với các dòng vi điều khiển.

**Sơ đồ phần cứng:  
**

Hình 4. Sơ đồ RF24

### 2.2. Cảm biến lưu lượng

Tính năng   
- Chất liệu bằng nhựa bên trong có cánh quạt nước và cảm biến hall.   
Khi nước chảy qua van cảm biến làm động cơ quay dẫn đến sự thay đổi trạng thái đầu ra của cảm biến Hall, đâu ra tín hiệu xung.

**Thông số**

|  |  |
| --- | --- |
| **Working voltage** | 5V-24V |
| Maximum current | 15 mA(DC 5V) |
| Weight | 43 g |
| Flow rate range | 1~30 L/min |
| Operating temperature  Liquid temperature | 0°C~80°C <120°C |
| Operating humidity | 35%~90%RH |
| Operating pressure | under 1.75Mpa |
| Store temperature | -25°C~+80°C |
| Store humidity | 25%~90%RH |

Ký hiệu dây cảm biến:

- Black (Đen): GND

- Red (Đỏ): VCC supply

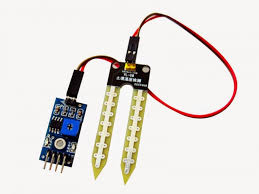
- Yellow (Vàng): Tín hiệu ra của hall sensor

Tần số tín hiệu đầu ra: F=7.5xQ ( L/Phút)

Trong đó:

Q: Lưu lượng nước  
F: Tần số tín hiệu đầu ra (Hz)  
7.5: Hằng số   
VD:  
1L nước sẽ có công thức : 1x7.5x60 = 450 xung

### ****2.3. Cảm biến độ ẩm đất TH-50K****

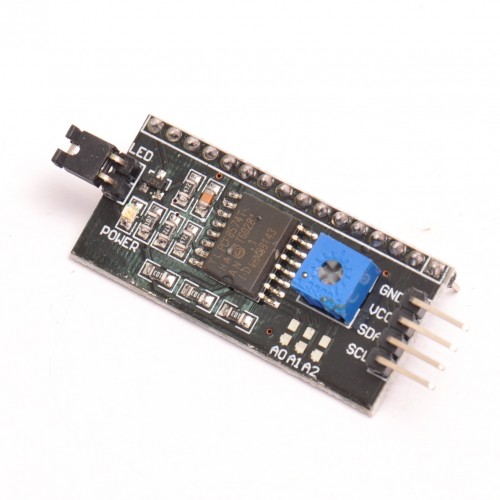


* Tính năng :
  + Phát hiện độ ẩm trong đất
  + Sử dụng bằng cách cắm cảm biến vào trong đất
* Thông số kỹ thuật:
  + Sử dụng điện áp 5V
  + Đầu ra A0, D0, VCC,GND
  + Module khi chưa phát hiện độ ẩm thì D0 cho mức 1 , độ ẩm vượt ngưỡng thiết lập cho mức 0. Chân A0 cho kết quả đo dưới dạng Analog có giá trị từ 0÷1024

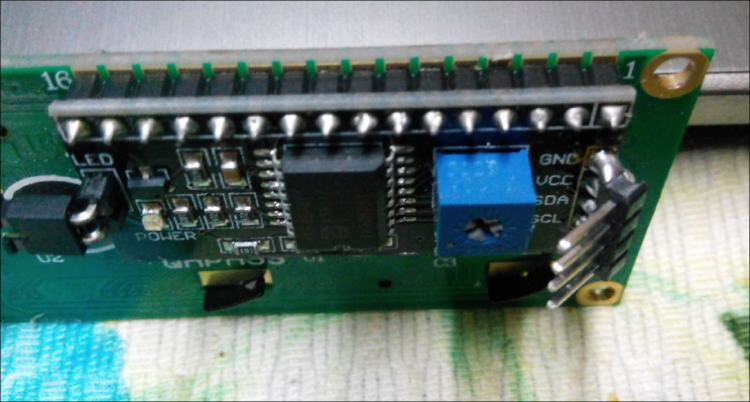
### ****2.4. Màn hình LCD****

[](http://k1.arduino.vn/img/2014/05/27/0/509_12320-1401186400-0-lcd16x2-truoc-500x500.jpg)

Thông thường, để sử dụng màn hình LCD, bạn sẽ phải mất rất nhiều chân trên Arduino để điều khiển. Do vậy, để đơn giản hóa công việc, người ta đã tạo ra một loại mạch điều khiển màn hình LCD sử dụng giao tiếp I2C. Nói một cách đơn giản, bạn chỉ tốn ... 2 dây để điều khiển màn hình, thay vì 8 dây như cách thông thường.

[](http://k2.arduino.vn/img/2014/05/28/0/493_88220-1401264055-0-mach-dieu-khien-man-hinh-lcd-1602-1-500x500.jpg)

Bạn chỉ việc hàn mạch vào như thế này là xong.

[](http://k2.arduino.vn/img/2014/05/28/0/494_88220-1401265279-0-eugffgj.750x402.png)

2 chân SDA và SCL là 2 chân tín hiệu dùng cho giao tiếp I2C.

**Kết nối module màn hình với Arduino**

|  |  |
| --- | --- |
| **Module màn hình LCD (16x2)** | **Arduino** |
| GND | GND |
| Vcc | 5V |
| SDA | A4 |
| SCL | A5 |

**Lập trình**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,16,2);

//0x27 là địa chỉ màn hình trong bus I2C. Phần này chúng ta không cần phải quá bận tâm vì hầu hết màn hình (20x4,...) đều như thế này!

//16 là số cột của màn hình (nếu dùng loại màn hình 20x4) thì thay bằng 20

//2 là số dòng của màn hình (nếu dùng loại màn hình 20x4) thì thay bằng 4

void setup() {

lcd.init(); //Khởi động màn hình.

lcd.backlight(); //Bật đèn nền

lcd.print("Hello world"); //Xuất ra chữ Hello world

lcd.setCursor(0,1); //Đưa con trỏ tới hàng 1, cột 0

lcd.print("I love Arduino !");

}

void loop()

{

}

### 2.5. Van

Van nước điện tử Solenoid là thiết bị đóng ngắt nước tự động. Được ứng dụng nhiều trong hệ thống tưới tự động, chiết rót tự động, thiết bị vê sinh..v.v

Van là dạng thường đóng , khi chưa có điện vào cuộn dây thì van khoá , khi có điện van mở.

**Thông số kỹ thuật**

* Điện áp cấp cho cuộn dây: 12VDC
* Áp lực nước : 0.2 Mpa - 0.8 Mpa
* Nhiệt độ làm việc : 1 ℃ -75 ℃
* Thời gian đáp ứng : ON  ≤0.15s OFF ≤0.3s
* Đường kính ống : 1/2'' (inch)

### 2.6 Máy bơm MB385



* Tính năng:
  + Máy bơm nước mini
* Thông số kỹ thuật
  + Điện áp 6 đến 12V
  + Dòng tiêu thụ 0.6 – 2A
  + Công suất 5-12W
  + Lưu lượng 1-2l/phút

# CHƯƠNG II : LOGIC MỜ

## Giới thiệu về Logic mờ

Logic mờ (tiếng Anh: *Fuzzy logic*) được phát triển từ lý thuyết tập mờ để thực hiện lập luận một cách xấp xỉ thay vì lập luận chính xác theo lôgic vị từ cổ điển. Logic mờ có thể được coi là mặt ứng dụng của lý thuyết tập mờ để xử lý các giá trị trong thế giới thực cho các bài toán phức tạp

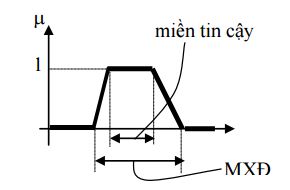
Lôgic mờ cho phép độ liên thuộc có giá trị trong khoảng đóng 0 và 1, và ở hình thức ngôn từ, các khái niệm không chính xác như "hơi hơi", "gần như", "khá là" và "rất". Cụ thể, nó cho phép quan hệ thành viên không đầy đủ giữa thành viên và tập hợp.

Logic mờ được công bố lần đầu tiên tại Mỹ vào năm 1965 do giáo sư Lotfi Zadeh. Kể từ đó, logic mờ đã có nhiều phát triển qua các chặng đường sau : phát minh ở Mỹ, áp dụng ở Châu Âu và đưa vào các sản phẩm thương mại ở Nhật. Ứng dụng đầu tiên của logic mờ vào công nghiệp được thực hiện ở Châu Âu, khoảng sau năm 1970. Tại trường Queen Mary ở Luân Đôn – Anh, Ebrahim Mamdani dùng logic mờ để điều khiển một máy hơi nước mà trước đây ông ấy không thể điều khiển được bằng các kỹ thuật cổ điển. Và tại Đức, Hans Zimmermann dùng logic mờ cho các hệ ra quyết định. Liên tiếp sau đó, logic mờ được áp dụng vào các lĩnh vực khác như điều khiển lò xi măng, … nhưng vẫn không được chấp nhận rộng rãi trong công nghiệp. Kể từ năm 1980, logic mờ đạt được nhiều thành công trong các ứng dụng ra quyết định và phân tích dữ liệu ở Châu Âu. Nhiều kỹ thuật logic mờ cao cấp được nghiên cứu và phát triển trong lĩnh vực này. Cảm hứng từ những ứng dụng của Châu Âu, các công ty của Nhật bắt đầu dùng logic mờ vào kỹ thuật điều khiển từ năm 1980. Nhưng do các phần cứng chuẩn tính toán theo giải thuật logic mờ rất kém nên hầu hết các ứng dụng đều dùng các phần cứng chuyên về logic mờ. Một trong những ứng dụng dùng logic mờ đầu tiên tại đây là nhà máy xử lý nước của Fuji Electric vào năm 1983, hệ thống xe điện ngầm của Hitachi vào năm 1987. Những thành công đầu tiên đã tạo ra nhiều quan tâm ở Nhật. Có nhiều lý do để giải thích tại sao logic mờ được ưa chuộng. Thứ nhất, các kỹ sư Nhật thường bắt đầu từ những giải pháp đơn giản, sau đó mới đi sâu vào vấn đề. Phù hợp với việc logic mờ cho phép tạo nhanh các bản mẫu rồi tiến đến việc tối ưu. Thứ hai, các hệ dùng logic mờ đơn giản và dễ hiểu. Sự “thông minh” của hệ không nằm trong các hệ phương trình vi phân hay mã nguồn. Cũng như việc các kỹ sư Nhật thường làm việc theo tổ, đòi hỏi phải có một giải pháp để mọi người trong tổ đều hiểu được hành vi của hệ thống, cùng chia sẽ ý tưởng để tạo ra hệ. Logic mờ cung cấp cho họ một phương tiện rất minh bạch để thiết kế hệ thống. Và cũng do nền văn hóa, người Nhật không quan tâm đến logic Boolean hay logic mờ; cũng như trong tiếng Nhật , từ “mờ‟ không mang nghĩa tiêu cực. Do đó, logic mờ được dùng nhiều trong các ứng dụng thuộc lĩnh vực điều khiển thông minh hay xử lý dữ liệu. Máy quay phim và máy chụp hình dùng logic mờ để chứa đựng sự chuyên môn của người nghệ sĩ nhiếp ảnh. Mitsubishi thông báo về chiếc xe đầu tiên trên thế giới dùng logic mờ trong điều khiển, cũng như nhiều hãng chế tạo xe khác của Nhật dùng logic mờ trong một số thành phần. Trong lĩnh vực tự động hóa, Omron Corp. có khoảng 350 bằng phát minh về logic mờ. Ngoài ra, logic mờ cũng được dùng để tối ưu nhiều quá trình hóa học và sinh học.

## 2. Điều khiển mờ

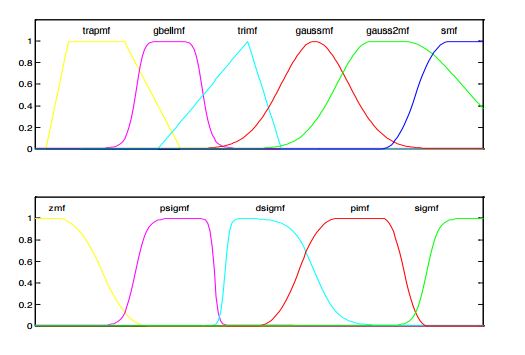
### 2.1. Định nghĩa tập mờ

Tập mờ F xác định trên tập kinh điển B là một tập mà mỗi phần tử của nó là một cặp giá trị (x,µF(x)), với x∈ X và µF(x) là một ánh xạ : µF(x) : B → [0 1] Điều khiển mờ Trang 250 trong đó : µF gọi là hàm thuộc , B gọi là tập nền.



### 2.2 Các thuật ngữ trong logic mờ

* Độ cao tập mờ F là giá trị h = SupµF(x), trong đó supµF(x) chỉ giá trị nhỏ nhất trong tất cả các chặn trên của hàm µF(x).
* Miền xác định của tập mờ F, ký hiệu là S là tập con thoả mãn : S = SuppµF(x) = { x∈B | µF(x) > 0}
* Miền tin cậy của tập mờ F, ký hiệu là T là tập con thoả mãn : T = { x∈B | µF(x) = 1 }
* Các dạng hàm thuộc (membership function) trong logic mờ. Có rất nhiều dạng hàm thuộc như : Gaussian, PI-shape, S-shape, Sigmoidal, Z-shape …



### 2.3 Biến ngôn ngữ

Biến ngôn ngữ là phần tử chủ đạo trong các hệ thống dùng logic mờ. Ở đây các thành phần ngôn ngữ của cùng một ngữ cảnh được kết hợp lại với nhau. Để minh hoạ về hàm thuộc và biến ngôn ngữ ta xét ví dụ sau : Xét độ ẩm đo được , ta có thể phát biểu: - Rất cao - Cao - Trung bình - Thấp - Rất thấp . Những phát biểu như vậy gọi là biến ngôn ngữ của tập mờ.

**Các phép toán trên tập mờ**

Cho X, Y là hai tập mờ trên không gian nền B, có các hàm thuộc tương ứng là µX, µY. Khi đó: - Phép hợp hai tập mờ: X∪Y

+ Theo luật Max µX∪Y(b) = Max{ µX(b) , µY(b) }

+ Theo luật Sum µX∪Y(b) = Min{ 1, µX(b) + µY(b) }

+ Tổng trực tiếp µX∪Y(b) = µX(b) + µY(b) - µX(b).µY (b)

- Phép giao hai tập mờ: X∩Y

+ Theo luật Min µX∪Y(b) = Min{ µX(b) , µY(b) }

+ Theo luật Lukasiewicz µX∪Y(b) = Max{0, µX(b)+µY(b)-1}

+ Theo luật Prod µX∪Y(b) = µX(b).µY(b)

- Phép bù tập mờ: X µ (b) = 1- µX(b)

**Luật hợp thành**

Mệnh đề hợp thành

Ví dụ điều khiển mực nước trong bồn chứa, ta quan tâm đến 2 yếu tố:

+ Mực nước trong bồn L = {rất thấp, thấp, vừa}

+ Góc mở van ống dẫn G = {đóng, nhỏ, lớn}

Ta có thể suy diễn cách thức điều khiển như thế này:

Nếu mực nước = rất thấp Thì góc mở van = lớn

Nếu mực nước = thấp Thì góc mở van = nhỏ

Nếu mực nước = vừa Thì góc mở van = đóng

Trong ví dụ trên ta thấy có cấu trúc chung là “Nếu A thì B”. Cấu trúc này gọi là mệnh đề hợp thành, A là mệnh đề điều kiện, C = A⇒B là mệnh đề kết luận.

Định lý Mamdani: “Độ phụ thuộc của kết luận không được lớn hơn độ phụ thuộc điều kiện” Nếu hệ thống có nhiều đầu vào và nhiều đầu ra thì mệnh đề suy diễn có dạng tổng quát như sau: If N = ni and M = mi and … Then R = ri and K = ki and ….

**Luật hợp thành mờ**

Luật hợp thành là tên gọi chung của mô hình biểu diễn một hay nhiều hàm thuộc cho một hay nhiều mệnh đề hợp thành.

Các luật hợp thành cơ bản Điều khiển mờ

+ Luật Max – Min

+ Luật Max – Prod

+ Luật Sum – Min

+ Luật Sum – Prod

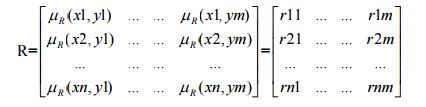
Thuật toán xây dựng mệnh đề hợp thành cho hệ SISO

Luật mờ cho hệ SISO có dạng “If A Then B”

Chia hàm thuộc µA(x) thành n điểm xi, i=1,2….,n

Chia hàm thuộc µB(y) thành m điểm yj , j = 1,2,…,m

Xây dựng ma trận quan hệ R



Hàm thuộc µB’(y) đầu ra ứng với giá trị rõ đầu vào xk có giá trị µB’(y) = a T .R , với a T = { 0,0,0,…,0,1,0….,0,0 }. Số 1 ứng với vị trí thứ k.

Trong trường hợp đầu vào là giá trị mờ A’ thì µB’(y) là: µB’(y) = { l1,l2,l3,…,lm } với lk=maxmin{ai,rik }.

### 2.4 Giải mờ

Giải mờ là quá trình xác định giá trị rõ ở đầu ra từ hàm thuộc µB’(y) của tập mờ B’. Có 2 phương pháp giải mờ :

**Phương pháp cực đại**

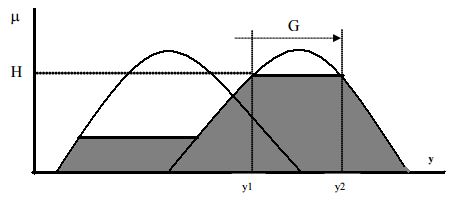
Các bước thực hiện : - Xác định miền chứa giá trị y’, y’ là giá trị mà tại đó µB’(y) đạt Max G = { y∈Y | µB’(y) = H }

- Xác định y’ theo một trong 3 cách sau :

+ Nguyên lý trung bình

+ Nguyên lý cận trái

+ Nguyên lý cận phải



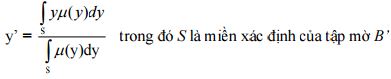
• Nguyên lý trung bình: y’ = 2 y1+ y2

• Nguyên lý cận trái : chọn y’ = y1

• Nguyên lý cận phải : chọn y’ = y2  
**Phương pháp trọng tâm**

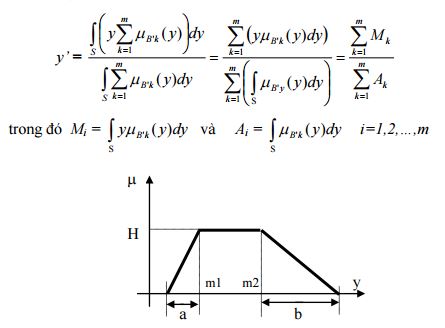
Điểm y’ được xác định là hoành độ của điểm trọng tâm miền được bao bởi trục hoành và đường µB’(y).

Công thức xác định :

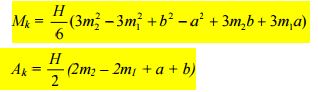


♦Phương pháp trọng tâm cho luật Sum-Min

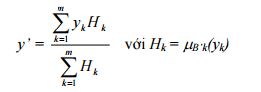
Giả sử có m luật điều khiển được triển khai, ký hiệu các giá trị mờ đầu ra của luật điều khiển thứ k là µB’k(y) thì với quy tắc Sum-Min hàm thuộc sẽ là µB’(y) =), và y’ được xác định :



Xét riêng cho trường hợp các hàm thuộc dạng hình thang như hình trên :



♦ Phương pháp độ cao



# CHƯƠNG III : TRIỂN KHAI LẮP RÁP HỆ THỐNG

## 1. Ý tưởng và chức năng hệ thống

Hệ thống thực hiện việc tưới cây tự động dựa trên thời gian và độ ẩm của đất. Vào một thời gian nào đó trong ngày, ví dụ như vào 18h tối, hệ thống sẽ đo độ ẩm của đất từ đó tính toán lượng nước phù hợp và tiến hành bơm nước. Hệ thống bao gồm hai thành phần: bộ phát và bộ điều khiển. Bộ phát làm nhiệm vụ đo độ ẩm của đất, thu lại tín hiệu, truyền dữ liệu đo cho bộ điều khiển thông qua sóng RF . Bộ điều khiển khi đến thời gian tưới, sẽ lấy số liệu đo từ bộ phát , tính toán lượng nước cần bơm , sau đó bật máy bơm để bơm nước . Trong quá trình bơm , cảm biến lưu lượng cho biết lượng nước đã tưới. Khi lượng nước này bằng lượng nước cần bơm thì bộ điều khiển tắt máy bơm. Người dùng có thể chỉnh thời gian tưới và mối quan hệ giữa lượng nước tưới với độ ẩm của đất để phù hợp với từng loại cây trồng và đất.

## 2. Cách thức tính toán lượng nước cần bơm

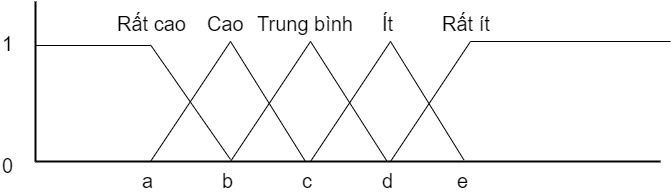
Lượng nước cần bơm là lượng nước cần tưới trên 1 kg đất được tính toán dựa trên lý thuyết điều khiển mờ

### 2.1. Mờ hóa

Giá trị đo của độ ẩm được mờ hóa thành độ thuộc của các tập mờ tương ứng với độ ẩm .

Có 5 tập mờ độ ẩm: rất cao, cao, trung bình , thấp , rất thấp với các hàm độ thuộc như sau:

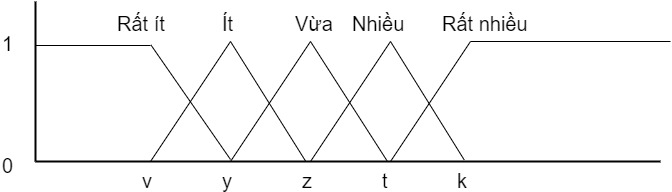
Với a,b,c,d,e là các tham số do người dùng quyết định



Hình 5. Tập mờ độ ẩm

Có 5 tập mờ của lượng nước đầu ra: rất nhiều, nhiều, trung bình, ít, rất ít

Với v,y,z,t,k là các tham số do người dùng quyết định



Hình 6. Tập mờ lượng nước cần bơm

### 2.2. Luật hợp thành mờ

Hệ thống chỉ có một đầu vào và một đầu ra ( SISO) nên luật hợp thành mờ như sau:

* Nếu độ ẩm rất cao thì lượng nước tưới rất ít
* Nếu độ ẩm cao thì lượng nước tưới ít
* Nếu độ ẩm trung bình thì lượng nước tưới trung bình
* Nếu độ ẩm thấp thì lượng nước tưới nhiều
* Nếu độ ẩm rất thấp thì lượng nước tưới rất nhiều

### 2.3. Giải mờ

Sử dụng phương pháp trọng tâm để xác định đầu ra

## 3.Sơ đồ hệ thống

Hình 7. Sơ đồ hệ thống

## 4. Lắp ráp các thành phần

### 4.1 Bộ phát dữ liệu

+ Sơ đồ đi dây giữa Arduino và module RF

|  |  |  |
| --- | --- | --- |
| Signal | Color | RF24 |
| GND | Brown | GND |
| VCC | Red | 5V |
| CE | Orange | 7 |
| CSN | Yellow | 8 |
| SCK | Green | 13 |
| MOSI | Blue | 11 |
| MISO | Violet | 12 |
| IRQ | Gray | - |

+ Sơ đồ đi dây giữa Arduino và cảm biến độ ẩm đất

|  |  |
| --- | --- |
| Signal | Cảm biến độ ẩm đất |
| A0 | A0 |
| - | D0 |
| GND | GND |
| 5V | VCC |

+ Lập trình

#include <SPI.h>

#include "RF24.h"

// Xac dinh radio la radio phat

bool radioNumber = 1;

/\* Hardware configuration: Set up nRF24L01 radio on SPI bus plus pins 7 & 8 \*/

RF24 radio(7,8);

byte addresses[][6] = {"1Node","2Node"};

// Used to control whether this node is sending or receiving

bool role = 0;

void setup() {

Serial.begin(115200);

Serial.println(F("CAM BIEN DO AM + RADIO PHAT"));

Serial.println(F("\*\*\* AN 'T' DE BAT DAU QUA TRINH CHUYEN GIU LIEU \*\*\* "));

pinMode (2, INPUT);

pinMode(A0, INPUT);

radio.begin();

// Set the PA Level low to prevent power supply related issues since this is a

// getting\_started sketch, and the likelihood of close proximity of the devices. RF24\_PA\_MAX is default.

radio.setPALevel(RF24\_PA\_LOW);

// Open a writing and reading pipe on each radio, with opposite addresses

if(radioNumber){

radio.openWritingPipe(addresses[1]);

radio.openReadingPipe(1,addresses[0]);

}else{

radio.openWritingPipe(addresses[0]);

radio.openReadingPipe(1,addresses[1]);

}

// Start the radio listening for data

radio.startListening();

}

void loop() {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Ping Out Role \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if (role == 1) {

radio.stopListening(); // First, stop listening so we can talk.

long value = analogRead(A0); // Ta sẽ đọc giá trị hiệu điện thế của cảm biến

// Giá trị được số hóa thành 1 số nguyên có giá trị

// trong khoảng từ 0 đến 1023

Serial.print("Do am do duoc tren cay la: ");

Serial.println(value);//Xuất ra serial Monitor

delay(10);

Serial.println(F("Bat dau gui"));

unsigned long start\_time = value; // Take the time, and send it. This will block until complete

if (!radio.write( &start\_time, sizeof(unsigned long) )){

Serial.println(F("LOI"));

}

radio.startListening(); // Now, continue listening

unsigned long started\_waiting\_at = micros(); // Set up a timeout period, get the current microseconds

boolean timeout = false; // Set up a variable to indicate if a response was received or not

while ( ! radio.available() ){ // While nothing is received

if (micros() - started\_waiting\_at > 200000 ){ // If waited longer than 200ms, indicate timeout and exit while loop

timeout = true;

break;

}

}

if ( timeout ){ // Describe the results

Serial.println(F("LOI , KHONG KET NOI DUOC VOI RADIO THU."));

}else{

unsigned long got\_time; // Grab the response, compare, and send to debugging spew

radio.read( &got\_time, sizeof(unsigned long) );

unsigned long end\_time = micros();

// Spew it

Serial.print(F("Sent "));

Serial.print(start\_time);

Serial.print(F(", Got response "));

Serial.print(got\_time);

Serial.print(F(", Round-trip delay "));

Serial.print(end\_time-start\_time);

Serial.println(F(" microseconds"));

}

// Try again 1s later

delay(1000);

}

if ( Serial.available() )

{

char c = toupper(Serial.read());

if ( c == 'T' && role == 0 ){

Serial.println(F("\*\*\* CHANGING TO TRANSMIT ROLE -- PRESS 'R' TO SWITCH BACK"));

role = 1; // Become the primary transmitter (ping out)

}else

if ( c == 'R' && role == 1 ){

Serial.println(F("\*\*\* CHANGING TO RECEIVE ROLE -- PRESS 'T' TO SWITCH BACK"));

role = 0; // Become the primary receiver (pong back)

radio.startListening();

}

}

}

### 4.2.Bộ điều khiển

+ Sơ đồ đi dây giữa Arduino và RF

|  |  |  |
| --- | --- | --- |
| Signal | Color | RF24 |
| GND | Brown | GND |
| VCC | Red | 5V |
| CE | Orange | 7 |
| CSN | Yellow | 8 |
| SCK | Green | 13 |
| MOSI | Blue | 11 |
| MISO | Violet | 12 |
| IRQ | Gray | - |

+ Sơ đồ đi dây giữa Arduino và cảm biến lưu lượng

|  |  |
| --- | --- |
| Arduino | Cảm biến lưu lượng |
| 2 | Tín hiệu |

+ Sơ đồ đi dây giữa Arduino và máy bơm ( thông qua relay )

Arduino qua Relay

|  |  |
| --- | --- |
| Arduino | Relay |
| 4 | Tín hiệu |
| Dương | VCC |
| Dương | VSS+ |
| Âm | GND |
| Âm | VSS- |

Relay qua máy bơm

|  |  |
| --- | --- |
| Relay | Máy Bơm |
| NC | Âm |
| COM | Dương |

+ Lập trình

#include <SPI.h>

#include "RF24.h"

//set lam radio thu

bool radioNumber = 0;

RF24 radio(7,8);

byte addresses[][6] = {"1Node","2Node"};

//dia chi cho radio thuu

bool role = 0;

//set chuong trinh cho cam bien Hall

int pushButton = 2;

//cac tham so cua do am

const float a=200;

const float b=400;

const float c=600,d=800,e=1000;

//cac tham so cua luong nuoc

const float v=200,y=400,z=600,t=800,k=1000;

float luongnuoc;

int st;

int dem;

float tb;

int solanthaydoitrangthai=0;

int trangthaihientai;

int trangthaitruoc;

void setup() {

Serial.begin(115200);

Serial.println(F("CAM BIEN DO AM + RADIO THU"));

pinMode(pushButton,INPUT);

radio.begin();

radio.setPALevel(RF24\_PA\_LOW);

// Open a writing and reading pipe on each radio, with opposite addresses

if(radioNumber){

radio.openWritingPipe(addresses[1]);

radio.openReadingPipe(1,addresses[0]);

}else{

radio.openWritingPipe(addresses[0]);

radio.openReadingPipe(1,addresses[1]);

}

radio.startListening();

st=0; //trang thai doi

dem=0;

tb=0;

//van

pinMode (2,OUTPUT);

//cambienHALL

pinMode (3,INPUT);

}

float ratcao(float x,float a, float b,float c,float d,float e){

if(0<x && x<=a){

return 1.0;

} else if(a<x && x<b){

return (b-x)/(b-a);

} else {

return 0.0;

}

}

float cao(float x,float a, float b,float c,float d,float e){

if(x<=a){

return 0.0;

} else if(a<x && x<=b){

return (x-a)/(b-a);

}else if(b<x && x<c){

return (c-x)/(c-b);

}else {

return 0.0;

}

}

float trungbinh(float x,float a, float b,float c,float d,float e){

if(x<=b){

return 0.0;

}else if(b<x && x<=c){

return (x-b)/(c-b);

}else if(c<x && x<d){

return (d-x)/(d-c);

}else {

return 0.0;

}

}

float thap(float x,float a, float b,float c,float d,float e){

if(x<=c){

return 0.0;

}else if(c<x && x<=d){

return (x-c)/(d-c);

}else if(d<x && x<e){

return (e-x)/(e-d);

}else {

return 0.0;

}

}

float ratthap(float x,float a, float b,float c,float d,float e){

if(x<=d){

return 0.0;

} else if(d<x && x<e){

return (x-d)/(e-d);

} else {

return 1.0;

}

}

void loop() {

//kiem tra gio de chuyen st=0 sang st=1

st=1;

if(st==1){

float x;

if( radio.available()){

while (radio.available()) {

radio.read( &x, sizeof(float) );

}

dem++;

tb+=x;

if( dem>=10 ){

st=2;

dem=0;

tb=tb/10;

}

}

if(st==2){

//radio.stopListening();

//radio.write( &x, sizeof(float) );

//radio.startListening();

Serial.print(F("Do am duoc nhan la : "));

Serial.println(tb);

Serial.println("--------------------\n");

Serial.println("rat cao = ");Serial.print(ratcao(tb,a,b,c,d,e));

Serial.println("cao = ");Serial.print(cao(tb,a,b,c,d,e));

Serial.println("trung binh = ");Serial.print(trungbinh(tb,a,b,c,d,e));

Serial.println("thap = ");Serial.print(thap(tb,a,b,c,d,e));

Serial.println("rat thap = ");Serial.print(ratthap(tb,a,b,c,d,e));

float ratit,it,vua,nhieu,ratnhieu;

ratit=ratcao(tb,a,b,c,d,e);

it=cao(tb,a,b,c,d,e);

vua=trungbinh(tb,a,b,c,d,e);

nhieu=thap(tb,a,b,c,d,e);

ratnhieu=ratthap(tb,a,b,c,d,e);

luongnuoc=(v\*ratit+y\*it+z\*vua+t\*nhieu+k\*ratnhieu)/(ratit+it+vua+nhieu+ratnhieu);

Serial.println("luongnuoc= ");Serial.print(luongnuoc);

st=3;

tb=0;

}

if(st==3){

digitalWrite(2,HIGH); //relay

int buttonState = digitalRead(pushButton);

//Serial.print(buttonState);

float f;

f=luongnuoc\*450;

trangthaihientai = buttonState;

if (trangthaihientai != trangthaitruoc )

{

solanthaydoitrangthai++;

//Serial.println("So lan thay doi trang thai la: ");

//Serial.println(solanthaydoitrangthai);

}

trangthaitruoc=trangthaihientai;

if(solanthaydoitrangthai>=f){

st=0;

f=0;

digitalWrite(2,LOW);

}

}

}

}

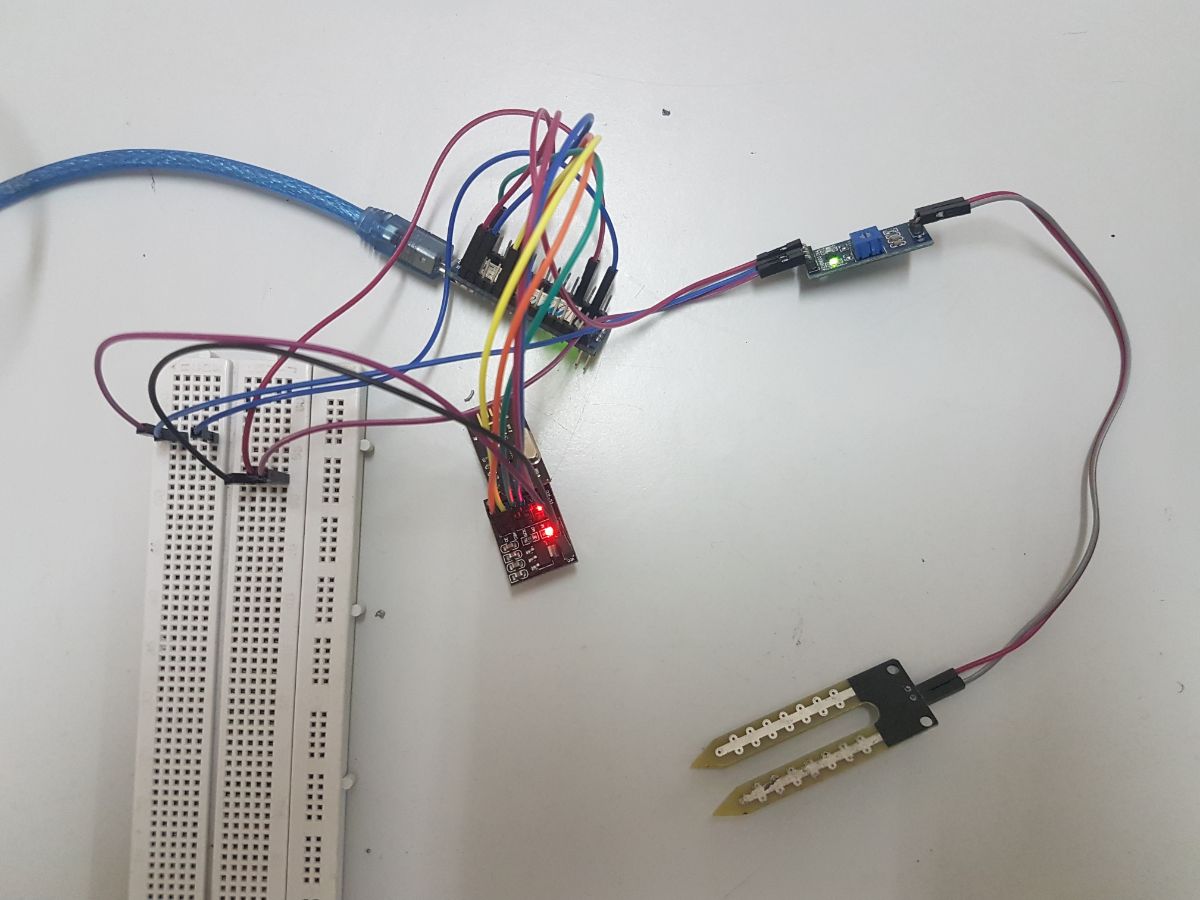
# CHƯƠNG IV : THỬ NGHIỆM SẢN PHẨM

## 1. Điều kiện thử nghiệm:

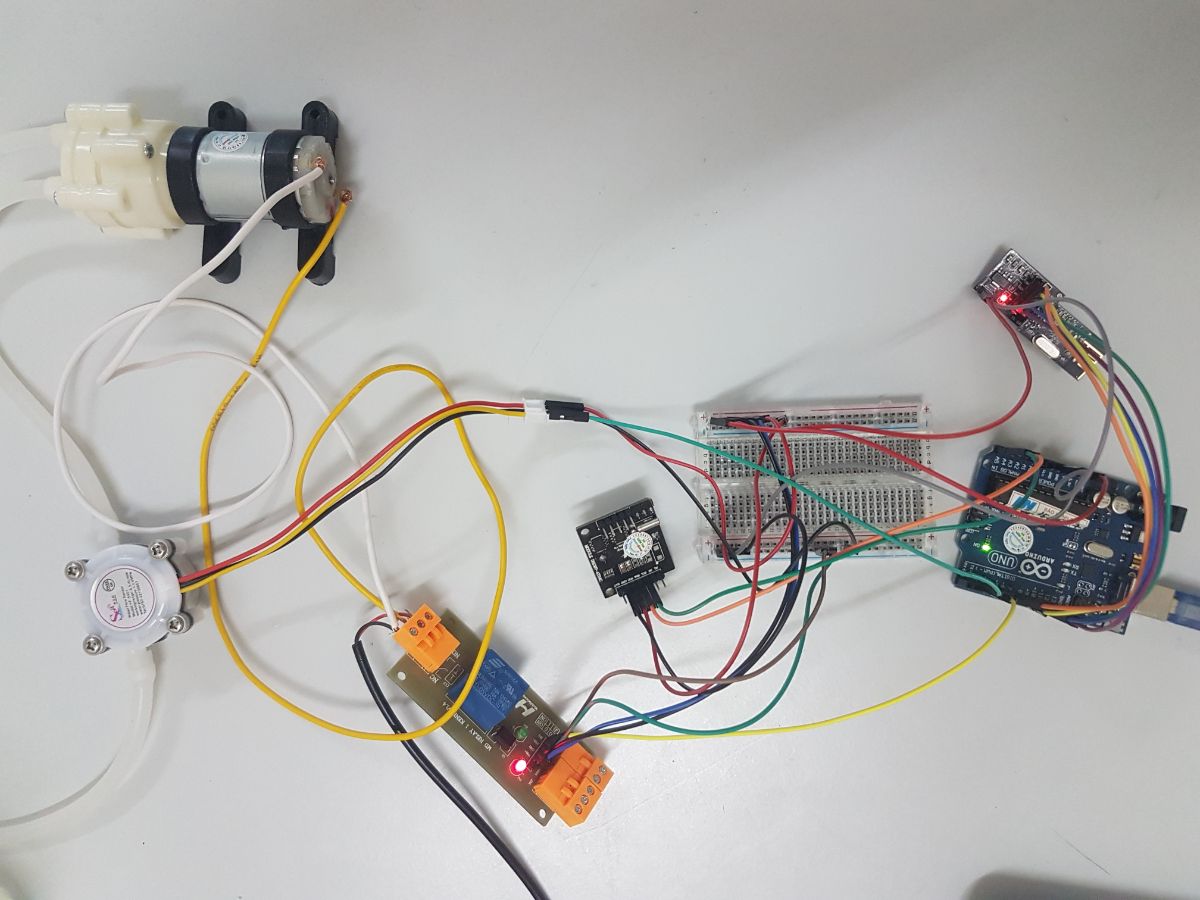
a) Nơi thử nghiệm: Trong phòng thí nghiệm

b) Thiết bị :

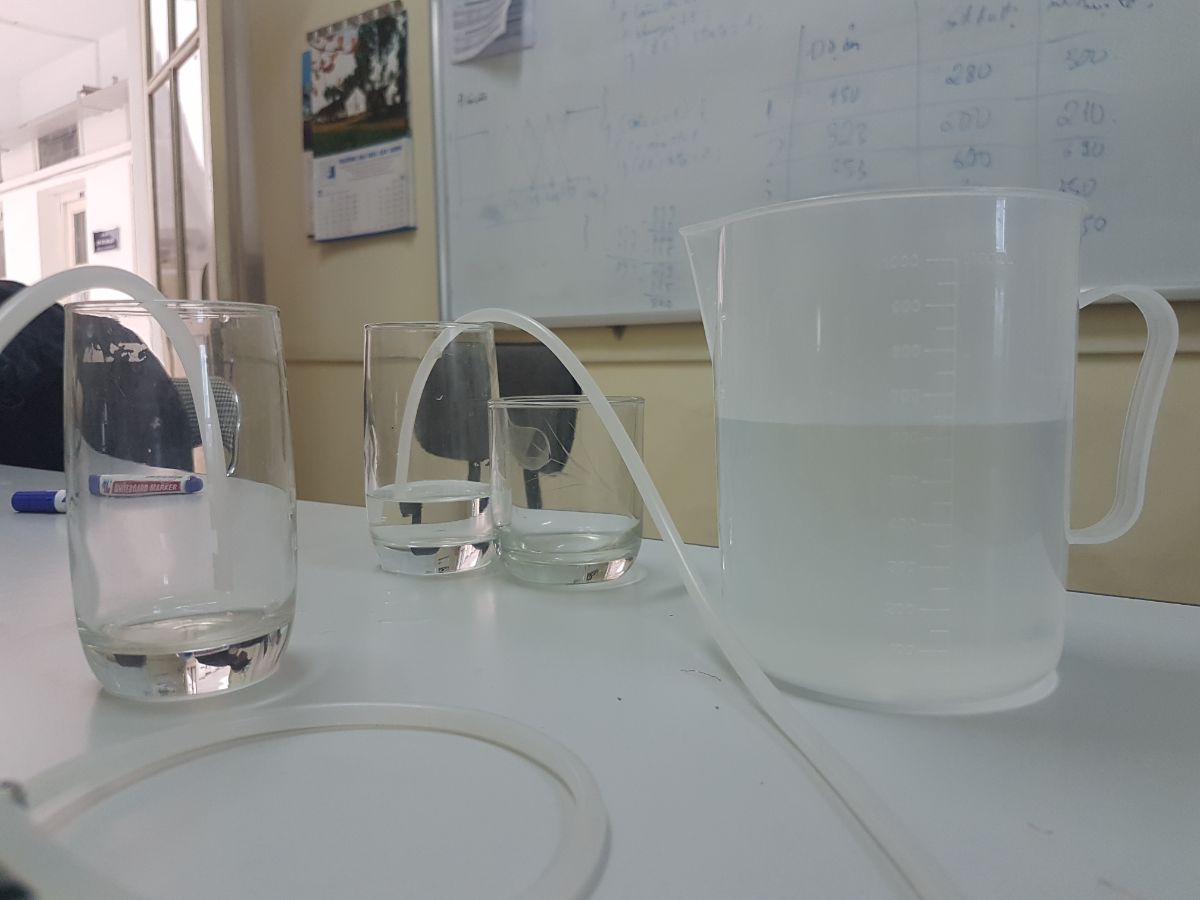
* Bộ thu
  + Máy bơm MB385
  + Relay 5V
  + Arduino UNO
  + Bộ phát sóng RF
  + Module RTC
  + Cảm biến lưu lượng S401
* Bộ phát
  + Arduino Nano
  + Cảm biến độ ẩm đất
  + Bộ phát sóng RF
* Bình nước 1l , có chia độ
* Dây dẫn
* Ống nước



Hình 8. Bộ đo đạc và phát tín hiệu



Hình 9. Bộ thu và điều khiển



Hình 10. Thiết bị thí nghiệm

## 2. Kết quả

Hệ thống được thử nghiệm 5 lần đo với các mức độ ẩm khác nhau.

Thông số đầu vào :

- Lượng đất : 1 kg

- Các giá trị tập mờ tương ứng với giá trị đo được:

a = 400, b = 524, c = 649, d = 774, e= 899

- Các giá trị tập mờ tương ứng với lượng nước cần bơm:

v=200, y=400 , z=600 , t=800 , k=1000

Ta có được lượng nước tính toán và lượng nước thực tế thu được như sau:

Lưu ý: Lượng nước tính toán là lượng nước được tính khi tưới cho 1 kg đất

|  |  |  |  |
| --- | --- | --- | --- |
| Số lần đo | Độ ẩm | Lượng nước tính toán(ml) | Lượng nước thực tế (ml) |
| 1 | 450 | 280 | 300 |
| 2 | 323 | 200 | 210 |
| 3 | 653 | 600 | 690 |
| 4 | 482 | 333 | 350 |
| 5 | 734 | 736 | 650 |

## 3. Phân tích và kết luận

- Kết quả lượng nước tính toán là đúng với độ ẩm đo được

VD: độ ẩm 450 => . Ta có tập mờ đầu ra theo luật hợp thành mờ là .

- Lượng nước thực tế có chênh lệch với lượng nước tính toán do sai số của thiết bị đo và khoảng thời gian lấy mẫu. VD: thời gian cập nhật kết quả đo lưu lượng là 500ms lượng nước bơm có thể vượt so với lượng nước tính toán do chưa cập nhật kịp kết quả đo.

# CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 1. Kết luận

Qua 6 tháng tích cực nghiên cứu, chúng em đã có kết quả như sau :

- Về lý thuyết : nghiên cứu , thử nghiệm và áp dụng các phần ứng như Arduino và các loại cảm biến ; tìm hiểu về điều khiển mờ và các ứng dụng của nó

- Về sản phẩm: phát triển hệ thống tưới cây tự động ứng dụng điều khiển mờ. Hệ thống đã hoạt động ở dạng demo trong phòng thí nghiệm.

## 2. Phát triển

-Tiếp tục phát triển hệ thống để hỗ trợ thêm nhiều loại cảm biến khác nhau , quy mô lớn hơn.

- Đóng gói, hoàn thiện sản phẩm để có thể triển khai trên thực tế

# Tài liệu tham khảo

* Các trang web giới thiệu hướng dẫn các thiết bị phần cứng
* https://vi.wikipedia.org/wiki/Logic\_m%E1%BB%9D
* http://www.vnua.edu.vn/khoa/fita/wp-content/uploads/2013/06/F\_Ch.1.pdf
* hachi.com.vn%2Ftrong-rau-bang-smartphone%2F&h=ATMyvoIcIM0LuLfN0nyq-knI0T426ldlQ2pLZotoMn2jwN8ntk6ihlYNJTKqnvFBIqyDps1YdO8F2jAeUM3UltXGBEDsS6KLIuj49WQnniQX6nd3w0B9lupnHxLIJqSLxxJt9YsL
* khoahoc.tv%2F12-cach-nguoi-israel-thay-doi-nen-nong-nghiep-the-gioi-p1-56789&h=ATMyvoIcIM0LuLfN0nyq-knI0T426ldlQ2pLZotoMn2jwN8ntk6ihlYNJTKqnvFBIqyDps1YdO8F2jAeUM3UltXGBEDsS6KLIuj49WQnniQX6nd3w0B9lupnHxLIJqSLxxJt9YsL