

MỤC LỤC

LỜI NÓI ĐẦU	1
CHƯƠNG 1. GIỚI THIỆU TỔNG QUÁT VỀ TƯỚI TIÊU TỰ ĐỘNG TRONG NÔNG NGHIỆP	2
1.1. KHÁI QUÁT VỀ NHÀ TRỒNG THÔNG MINH.....	2
1.2. THỰC TIỄN ỨNG DỤNG CỦA ARDUINO VÀO ĐỀ TÀI.....	3
1.3. KHÁI NIỆM, NHIỆM VỤ CỦA HỆ THỐNG TƯỚI NƯỚC TỰ ĐỘNG.....	5
1.3.1. Tổng quan về thiết kế một hệ thống tưới.....	5
1.3.2. Thiết kế hệ thống tưới.....	5
CHƯƠNG 2.CẤU TRÚC VÀ TÍNH NĂNG CỦA ARDUINO	8
2.1. GIỚI THIỆU CHUNG VỀ ARDUINO.....	8
2.2. BOARD ARDUINOMEGA 2560	9
2.3. ARDUINO LCD KEYPAD SHIELD.	16
2.4. CẢM BIẾN NHIỆT ĐỘ, ĐỘ ẨM DHT 21.	17
2.5. MODULE CẢM BIẾN MƯA (SENSRAIN).....	18
2.6. GIỚI THIỆU VỀ MODULE RELAY 12VDC.	20
2.7. MODULE THỜI GIAN THỰC.	22
2.8. ĐỘNG CƠ MỘT CHIỀU 12VDC.....	23
2.9. NGUỒN TỎ ONG 12VDC.	24
2.10. PHẦN MỀM MÔ PHỎNG PROTEUS.....	25
2.11. THƯ VIỆN ARDUINO TRONG PROTEUS.	26
2.12. ARDUINO IDE VÀ LẬP TRÌNH CHO ARDUINO.	27
CHƯƠNG 3.THIẾT KẾ, LẬP TRÌNH, LẮP ĐẶT MẠCH THỰC TẾ	29
3.1. GIỚI THIỆU CHUNG CỦA HỆ THỐNG.....	29
3.2. THIẾT KẾ MẠCH TRÊN PROTEUS.	30
3.3. LẬP TRÌNH HIỂN THỊ LCD.	31
3.3.1. Lập trình đo nhiệt độ, độ ẩm.....	31
3.3.2. Lập trình trạng thái cảm biến mưa.....	34

3.3.3. Lập trình cho module DS3231(DS1307).....	34
3.3.4. Lập trình hiển thị giá trị lên LCD 16.2	35
3.4. LẬP TRÌNH HỆ THỐNG CÀI ĐẶT THỜI GIAN, NHIỆT ĐỘ, ĐỘ ÂM, CẢM BIẾN MƯA ĐIỀU KHIỂN ĐÓNG CẮT ĐỘNG CƠ.....	37
3.4.1. Thiết kế hệ thống điều khiển đóng cắt động cơ.....	37
3.4.2. Tổng thể mạch đang hoạt động khi đã cấp nguồn và nạp Code.	39
Kết luận.....	53
TÀI LIỆU THAM KHẢO.....	57

LỜI NÓI ĐẦU

Ngày nay khoa học công nghệ ngày càng phát triển, vi điều khiển AVR và vi điều khiển PIC ngày càng thông dụng và hoàn thiện hơn , nhưng có thể nói sự xuất hiện của Arduino vào năm 2005 tại Italia đã mở ra một hướng đi mới cho vi điều khiển. Sự xuất hiện của Arduino đã hỗ trợ cho con người rất nhiều trong lập trình và thiết kế, nhất là đối với những người bắt đầu tìm tòi về vi điều khiển mà không có quá nhiều kiến thức, hiểu biết sâu sắc về vật lý và điện tử . Phần cứng của thiết bị đã được tích hợp nhiều chức năng cơ bản và là mã nguồn mở. Ngôn ngữ lập trình trên nền Java lại vô cùng dễ sử dụng tương thích với ngôn ngữ C và hệ thư viện rất phong phú và được chia sẻ miễn phí. Chính vì những lý do như vậy nên Arduino hiện đang dần phổ biến và được phát triển ngày càng mạnh mẽ trên toàn thế giới.

Trên cơ sở kiến thức đã học trong môn học : Tin học đại cương , Điện tử tương tự và số... cùng với những hiểu biết về các thiết bị điện tử, em đã quyết định thực hiện đề tài :**Thiết kế hệ thống tưới cây tự động** do GS. TSKH Thân Ngọc Hoàn hướng dẫn. Đề tài gồm các nội dung sau:

- Chương 1. Giới thiệu tổng quan về tưới tiêu tự động trong nông nghiệp.
- Chương 2. Cấu trúc và tính năng của arduino.
- Chương 3. Thiết kế, lập trình, lắp đặt mạch thực tế

CHƯƠNG 1.

GIỚI THIỆU TỔNG QUÁT VỀ TƯỚI TIÊU TỰ ĐỘNG TRONG NÔNG NGHIỆP

1.1.KHÁI QUÁT VỀ NHÀ TRỒNG THÔNG MINH.

Tại một số địa phương đã canh tác một số loại cây, hoa, rau có giá trị kinh tế cao, tuy nhiên hiện nay vẫn còn ít các đơn vị nào tiến hành nghiên cứu thiết kế các mô hình tự động đáp ứng điều kiện kinh tế, môi trường của nước ta.

Thực tế trong cuộc sống ngày càng bận rộn, nhiều người vẫn có thú vui là trồng những cây cảnh, vườn rau trong không gian trồng của nhà mình như sân thượng, ban công. Tuy nhiên, trong những lúc bận rộn các công việc hằng ngày thì những cây cảnh và vườn hoa ở nhà sẽ không ai tưới nước.



Hình 1.1: Hệ thống tưới nước tự động cho cây trồng.

Ngoài phương pháp tưới thủ công, các phương pháp tưới còn lại đều cần được thiết kế, tính toán các thông số cho phù hợp. Nhìn chung, các phương pháp tưới có dùng ống đều có cùng nguyên lý tính toán trên cơ sở môn học cấp thoát nước và thủy lực đường ống, đó là: xác định diện tích tưới, nguồn nước, nhu cầu nước tưới phù hợp với từng loại cây trồng, diện tích, địa hình vùng tưới. Từ các thông số này, ta sẽ tính toán đường kính ống chính, ống phụ, ống nhánh, vận tốc nước chảy trong ống, áp lực nước trong ống; tính toán chiều dài của các loại ống, các chi tiết nối (co, tê, van, lơi vv...), số lượng cácбет phun, бет đế chân, ống dẫn đến gốc vv..và cuối cùng là lập bảng tổng hợp số lượng các loại vật tư, tính toán chiít mua vật tư, tiền công xây lắp vv..

1.2. THỰC TIỄN ÁP DỤNG CỦA ARDUINO VÀO ĐỀ TÀI.

Ứng dụng của Arduino về mô hình hệ thống tự động. Sử dụng một cảm biến độ ẩm và nhiệt độ kết nối với một Arduino và điều khiển động cơ tạo ra hệ thống tưới tiêu tự động. Tất cả mọi việc đều tự động diễn ra trong quá trình cài đặt sẵn và qua các cảm biến để điều tiết việc tưới cây hợp lí trong mọi thời tiết. Từ ví dụ thực tiễn cùng với sự giúp đỡ của giáo viên hướng dẫn, em đã lựa chọn và phát triển đề tài theo hướng sử dụng kit Arduino để thực hiện đề tài của mình. Ứng dụng đơn giản qua thao tác nút bấm cài đặt thời gian hẹn giờ để tưới cây và làm việc thông minh qua các cảm biến. Trước hết về cảm biến nhiệt độ và độ ẩm được áp dụng với khí hậu thời tiết nhiệt đới ẩm gió mùa với 4 mùa rõ rệt vậy nên cảm biến nhiệt độ độ ẩm có tầm quan trọng trong khâu tự động. Đo được các mức nhiệt độ cần thiết đáp ứng cho cây trồng cũng là ưu tiên hàng đầu trong ứng dụng. Đối với mùa mưa thì cũng nên quan tâm nên em đã thêm phần thiết kế cảm biến mưa để tránh những lúc tưới tiêu không cần thiết vào mùa mưa gây ảnh hưởng cây trồng. Hệ thống đảm bảo về mảng thời gian cài đặt, đảm bảo về nhiệt độ thay đổi qua mùa, độ ẩm theo khí hậu và cảm biến mưa tất cả được kết hợp tạo thành hệ thống thông

minh đáp ứng đúng nghĩa với người trồng cây. Vì vậy việc tưới tiêu cây trồng với con người trong công nghệ này đã thay thế hoàn toàn con người. Một hệ thống trồng thông minh đáp ứng về thời tiết kết hợp việc cài đặt thời gian tưới phù hợp cây trồng tất cả chỉ trong một thiết bị vi điều khiển arduino. Phù hợp với người bận công việc, phù hợp với việc sản xuất nông nghiệp cao đưa ra chất lượng cây trồng tốt. Một số ứng dụng cụ thể áp dụng qua đề tài : Tưới phun sương cho cây lan, tưới phun sương trồng nấm, tưới phun tia mưa cho hoa cúc, Các hình ảnh ứng dụng thực tế nhà cây trồng hình 1.2.



Hình 1.2: Hệ thống phun sương tự động nhà trồng cây lan áp dụng arduino.

1.3. KHÁI NIỆM, NHIỆM VỤ CỦA HỆ THỐNG TƯỚI NƯỚC TỰ ĐỘNG.

Thiết kế hệ thống tưới cây tự động đòi hỏi phải có một số thông tin về các vật tư thiết bị, về bộ vi xử lý, các bộ cảm biến, bộ điều khiển đóng cắt...Vậy nên việc đặt ra bài toán thiết kế là rất cần thiết.

1.3.1.Tổng quan về thiết kế một hệ thống tưới.

Để thiết kế hệ thống tưới nước cho cây trồng, cần quan tâm đến các vấn đề sau:

- Hình dạng vùng tưới.
- Diện tích vùng tưới.
- Số cây cần cung cấp nước tưới.
- Nhu cầu nước của loại cây trồng/đơn vị thời gian (lít/ngày).
- Địa hình khu tưới.

Để xác định hình dáng, diện tích vùng tưới, không có cách nào khác hơn là phải đo đạc. Khoảng cách giữa các cây cũng là yếu tố tạo nên chất lượng yêu cầu đề ra. Khoảng cách cũng tạo ra sự chênh lệch lượng nước tới nếu không đều chênh lệch cũng cao tạo sản phẩm kém. Với diện tích to quy hoạch lớn chúng ta nên đo chính xác vừa đủ với lưu lượng nước chúng ta cần tưới phù hợp với công suất bơm.

+ Khi đã có “cái nền” là hình dáng, diện tích ta bắt đầu phát họa sơ đồ bố trí cây trồng.

1.3.2 . Thiết kế hệ thống tưới.

a . Xác định lần tưới nhu cầu nước/lần tưới và khả năng cung cấp nước:

Tùy thuộc loại cây trồng, ta xác định lần tưới và nhu cầu nước cho mỗi lần tưới.Số lần tưới phụ thuộc vào đặc tính của loài cây trồng và khả năng giữ ẩm của đất.Ta chỉ cần tính toán gần đúng thông số về lần tưới dùng để tính toán nguồn nước.

Trong sản xuất, sẽ dựa vào thực tế đất đai, thời tiết để điều chỉnh số lần tưới cho phù hợp. Nhu cầu nước/lần tưới là thông số quan trọng để tính toán, thiết kế hệ thống tưới và tính toán nguồn nước. Chuyên ngành thủy lợi có bảng tra nhu cầu nước cho các loại cây trồng/vụ hoặc ngày hoặc có thể tra nghiên cứu trên mạng; tuy nhiên, người trồng cũng có nhiều kinh nghiệm thực tế để xác định nhu cầu nước tưới cho mỗi loại cây trồng. Trong thực tế, nhu cầu nước của cây trồng ít hơn nhiều so với lượng nước ta cung cấp; do vậy mà lượng nước tưới tùy thuộc vào phương pháp tưới. Thông thường nhu cầu nước tưới cho một cây cần tưới giao động từ 5-10 lít (tưới nhỏ giọt); 15-20 lít (tưới phun tia) 30 đến 40 lít nước (tưới rãnh, tưới phun mưa).

Từ xác định được nhu cầu nước của cây cho mỗi lần tưới, số lần tưới/tháng, số tháng cần tưới, ta xác định được nhu cầu nguồn nước tưới.

b) Phân chia khu tưới:

Nếu bạn chỉ tưới cho diện tích nhỏ trở lại thì chỉ là 1 khu tưới; nhưng nếu diện tích tưới lớn hơn phải phân chia vùng tưới thành nhiều khu tưới. Nếu chỉ khu tưới quá lớn, công suất máy bơm và đường kính ống dẫn nước chính sẽ tăng lên rất lớn dẫn đến không có hiệu quả kinh tế.

Khi phân chia khu tưới, bạn phải lên bản vẽ thể hiện rõ hình dáng, diện tích từng khu tưới, kích thước các cạnh của khu tưới, vẽ các hàng cây và chiều dài mỗi hàng cây, từ đây ta sẽ tính được số lượng cây trồng trong mỗi khu tưới, tính ra đường kính, chiều dài của đường ống chính.

c - Tính toán đường ống chính:

Đường ống chính tải nước tưới đến từng khu tưới và cho cả vùng tưới, do đó, ta phải tính toán được chiều dài và đường kính ống phù hợp và cả áp lực để chọn loại ống phù hợp (lớn quá sinh thừa – tốn tiền vô ích, ống nhỏ quá không cung cấp đủ nước cho khu tưới, ống kém dễ hỏng dẫn đến tốn kém...).

Ngoài ra, ta cần tính vị trí lắp đặt đường ống chính và chuyển nó lên bản vẽ. Thông thường nếu khu tưới có địa hình thấp dần thì ta bố trí đường ống chính đi theo cạnh có cao độ lớn nhất của khu tưới, nhờ đó khi xả nước ra khỏi đường ống chính, nước sẽ có khuynh hướng chảy từ nơi cao đến nơi thấp tới có lợi về năng lượng.

Nếu đất tương đối bằng phẳng hoặc gợn sóng nhỏ ở giữa đồng đất thì nên bố trí đường ống chạy dọc theo các đỉnh cao xuyên qua đồng đất để chia nước tưới về hai phía.

+ Tính chiều dài đường ống chính:

Dùng thước kẻ ly đo tổng chiều dài đường ống chính trên bản vẽ, nhân với tỷ lệ bản vẽ để xác định tổng chiều dài thực của đường ống chính.

+ Tính toán đường kính của đường ống chính:

Để tính toán được kích thước của đường ống chính ta cần xác định tổng nhu cầu nước tưới cho một lần tưới cho khu tưới lớn nhất của vùng tưới. Căn cứ vào chiều dài của mỗi hàng ta tính được số cây/hàng bằng công thức:

Số cây/hàng = chiều dài hàng chia cự ly trồng (cây cách cây) + 1

CHƯƠNG 2.

CẤU TRÚC VÀ TÍNH NĂNG CỦA ARDUINO

2.1.GIỚI THIỆU CHUNG VỀ ARDUINO.

Arduino thực sự đã gây sóng gió trên thị trường người dùng DIY trên toàn thế giới trong vài năm gần đây, gần giống với những gì Apple đã làm được trên thị trường thiết bị di động. Số lượng người dùng cực lớn và đa dạng với trình độ trải rộng từ bậc phổ thông lên đến đại học đã làm cho ngay cả những người tạo ra chúng (hình 2.1) phải ngạc nhiên về mức độ phổ biến.



Hình 2.1: Những thành viên khởi xướng Arduino.

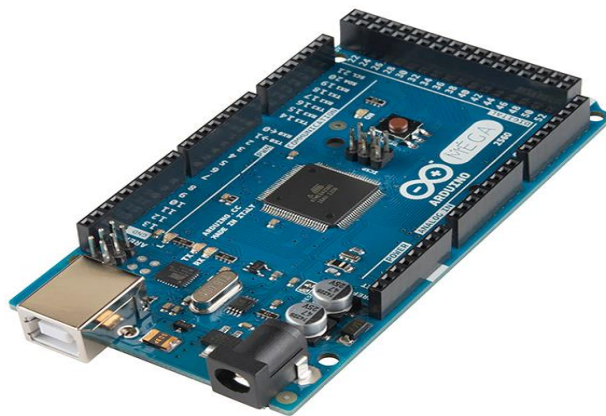
Arduino là gì mà có thể khiến ngay cả những sinh viên và nhà nghiên cứu tại các trường đại học danh tiếng như MIT, Stanford, Carnegie Mellon phải sử dụng; hoặc ngay cả Google cũng muốn hỗ trợ khi cho ra đời bộ kit Arduino Mega ADK dùng để phát triển các ứng dụng Android tương tác với cảm biến và các thiết bị khác.

Arduino thật ra là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả

với người ít am hiểu về điện tử và lập trình. Và điều làm nên hiện tượng Arduino chính là mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm. Chỉ với khoảng \$30, người dùng đã có thể sở hữu một board Arduino có 20 ngõ I/O có thể tương tác và điều khiển chừng ấy thiết bị.

Arduino ra đời tại thị trấn Ivrea thuộc nước Ý và được đặt theo tên một vị vua vào thế kỷ thứ 9 là King Arduino. Arduino chính thức được đưa ra giới thiệu vào năm 2005 như là một công cụ khiêm tốn dành cho các sinh viên của giáo sư Massimo Banzi, là một trong những người phát triển Arduino, tại trường Interaction Design Institute Ivrea (IDII). Mặc dù hầu như không được tiếp thị gì cả, tin tức về Arduino vẫn lan truyền với tốc độ chóng mặt nhờ những lời truyền miệng tốt đẹp của những người dùng đầu tiên. Hiện nay Arduino nổi tiếng tới nỗi có người tìm đến thị trấn Ivrea chỉ để tham quan nơi đã sản sinh ra Arduino.

2.2. BOARD ARDUINOMEGA 2560

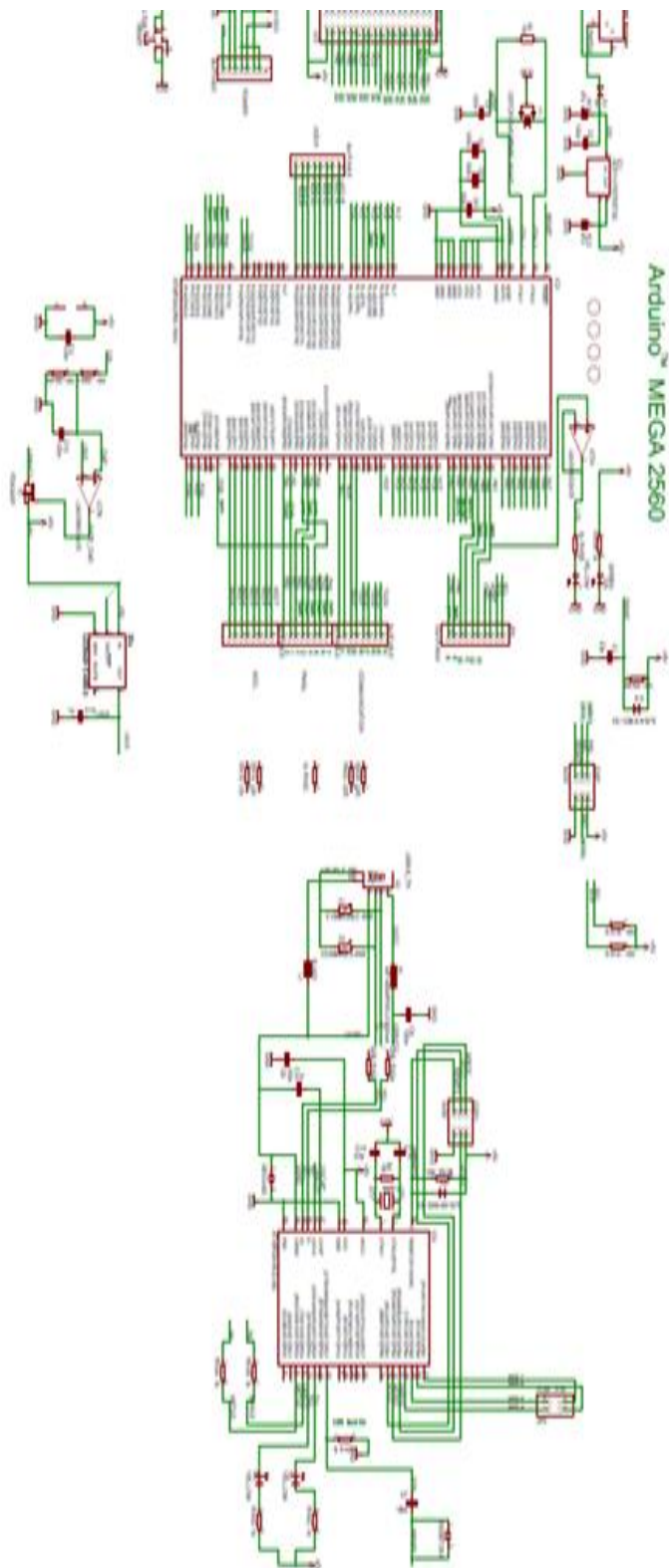


Hình 2.2: Board Arduino Mega.

Ở hình 2.2 Arduino Mega 2560 là một bo mạch chủ sử dụng vi điều khiển ATmega 2560. Nó có 54 chân kỹ thuật số đầu vào / đầu ra (trong đó 15 chân có thể xuất ra xung PWM), 16 đầu vào analog, 4 UARTS (cổng nối tiếp phần cứng), 16 MHz dao động tinh thể, kết nối USB, một jack cắm điện, một

tiêu đề ICSP, và một nút reset. Nó chứa tất cả mọi thứ cần thiết để hỗ trợ các vi điều khiển; chỉ cần kết nối nó với máy tính bằng cáp USB hoặc đầu nó với một bộ chuyển đổi AC-to-DC hoặc pin để bắt đầu. Mega là tương thích với hầu hết các lá chắn được thiết kế cho các Arduino Duemilanove hoặc Diecimila. Mega 2560 là một bản cập nhật cho Mega Arduino, mà nó thay thế. Các Mega2560 khác với tất cả các bảng trước ở chỗ nó không sử dụng các FTDI chip điều khiển USB-to-serial. Thay vào đó, nó có tính năng ATmega16U2

Sơ đồ nguyên lý như theo hình 2.3:



Hình 2.3: Sơ đồ nguyên lý của boar Arduino mega 2650.

Bảng 2.1:Chi tiết thông số

Vi điều khiển	ATmega2560
Điện áp hoạt động	5V
Điện áp đầu vào (được đề nghị)	7-12V
Điện áp đầu vào (giới hạn)	6-20V
Digital I / O Pins	54 (trong đó 15 người cung cấp đầu ra PWM)
Analog Input Pins	16
DC hiện tại mỗi I / O Pin	40 mA
DC hiện tại cho 3.3V Pin	50 mA
Bộ nhớ flash	256 KB trong đó 8 KB sử dụng bởi bộ nạp khởi động
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Cấu trúc của Arduino Mega

Arduino Mega có thể được cấp nguồn thông qua kết nối USB hoặc với một nguồn cung cấp điện bên ngoài. Các nguồn năng lượng được chọn tự động.

Bên ngoài (không-USB) điện có thể đến hoặc từ một bộ chuyển đổi AC-to-DC hoặc pin. Các bộ chuyển đổi có thể được kết nối bằng cách cắm một plug-2.1mm trung tâm tích cực vào jack cắm điện của bo mạch. Dẫn từ một pin có thể được chèn vào trong Gnd và Vin pin tiêu đề của kết nối POWER.

Bo mạch có thể hoạt động trên một nguồn cung cấp bên ngoài của 6-20 volt. Nếu cung cấp ít hơn 7V, tuy nhiên, pin 5V có thể cung cấp ít hơn năm

volt và bo mạch có thể không ổn định. Nếu sử dụng nhiều hơn 12V, bộ điều chỉnh điện áp có thể bị quá nóng và làm hỏng các bảng mạch. Phạm vi đề nghị là 7 đến 12 V.

- **Các chân điện như sau:**

- **VIN.** Các điện áp đầu vào cho các board Arduino khi nó được sử dụng một nguồn điện bên ngoài (như trái ngược với 5 volts từ các kết nối USB hoặc nguồn điện quy định khác). Bạn có thể cung cấp điện áp thông qua pin này, hoặc, nếu cung cấp điện áp thông qua jack cắm điện, truy cập thông qua pin này.

- **5V.** pin này xuất ra một 5V quy định từ điều trên diễn đàn. Từ nguồn có thể được cung cấp nguồn điện hoặc từ các jack cắm điện DC (7 - 12V), kết nối USB (5V), hoặc pin VIN của hội đồng quản trị (7-12V). Cung cấp điện áp qua các 5V hoặc 3.3V chân đi qua bộ điều chỉnh để có điện áp phù hợp vs một số linh kiện cần áp như vậy. Có các thông số kí hiệu:

- **3v3.** Một nguồn cung cấp 3,3 volt được tạo ra bởi những điều trên tàu. Về hiện hành tối đa là 50 mA.

- **GND.** trệt chân.

- **IOREF.** pin này trên bảng Arduino cung cấp các tài liệu tham khảo điện áp mà các vi điều khiển hoạt động. Một lá chắn cấu hình đúng cách có thể đọc các pin điện áp IOREF và chọn nguồn năng lượng thích hợp hoặc cho phép dịch điện áp trên các kết quả đầu ra để làm việc với các 5V hoặc 3.3V.

- **Bộ nhớ**

Các ATmega2560 có 256 KB bộ nhớ flash để lưu trữ mã (trong đó có 8 KB được sử dụng cho các bộ nạp khởi động), 8 KB của SRAM và 4 KB của EEPROM (mà có thể được đọc và ghi với các thư viện EEPROM).

- **Đầu vào và đầu ra**

Mỗi phòng trong số 54 chân kỹ thuật số trên Mega có thể được sử dụng như một đầu vào hoặc đầu ra, sử dụng pinMode (), digitalWrite

(), digitalRead () chức năng. Chúng hoạt động tại 5 volts. Mỗi pin có thể cung cấp hoặc nhận được tối đa 40 mA và có một điện trở kéo lên bên trong (ngắt kết nối theo mặc định) 20-50 kOhms. Ngoài ra, một số chân có chức năng đặc biệt:

- **Serial: 0 (RX) và 1 (TX); Nối tiếp 1: 19 (RX) và 18 (TX); Nối tiếp 2: 17 (RX) và 16 (TX); Nối tiếp 3: 15 (RX) và 14 (TX).** Được sử dụng để nhận (RX) và truyền (TX) TTL dữ liệu nối tiếp. Pins 0 và 1 cũng được kết nối với các chân tương ứng của ATmega16U2 USB-to-TTL nối tiếp chip.

- **Ngắt ngoài: 2 (gián đoạn 0), 3 (gián đoạn 1), 18 (gián đoạn 5), 19 (gián đoạn 4), 20 (gián đoạn 3) và 21 (gián đoạn 2).** Các chân này có thể được cấu hình để kích hoạt một ngắt trên một giá trị thấp, một góc lên và xuống, hoặc một sự thay đổi trong giá trị. Xem các attachInterrupt () chức năng để biết chi tiết.

- **PWM: 2-13 và 44-46.** Cung cấp 8-bit PWM với analogWrite () chức năng.

- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS)** Các chân này hỗ trợ SPI giao tiếp bằng cách sử dụng thư viện SPI . Các chân SPI cũng được chia ra trên tiêu đề ICSP, đó là chất tương thích với Uno, Duemilanove và Diecimila.

- **LED: 13** Có một built-in LED kết nối với pin số 13. Khi pin là giá trị cao, đèn LED được bật, khi pin là LOW, nó ra.

- **TWI: 20 (SDA) và 21 (SCL).** Hỗ trợ TWI giao tiếp sử dụng các thư viện Wire. Lưu ý rằng các chân không ở cùng một vị trí như các chân TWI trên Duemilanove hoặc Diecimila.

Các Mega2560 có 16 đầu vào analog, mỗi trong số đó cung cấp 10 bit độ phân giải (tức là 1024 giá trị khác nhau). Theo mặc định họ đo từ mặt đất đến 5 volts, mặc dù là nó có thể thay đổi vào cuối trên của phạm vi của họ bằng cách sử dụng pin Aref và analogReference chức năng.

- **Một số chân khác:**

Aref. Điện áp tham chiếu cho các đầu vào analog. Được sử dụng với analogReference.

Thiết lập lại. Mang dòng LOW này để thiết lập lại các vi điều khiển. Thường được sử dụng để thêm một nút reset để lá chắn ngăn chặn sự một trên bảng.

- **Truyền thông kết nối với máy tính**

Arduino Mega2560 có một số phương tiện truyền thông với một máy tính, một Arduino, hoặc vi điều khiển khác. Các ATmega2560 cung cấp bốn phần cứng UARTS cho TTL (5V) giao tiếp nối tiếp.

Một ATmega16U2 (ATmega 8U2 về sửa đổi 1 và phiên bản 2 bảng) trên kênh board một trong những trên USB và cung cấp một cổng com ảo với phần mềm trên máy tính (máy tính Windows sẽ cần một file .inf, nhưng OSX và Linux máy sẽ công nhận hội đồng quản trị như một cổng COM tự động. Các phần mềm Arduino bao gồm một màn hình nối tiếp cho phép dữ liệu văn bản đơn giản được gửi đến và đi từ hội đồng quản trị. Các RX và TX đèn LED trên bảng sẽ nhấp nháy khi dữ liệu đang được truyền đi thông qua ATmega8U2 ATmega16U2 chip và USB kết nối với máy tính (nhưng không cho giao tiếp nối tiếp trên các chân 0 và 1).

Một thư viện SoftwareSerial cho phép giao tiếp nối tiếp trên bất kỳ của các chân kỹ thuật số của Mega2560.

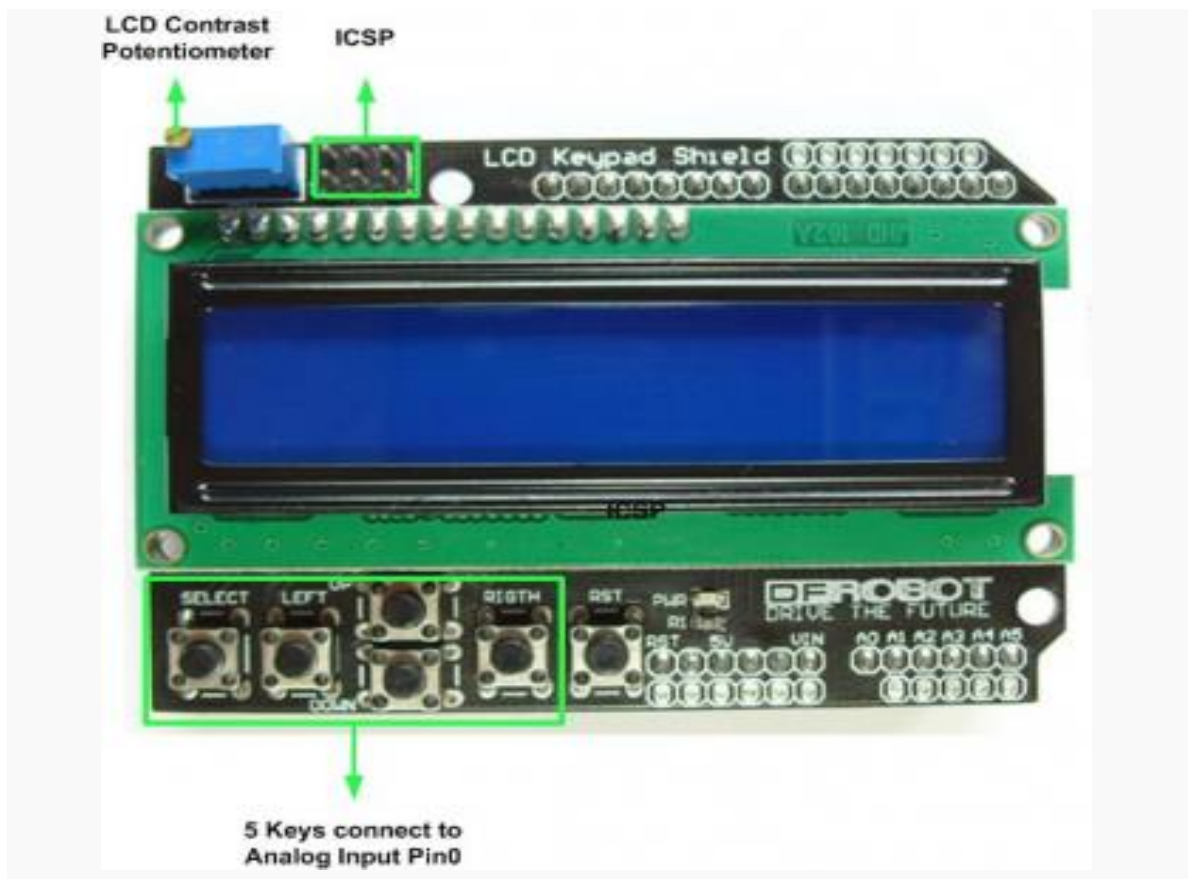
Các ATmega2560 cũng hỗ trợ TWI và SPI truyền thông. Các phần mềm Arduino bao gồm một thư viện Wire để đơn giản hóa việc sử dụng các bus TWI; Đối với SPI giao tiếp, sử dụng các thư viện SPI.

- **Lập trình**

Arduino Mega có thể được lập trình với các phần mềm Arduino. Các ATmega2560 trên Mega Arduino đi kèm preburned với một bộ nạp khởi động cho nó mà không sử dụng cho phép bạn tải lên mã mới dùng một lập trình viên phần cứng bên ngoài. Nó giao tiếp bằng cách sử dụng cổng STK500 giao thức (tài liệu tham khảo , các tập tin tiêu đề C).

2.3.ARDUINO LCD KEYPAD SHIELD.

Các tấm LCD Bàn phím như hình 2.4 được phát triển cho tương thích với các board Arduino, để cung cấp một giao diện người dùng thân thiện cho phép người dùng kiểm tra thông số, thực hiện lựa chọn vv... Nó bao gồm một 1602 ký tự trắng đèn nền màu xanh LCD. Bàn phím bao gồm 5 phím -, xuống chọn, lên, phải và bỏ đi. Để lưu các chân IO kỹ thuật số, giao diện bàn phím chỉ sử dụng một kênh ADC. Giá trị quan trọng được đọc thông qua một điện áp chia 5 giai đoạn.



Hình 2.4:LCD KeyPad Shield.

Bảng 2.2: Sơ đồ kết nối chân của Arduino LCD KeyPad Shield

Ghim	Hàm số
Analog 0	Button (chọn, lên, phải, xuống và trái)
Kỹ thuật số 4	DB4
Kỹ thuật số 5	DB5
Kỹ thuật số 6	DB6
Kỹ thuật số 7	DB7
Kỹ thuật số 8	RS (Data hoặc tín hiệu Display Selection)
Kỹ thuật số 9	Cho phép
Kỹ thuật số 10	Backlit kiểm soát

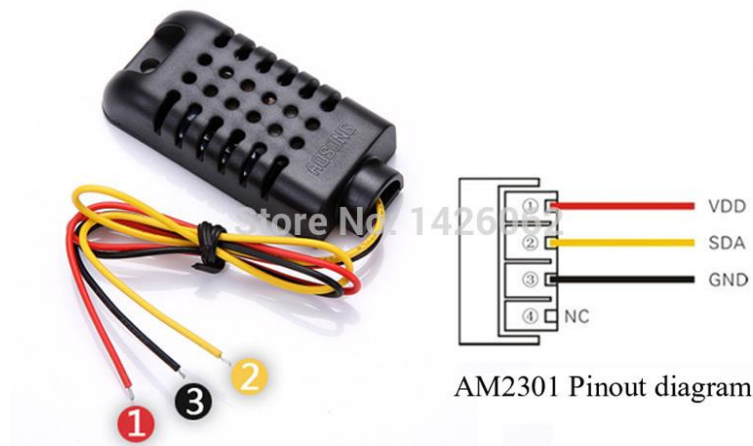
2.4. CẢM BIẾN NHIỆT ĐỘ, ĐỘ ẨM DHT 21.

Cảm biến DHT21/AM2301 như hình 2.5 là module tích hợp cảm biến độ ẩm điện dung và cảm biến nhiệt độ có độ chính xác cao, đầu ra tín hiệu số có thể kết nối với một vi điều khiển 8 bit. Chất lượng cảm biến cao, đáp ứng nhanh, có khả năng chống nhiễu mạnh, giao tiếp duy nhất một dây. Tiêu thụ điện năng thấp, khoảng cách truyền dẫn tín hiệu khoảng 20m hiệu chuẩn hoàn toàn tự động.

Cảm biến DHT21 có 3 dây với 3 màu đen đỏ vàng có chức năng :

+ Dây màu đỏ: VCC 3.3 - 5V.

- + Dây màu vàng: dây truyền dữ liệu (DATA).
- + Dây màu đen: GND.



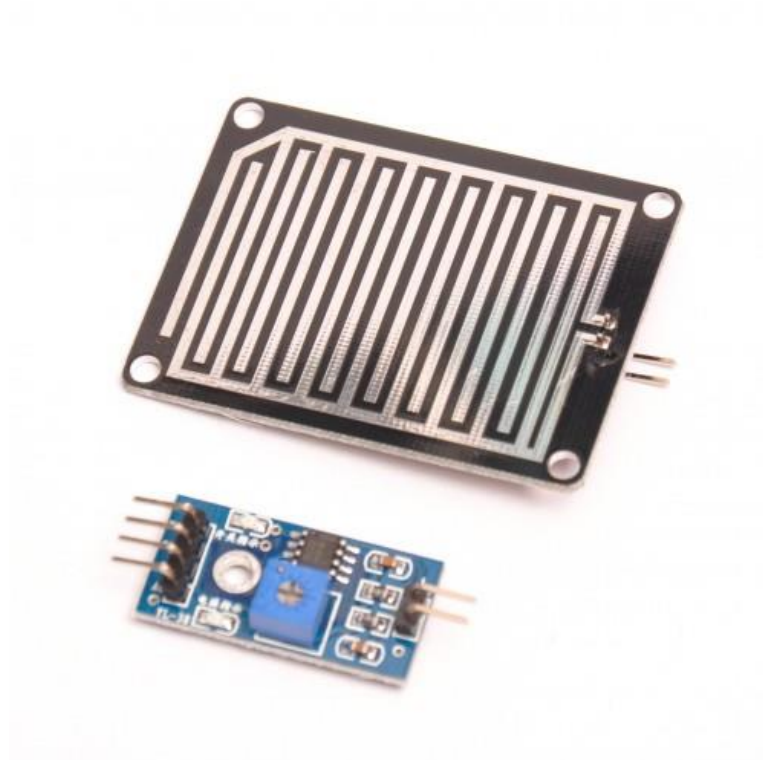
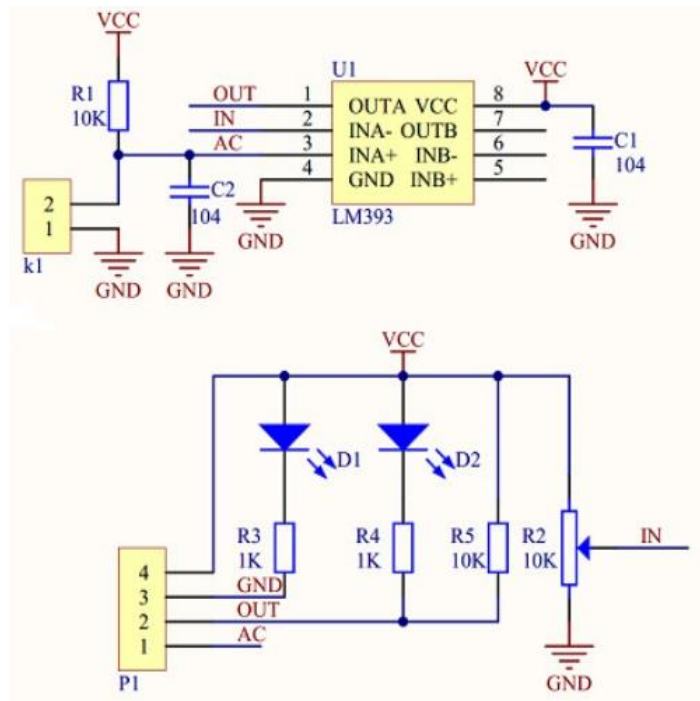
Hình 2.5: Cảm biến DHT21.

Thông số kỹ thuật :

- Điện áp hoạt động: 3.3 – 5 V.
- Dải độ ẩm: 0 – 99.9%.
- Dải đo nhiệt độ: -40 – 80 °C.
- Sai số độ ẩm: $\pm 3\%$.
- Sai số nhiệt độ: ± 0.3 °C.
- Chuẩn giao tiếp: 1 dây (1 wire).

2.5.MODULE CẢM BIẾN MƯA (SENSRAIN).

Chúng ta có thể dễ dàng nhận biết mưa quá mắt hoặc cảm nhận da. Hệ thốn điện tử cũng có thể làm tương tự như con người vậy. Nhưng để tiết kiệm và dễ dàng người ta sử dụng cảm biến cũng tương tự như da con người. Cảm biến mưa là board có các đường kim loại xen kẽ tạo khoảng cách điện để khi nước rơi vào tấm sẽ tạo ra môi trường dẫn điện. Trên hình 2.6 hình ảnh sơ đồ nguyên lý và bo mạch cảm biến mưa.



Hình 2.6: Module cảm biến mưa (SENSRAIN).

Thông số kỹ thuật:

- Điện áp: 5V.
- Led báo nguồn: PWR LED.
- Led báo mưa: D0 LED.

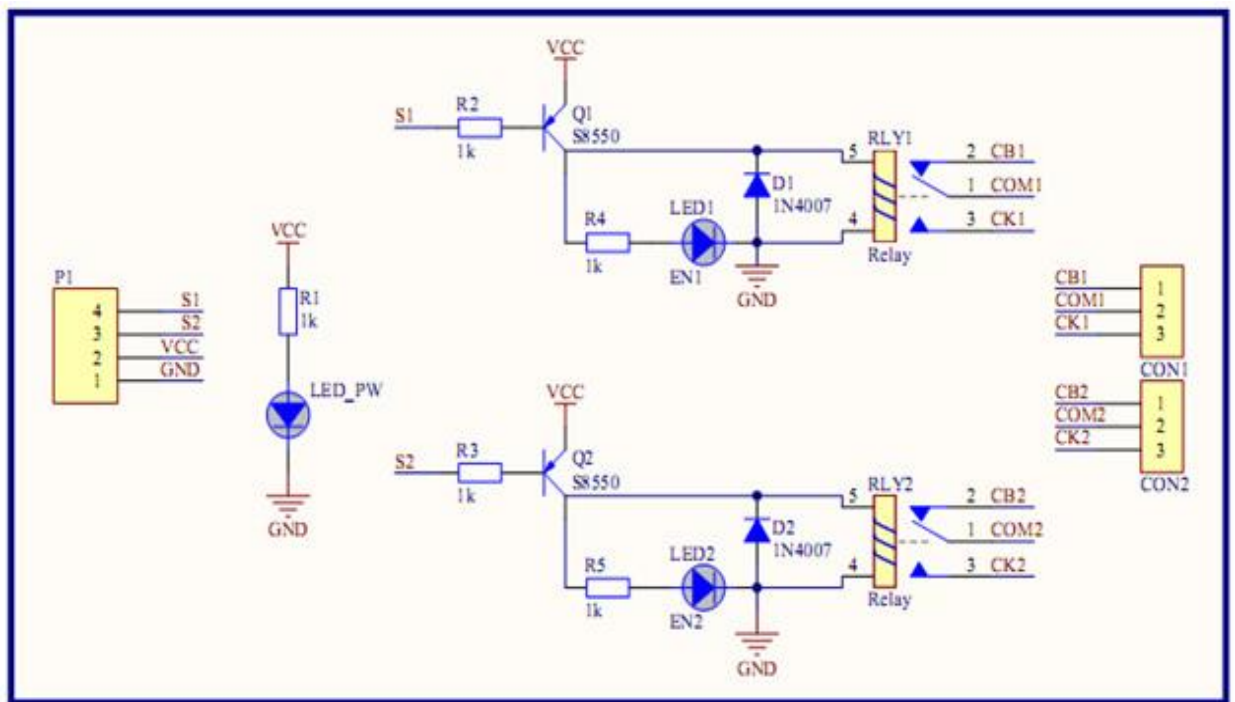
Có 2 dạng tín hiệu: Analog(A0) và Digital (D0).

- Dạng tín hiệu: TTL, đầu ra 100mA (Có thể sử dụng trực tiếp relay).
- Điều chỉnh độ nhạy bằng biến trở.
- Sử dụng LM393 để chuyển A0 sang D0.
- D0: Đầu ra ở mức cao (1), khi có nước đèn LEDD0 báo sáng, đồng thời đầu ra về mức thấp (0). Có thể dùng đầu nối qua relay hoặc đưa vào chân I/O của VĐK.
- A0: Dùng để xác định độ lớn của giọt nước, bằng cách đưa vào ADC của VĐK.

2.6.GIỚI THIỆU VỀ MODULE RELAY 12VDC.

Relay hình 2.7 là thiết bị đóng cắt cơ bản, nó được sử dụng rất nhiều trong cuộc sống và trong các thiết bị điều khiển điện tử.

- Cấu tạo relay gồm 2 phần:
 - + Cuộn hút: tạo ra năng lượng từ trường để hút các tiếp điểm về mình.
 - + Cặp tiếp điểm: khi không cấp điện tiếp điểm 1 được tiếp xúc tiếp điểm 2 (tiếp điểm thường đóng). Khi cấp điện tiếp điểm 1 bị hút chuyển sang tiếp điểm 3 (tiếp điểm thường mở).



Hình2.7: Hình ảnh và sơ đồ của module relay 2 kênh 5VDC.

- Điều khiển Relay DC:

+ Trong mạch trên là sơ đồ của 2 cặp relay.

+ Hoạt động của relay 1 (sử dụng transistor PNP).

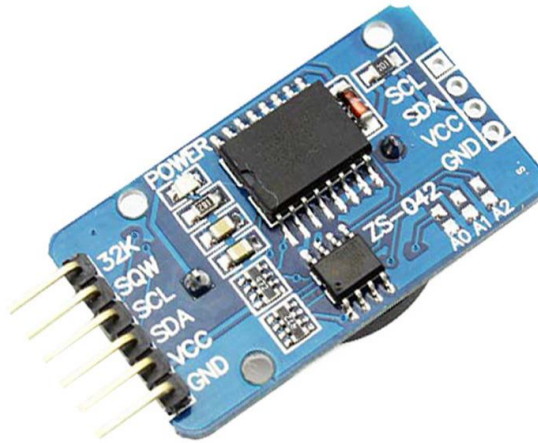
- Khi $S1 = 1$: Q1 khóa lại, không có dòng chạy qua cuộn hút relay1. Đèn LED1 tắt. Tiếp điểm 1 và 2 nối nhau.
- Khi $S1 = 0$: Q1 mở, có dòng từ VCC qua khóa Q1 cấp điện cho cuộn hút. Lúc này có chuyển mạch của cặp tiếp điểm. Tiếp điểm 1 nối với 3. Đèn LED Q1 sáng.
- Trong mạch có sử dụng Diode D1 mắc vào 2 đầu cuộn dây của relay. D1 có tác dụng xả dòng cho cuộn hút khi nó không hoạt động.
- Hoạt động relay 2 tương tự.

Thông số kỹ thuật:

- VCC,GND: Nguồn nuôi chung của thiết bị điều khiển (5V).
- VSS+,VSS-: Nguồn của relay.
- Các cấp điện áp tiếp điểm cho phép: 250VAC~125VAC -10A.
30VDC~28VDC -10A.
- NO,NC: Các tiếp điểm thường mở, thường đóng.

2.7.MODULE THỜI GIAN THỰC.

Module thời gian thực hình 2.8 là IC thời gian thực giá rẻ, rất chính xác với thạch anh tích hợp có sẵn. IC có đầu vào cho pin riêng, tách biệt khỏi nguồn chính đảm bảo cho việc giữ thời gian chính xác.



Hình 2.8 : Hình ảnh module thời gian thực (DS3231).

Thời gian IC được giữ ở trạng thái giờ, phút, giây, ngày, thứ, tháng, năm. Tất cả được điều chỉnh cho phù hợp với hiện tại, có các chế độ 12h AmPm hoặc 24h. Trong chip có mạch điện áp chuẩn dùng để theo dõi trạng thái nguồn VCC, phát hiện lỗi nguồn, tự động chuyển nguồn khi có vấn đề.

Thông số kỹ thuật :

- Nguồn VCC : 3,5 – 5 V.
- Clock : chip DS3231 (nâng cấp DS1307).
- Thông tin thời gian : thời gian đến 2100.
- I2C bus có tốc độ tối đa 400Khz.
- Có pin sạc khi mất điện.
- Memory IC AT24C32 (32k bit).

2.8.ĐỘNG CƠ MỘT CHIỀU 12VDC.

Động cơ được sử dụng là động cơ nam châm vĩnh cửu hình 2.9 ở đĩa máy bơm áp lực cao, kích thước nhỏ, trọng lượng nhẹ, mô men xoắn lớn, áp suất cao, hoạt động ổn định, bơm tự môi được xây dựng với công tắc áp suất tự động. Chức năng bảo vệ quá nhiệt, tiếng ồn thấp, chống axit kiềm, chống ăn mòn.



Hình 2.9: Hình ảnh động cơ 12VDC - 6A.

Máy bơm có một công tắc áp lực, áp lực làm việc cao nó sẽ tự động ngừng hoạt động. Có thể chạy khô mà không có thiệt hại cho động cơ.

Thông số kỹ thuật:

- Điện áp định mức 12VDC.
- Lưu lượng bơm 6L/p.
- Áp lực nước: 130PSI.
- Công suất: 70W.

2.9.NGUỒN TỔ ONG 12VDC.

Nguồn tổ ong 12V – 10A (hình 2.10) hay còn gọi là bộ nguồn một chiều 12V được thiết kế để chuyển đổi điện áp từ nguồn xoay chiều 110/220VAC thành nguồn một chiều 12VDC để cung cấp cho các thiết bị hoạt động.

Nguồn tổ ong 12V-10A được sử dụng rộng rãi trong các thiết bị công nghiệp và dân dụng. Trong công nghiệp chúng thường được sử dụng để cấp nguồn cho một số thiết bị của tủ điện.



Hình 2.10 : Hình ảnh nguồn tổ ong 12V-10A.

Thông số kĩ thuật :

- Điện áp ngõ vào : 110/220 VAC (Chân L và N).
- Điện áp ngõ ra 12 VDC (Chân V+, GND, V-).
- Sai số điện áp đầu ra 1-3%.
- Công suất : 120 W.
- Nhiệt độ làm việc : 0-70 °C.

2.10. PHẦN MỀM MÔ PHỎNG PROTEUS.

Phần mềm Proteus hình 2.11 là phần mềm cho phép mô phỏng hoạt động của mạch điện tử bao gồm phần thiết kế mạch và viết chương trình điều khiển cho các họ vi điều khiển như MCS-51, PIC, AVR, ... Proteus là phần mềm mô phỏng mạch điện tử của Lancenter Electronics, mô phỏng cho hầu hết các linh kiện điện tử thông dụng, đặc biệt hỗ trợ cho các MCU như PIC, 8051, AVR, Motorola.

Phần mềm bao gồm 2 chương trình: **ISIS** cho phép mô phỏng mạch và **ARES** dùng để vẽ mạch in. Proteus là công cụ mô phỏng cho các loại vi điều khiển khá tốt, nó hỗ trợ các dòng vi điều khiển PIC, 8051, PIC, dsPIC, AVR,

HC11,... Các giao tiếp I2C, SPI,CAN, USB, Ethenet... Ngoài ra còn mô phỏng các mạch số, mạch tương tự một cách hiệu quả.



Hình 2.11: Giao diện khởi động phần mềm Proteus.

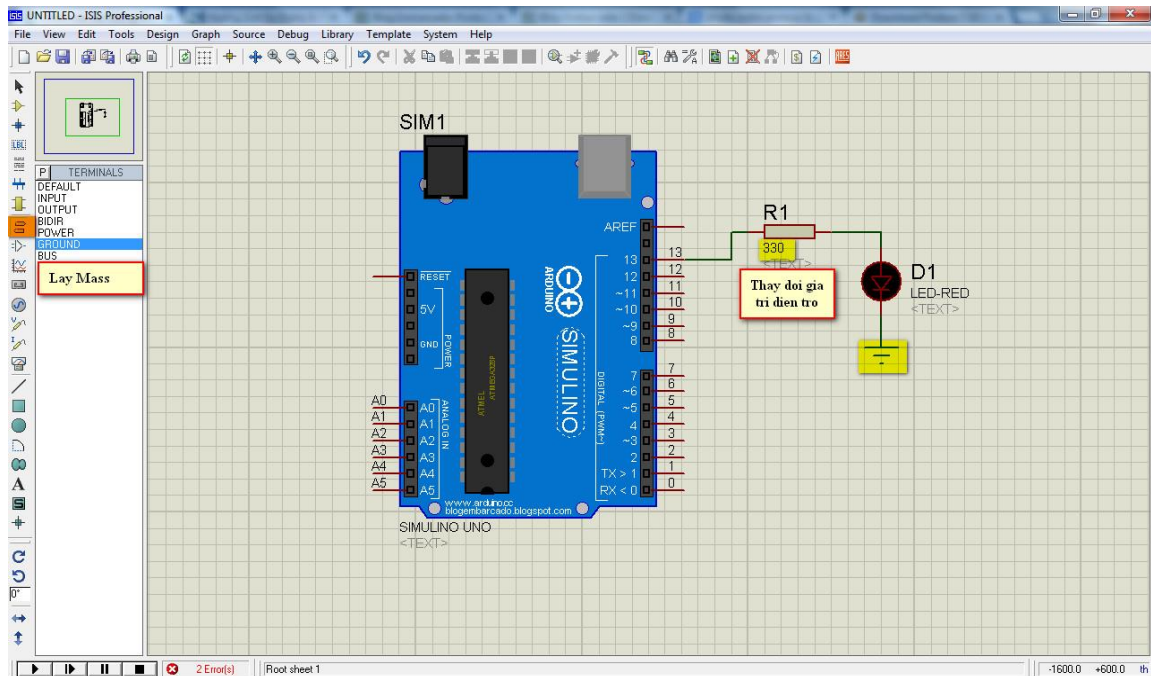
2.11.THƯ VIỆN ARDUINO TRONG PROTEUS.

Thư viện Arduino là một bổ sung rất hay cho phần mềm mô phỏng Proteus nó giúp cho việc mô phỏng Arduino được thuận tiện và dễ dàng hơn thay vì chỉ mô phỏng được chip ATmega328(nhân của Arduino), thư viện này được phát triển bởi các kĩ sư Cesar, Osaka, Daniel Cezar, Roberto Bauer và được đăng tải trên blog tiếng Bồ Đào Nha:<http://blogembarcado.blogspot.de/>

Thư viện bao gồm các linh kiện sau:

- Arduino Uno (Phiên bản chip ATmega328 chân DIP).
- Arduino Uno (Phiên bản chip ATmega328 chân SMD).
- Arduino Mega.
- Arduino Lilypad.
- Arduino Nano.
- Cảm biến siêu âm Ultrasonic V2.

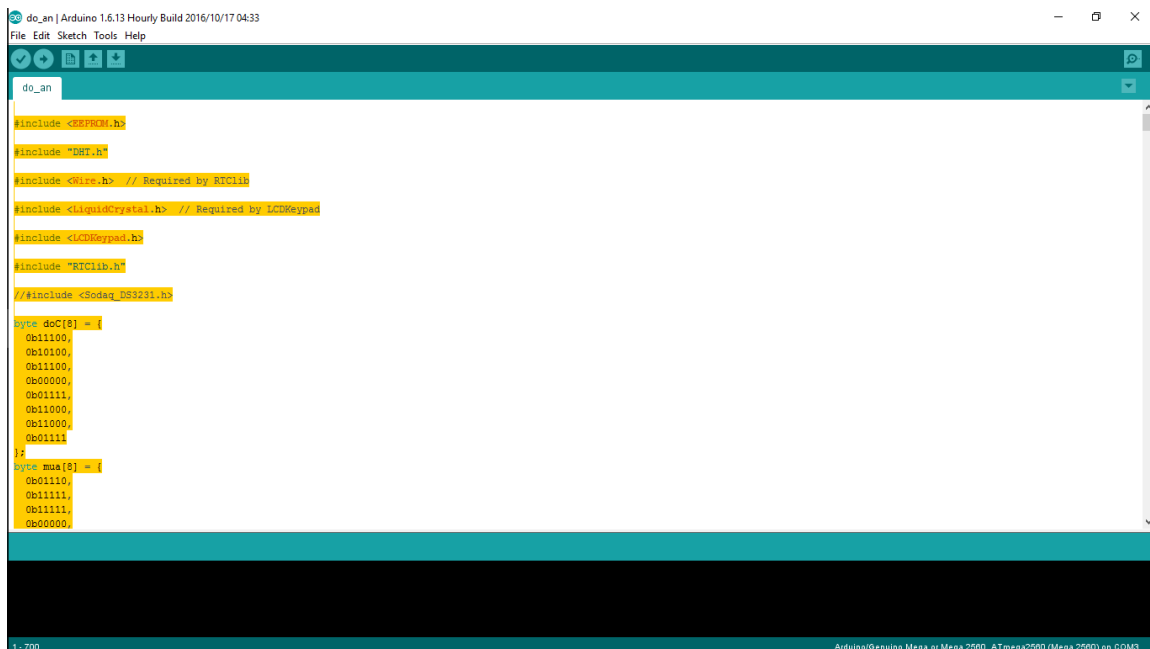
Hình 2.12 lấy minh họa cho mô phỏng là arduino Uno:



Hình 2.12 Các linh kiện trong thư viện Arduino cho Proteus.

2.12. ARDUINO IDE VÀ LẬP TRÌNH CHO ARDUINO.

Thiết kế bo mạch nhỏ gọn, trang bị nhiều tính năng thông dụng mang lại nhiều lợi thế cho Arduino, tuy nhiên sức mạnh thực sự của Arduino nằm ở phần mềm. Môi trường lập trình đơn giản dễ sử dụng, ngôn ngữ lập trình Wiring dễ hiểu và dựa trên nền tảng C/C++ rất quen thuộc với người làm kỹ thuật. Và quan trọng hơn là số lượng thư viện code được viết sẵn và chia sẻ bởi cộng đồng nguồn mở là cực kỳ lớn. Hình ảnh 2.13 khi mở giao diện Arduino IDE lên sử dụng.



Hình 2.13: Giao diện phần mềm Arduino IDE.

Arduino là phần mềm dùng để lập trình cho arduino. Môi trường lập trình cho arduino là IDE có thể chạy trên ba nền tảng phổ biến nhất hiện nay là Windows, Macintosh OSX và Linux. Do có tính chất nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng hơn bởi người dùng có kinh nghiệm.

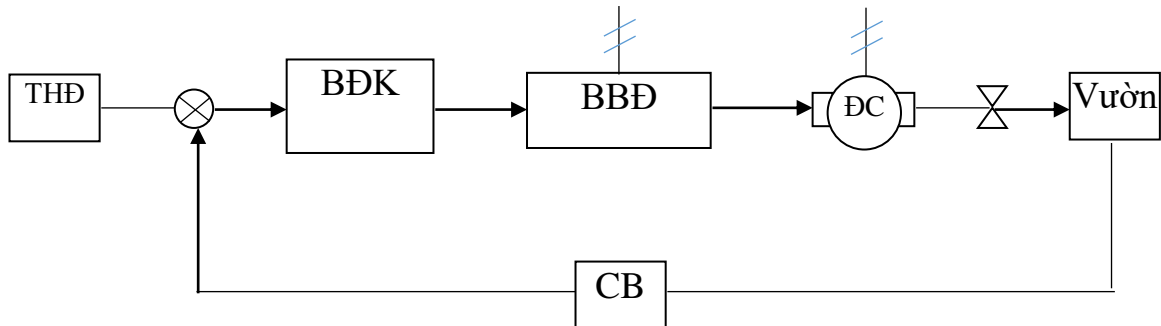
Ngôn ngữ lập trình có thể được mở rộng thông qua các thư viện C++. Và ngôn ngữ lập trình này dựa trên nền tảng của ngôn ngữ C của AVR nên người dùng hoàn toàn có thể nhúng thêm code viết bằng AVR vào chương trình nếu muốn. Hiện tại, Arduino IDE có thể download từ trang chủ <http://arduino.cc/>

CHƯƠNG 3.

THIẾT KẾ, LẬP TRÌNH, LẮP ĐẶT MẠCH THỰC TẾ

3.1. GIỚI THIỆU CHUNG CỦA HỆ THỐNG.

Hệ thống tưới tự động cho máy bơm nước tự động được biểu diễn trên hình 3.1 như sau:



Hình 3.1: Hệ thống tưới tự động.

-THĐ: Tín hiệu đặt.

-BĐK: Bộ điều khiển.

-ĐC: Động cơ.

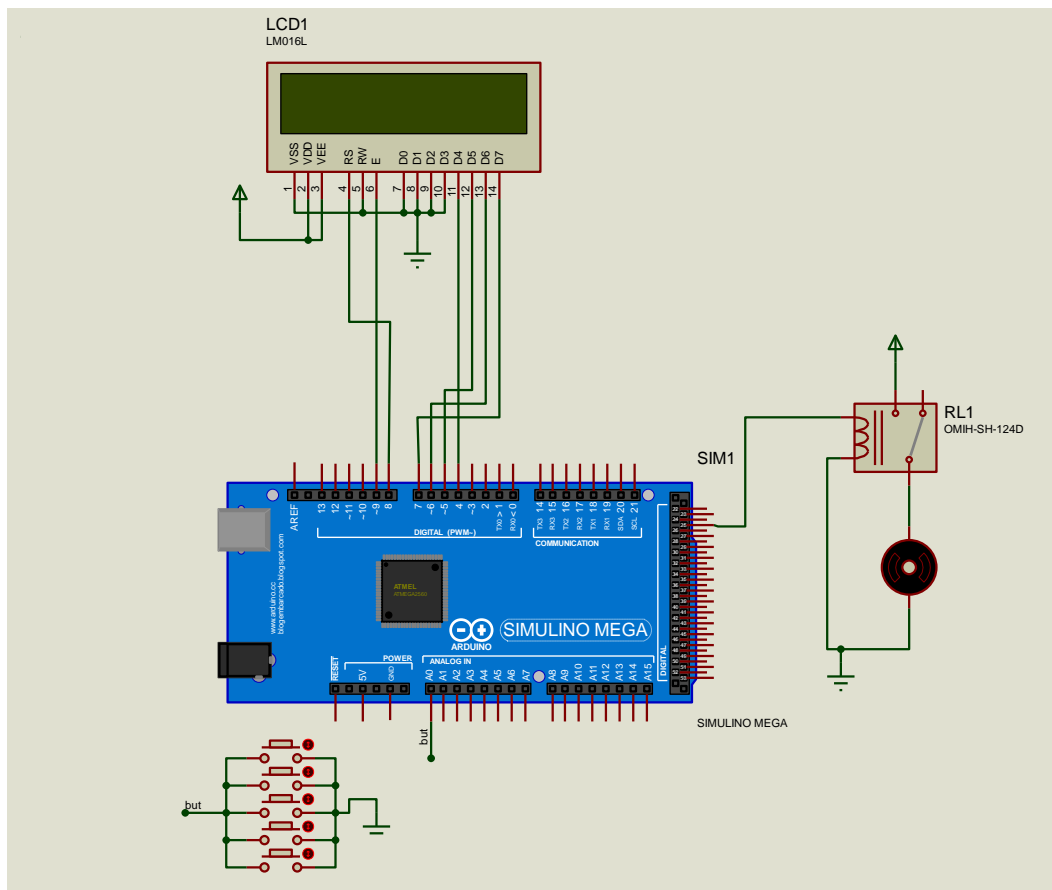
-CB: Các cảm biến.

Nguyên lý hoạt động:

Tín hiệu đặt là việc cài đặt do người lập trình đặt ra các điều kiện điều khiển cho hệ thống. Các giá trị thay đổi về thời tiết, nhiệt độ, độ ẩm được nhận biết qua các cảm biến. Các giá trị của cảm biến đó sẽ đưa về bộ điều khiển so sánh với các giá trị cài đặt qua bộ điều khiển. Bộ điều khiển xử lý đưa ra các tín hiệu điều khiển sang bộ biến đổi để thực hiện việc điều khiển động cơ hoạt động hợp lý. Động cơ hoạt động sẽ đưa nước tới các téc phun tưới cây đảm bảo việc chăm sóc các cây trồng phát triển tốt. Hệ thống làm việc liên tục khoa học tránh các thao tác thừa khi điều khiển động cơ.

3.2. THIẾT KẾ MẠCH TRÊN PROTEUS.

Nhóm nhiệt độ độ ẩm, mưa rồi tín hiệu từ này lại được truyền Arduino ra LCD. Về vấn đề mô phỏng vẫn phải thực hiện trên thực tế vì không có thư viện mô phỏng trên Proteus nhưng có thể thay thế bằng cảm biến nhiệt LM35 trên proteus để mô phỏng. Tiếp đến là phần kết nối với IC L293D để điều khiển 2 động cơ DC một chiều 12V. Động cơ được điều khiển qua các mức tín hiệu thông qua relay điều khiển qua các tiếp điểm. Việc cài đặt thông qua các button được so sánh qua điện áp trong vi điều khiển thông qua lập trình. Mô phỏng sơ bộ trên hình 3.2:



Hình 3.2: Sơ đồ mạch sơ bộ.

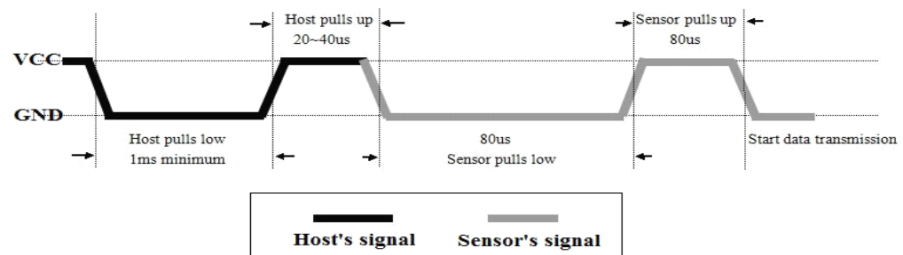
3.3.LẬP TRÌNH HIỂN THỊ LCD.

3.3.1. Lập trình đo nhiệt độ, độ ẩm.

Như chương trước đã giới thiệu, cảm biến DHT21 đo nhiệt độ, độ ẩm đó dưới dạng điện áp. Để có thể giao tiếp với DHT22 theo chuẩn 1 chân vì xử lý thực hiện theo 2 bước.

- Gửi tín hiệu muốn đo (Start) tới DHT22, sau đó xác nhận lại.
- Khi đã giao tiếp được với DHT22, cảm biến sẽ gửi lại 5 byte dữ liệu và nhiệt độ đo được.

-Bước 1 : Gửi tín hiệu start



Hình 3.3: Các mức tín hiệu.

+ MCU thiết lập chân DATA là output kéo chân DATA xuống 0 trong khoảng thời gian ≥ 1 ms. Khi đó DHT22 sẽ hiểu là MCU muốn đo nhiệt độ độ ẩm.

- MCU đưa chân DATA lên 1 sau đó thiết lập lại là chân đầu vào.
- Sau khoảng 20-40 us DHT sẽ kéo chân DATA xuống thấp. Nếu $>40\mu s$ mà chân DATA chưa được kéo xuống thấp nghĩa là chưa giao tiếp được với DHT22.
- Chân DATA sẽ ở mức thấp 80 us sau đó được DHT22 kéo lên mức cao trong 80 us. Bằng việc giám sát chân DATA, MCU có thể biết được có giao tiếp được với DHT22 hay không. Nếu tín hiệu đo được lên cao khi đó hoàn thiện quá trình giao tiếp của MCU với DHT22.

-Bước 2: Đọc giá trị trên DHT22.

DHT22 sẽ trả giá trị nhiệt độ và độ ẩm về dưới dạng 5 byte trong đó:

- + Byte 1 giá trị phần nguyên của độ ẩm.
- + Byte 2 giá trị phần thập phân của độ ẩm.
- + Byte 3 giá trị phần nguyên của nhiệt độ.
- + Byte 4 giá trị phần thập phân của nhiệt độ.
- + Byte 5 kiểm tra tổng.

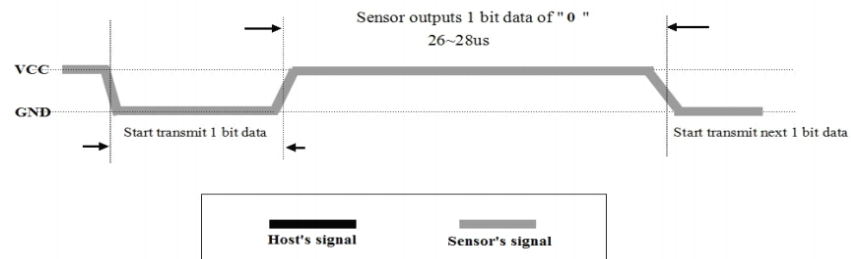
Nếu $\text{Byte 5} = \text{Byte 1} + \text{Byte 2} + \text{Byte 3} + \text{Byte 4}$ thì giá trị nhiệt độ và độ ẩm là chính xác còn nếu không thì kết quả đo bị sai.

Cách tính nhiệt độ và độ ẩm:

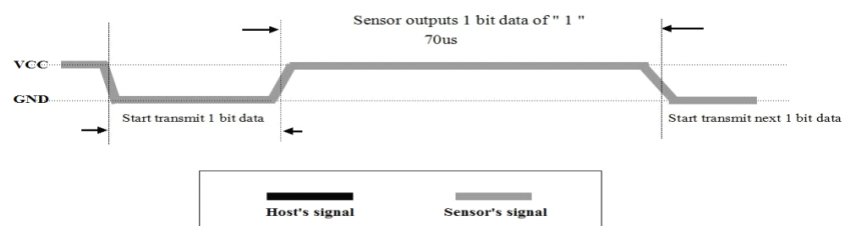
$(\text{Bytecao} * 256 + \text{Bytethấp}) / 10$.

Đọc dữ liệu : sau khi giao tiếp được với DHT22 , DHT22 sẽ gửi liên tiếp 40bit 0 hoặc 1 về MCU tương ứng với 5 byte giá trị nhiệt độ độ ẩm.

+ Bit 0



+ Bit 1



Sau khi tín hiệu được đưa về 0 ta đợi chân DATA của MCU được DHT22 kéo lên 1. Nếu chân 1 DATA trong khoảng 26-28 us thì là 0 còn

nếu tồn tại trong khoảng 70 us thì là 1. Do đó trong lập trình ta bắt sườn lên của DATA sau đó delay 50 us. Nếu giá trị đo được là 0 thì đọc được bit 0 nếu giá trị đo được là 1 thì đọc được bit 1 cứ thế ta đọc các bit tiếp theo.

Sau đây là đoạn code chính của phần đo nhiệt độ, độ ẩm:

```
// Đọc nhiệt độ độ ẩm mất khoảng 250ms  
// Đọc cảm biến có thể lên tới 2s (do khả năng thay đổi giá trị cũ rất chậm)  
float h = dht.readHumidity();  
float t = dht.readTemperature();  
//F. int sensorReading = analogRead(A0);  
// kiểm tra giá trị của sensor nếu sai báo failed  
if (isnan(t) || isnan(h)) {  
    lcd.setCursor(10,1);  
    lcd.println("Failed !");  
} else {  
    lcd.setCursor(10,1);           chọn vị trí hiển thị trên LCD  
    lcd.print(round(h));  
    lcd.print("%");  
    lcd.setCursor(13,1);  
    lcd.print(round(t));  
    lcd.write(4);                 // biểu thị biểu tượng độ C lên LCD
```

3.3.2. Lập trình trạng thái cảm biến mưa.

Bảng 3.1:Sơ đồ chân nối

Cảm biến mưa	Arduino
GND	GND
VCC	5V
D0	Digital 23

Đoạn code lập trình chính:

```
int rainSensor = 23; // Chân tín hiệu cảm biến mưa ở chân digital 6 (arduino)
void setup() {
    pinMode(rainSensor,INPUT); // Đặt chân cảm biến mưa là INPUT, vì tín hiệu sẽ được truyền đến cho Arduino
    Serial.println("Da khoi dong xong");
}
int value = digitalRead(rainSensor); //Đọc tín hiệu cảm biến mưa
if (value == LOW) {transition(OFF); // Cảm biến đang mưa
    lcd.setCursor(15,0);
    lcd.write(5);;
} else {
    lcd.setCursor(15,0);
    lcd.write(7);
```

3.3.3. Lập trình cho module DS3231(DS1307)

Khi lập trình cho module thời gian chúng ta cần chú ý lấy đúng thời gian hiện tại. Hiện nay ứng dụng khá cáo việc lấy thời gian hiện tại qua timer ngay trên máy tính nên độ khó không phức tạp lắm. Sau đây là đoạn code lập trình lấy thời gian thực:

```

void showTime()
{
    now = RTC.now();
    const char* dayName[] = { "CN", "T2", "T3", "T4", "T5", "T6", "T7" };
    const char* monthName[] = { "1", "2", "3", "4", "5", "6", "7", "8", "9",
    "10", "11", "12" };
    lcd.clear();
    lcd.print(String(dayName[now.dayOfTheWeek()]) + " " +
        (now.day() < 10 ? "0" : "") + now.day() + "." +
        monthName[now.month()-1] + "." + now.year());
    lcd.setCursor(0,1);

    lcd.print((now.hour() < 10 ? "0" : "") + String(now.hour()) + ":" +
        (now.minute() < 10 ? "0" : "") + now.minute() + ":" +
        (now.second() < 10 ? "0" : "") + now.second() );

    lcd.setCursor(8,1);
}

```

3.3.4. Lập trình hiển thị giá trị lên LCD 16.2

Sau khi các giá trị của các cảm biến được đọc thì shieuld LCD sẽ có nhiệm vụ đọc các giá trị lên màn hình. Do màn hình LCD 16.2 chỉ hiển thị được 2 dòng nên ở đây ưu tiên hiển thị về mảng thời gian và hiển thị về nhiệt độ, độ ẩm trạng thái thời tiết.

Các chân BSE D4 D5 D6 D7 của LCD sẽ lần lượt được kết nối trực tiếp vào các chân digital sau 8 9 4 5 6 7 để giúp cho nó có thể hiển thị. Tất cả quá trình đọc và ghi dữ liệu đều được thông qua thư viện có sẵn.

Code kết nối chân và hiển thị giá trị lên LCD:

```

#include <LiquidCrystal.h> // Thư viện LCD
#include <LCDKeypad.h>    //Thư viện keyboard LCD
byte doC[8] = {           //Hiển thị đơn vị độ C
    0b11100,
    0b10100,
    0b11100,
    0b00000,
    0b01111,
    0b11000,
    0b11000,
    0b01111
};

byte mua[8] = {           //Hiển thị trạng thái mưa
    0b01110,
    0b11111,
    0b11111,
    0b00000,
    0b10101,
    0b00000,
    0b10101,
    0b00000,
};

byte nang[8] = {         //Hiển thị trời không mưa
    0b00100,
    0b10101,
    0b01110,
    0b11111,
    0b01110,

```

```

0b10101,
0b00100,
0b00000,
};

byte chuong[8] = {          //Báo đang hẹn giờ
0b01110,
0b11111,
0b11111,
0b11111,
0b11111,
0b11111,
0b00000,
0b00100,
0b00000,
};

/ Khởi tạo LCD keyboard
// Xử lý màn hình và các nút trên LCD
LCDKeypad lcd;

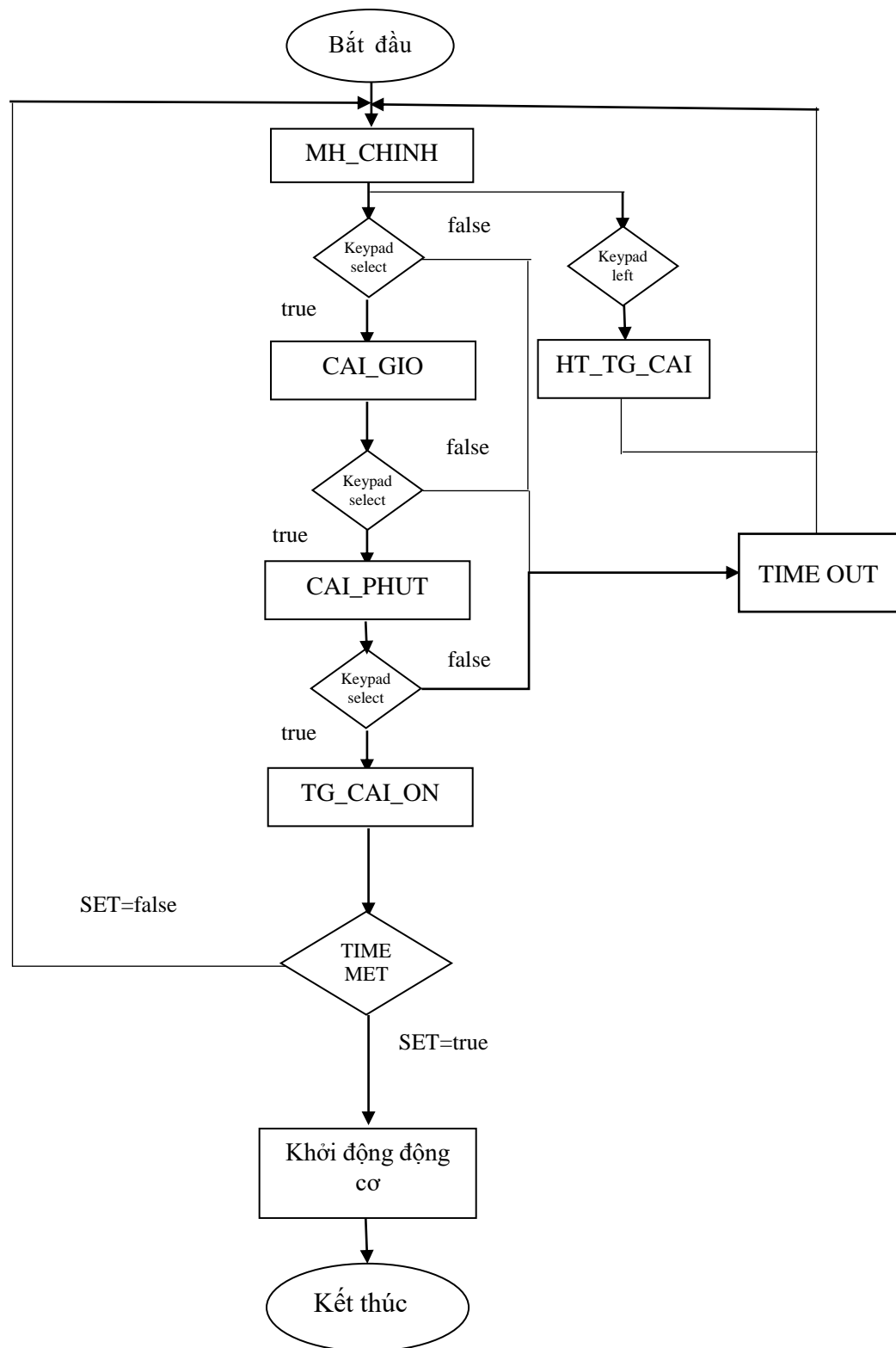
```

3.4. LẬP TRÌNH HỆ THỐNG CÀI ĐẶT THỜI GIAN, NHIỆT ĐỘ, ĐỘ ẨM, CẢM BIẾN MƯA ĐIỀU KHIỂN ĐÓNG CẮT ĐỘNG CƠ.

3.4.1.Thiết kế hệ thống điều khiển đóng cắt động cơ.

Để khởi động đóng cắt động cơ ta thường dùng relay dùng các tiếp điểm đóng cắt điều khiển động cơ. Đầu kích relay ta pin vào Arduino mega cho phù hợp và chính xác như lập trình. Trong vòng lặp việc đặt thời gian điều khiển động cơ như lưu đồ thuật toán chính cho bên dưới. Việc điều chỉnh động cơ phụ thuộc vào đối tượng yêu cầu mức thời gian phù hợp. Việc cài đặt ở đây hoàn toàn bằng các nút bấm trên LCD theo người sử dụng hoặc mặc định thời gian hoạt động theo đối tượng được đặt ra do người lập trình.

Lưu đồ thuật toán được thể hiện trên hình 3.2:



Hình 3.4 : Lưu đồ thuật toán sơ đồ hệ thống tưới cây tự động.

3.4.2. Tổng thể mạch đang hoạt động khi đã cấp nguồn và nạp Code Code lập trình đầy đủ bằng IDE cho mạch Arduino mega 2560

```
#include "DHT.h"
#include <Wire.h> // Required by RTCLib
#include <LiquidCrystal.h> // Thư viện LCD
#include <LCDKeypad.h> //Thư viện keyboard LCD
#include "RTCLib.h" //Thư viện module thời gian thực
```

```
byte doC[8] = { //Hiển thị đơn vị độ C
  0b11100,
  0b10100,
  0b11100,
  0b00000,
  0b01111,
  0b11000,
  0b11000,
  0b01111
};
```

```
byte mua[8] = { //Hiển thị trạng thái mưa
  0b01110,
  0b11111,
  0b11111,
  0b00000,
  0b10101,
  0b00000,
  0b10101,
  0b10101,
```

```

    0b000000,
};

byte nang[8] = {      //Hiển thị trời không mưa
    0b00100,
    0b10101,
    0b01110,
    0b11111,
    0b01110,
    0b10101,
    0b00100,
    0b000000,
};

```

```

byte chuong[8] = {      //Báo hiển thị đang hẹn giờ
    0b01110,
    0b11111,
    0b11111,
    0b11111,
    0b11111,
    0b11111,
    0b000000,
    0b00100,
    0b000000,
};

```

```

#define DHTPIN 22    // Pin in/out tín hiệu data cảm biến nhiệt độ độ ẩm
#define DHTTYPE DHT21 // DHT 21 (AM2301)
#define TIME_OUT 5 // One of the system's FSM transitions

```

```

#define TIME_MET 6 // One of the system's FSM transitions
#define OFF 33
#define BUZZER_PIN 25 // Pin điều khiển động cơ

// Các mệnh đề khác của hệ thống
enum states
{
    MH_CHINH,          // Hiển thị thời gian độ ẩm nhiệt độ cảm biến mưa

    TG_CAI_ON,          // Hiển thị thời gian độ ẩm nhiệt độ cảm biến mưa
                        // và bật hẹn giờ
    HT_TG_CAI,          // Hiển thị thời gian cài

    CAI_GIO,            // Cài hẹn giờ cho hệ thống khởi động
                        // Nếu không cài sẽ timeout ra màn hình chính sau 5 s
    CAI_PHUT,           // Cài hẹn phút cho hệ thống khởi động
                        // Nếu không cài sẽ timeout ra màn hình chính sau 5 s
};

// Khởi tạo LCD keyboard
// Xử lý màn hình và các nút trên LCD
LCDKeypad lcd;
// Khởi tạo thời gian DS1307(DS3231 tương tự)
// Tạo qua các thời gian qua DS1307 Real-Time Clock
RTC_DS1307 RTC;
int rainSensor = 23; // Chân tín hiệu cảm biến mưa ở chân digital 23
states state; // giữ trạng thái bắt đầu một hệ thống

```

```

int8_t button; // Trạng thái giữ các nút ấn
uint8_t MOHours = 8, MOMinutes = 0; // thời gian mặc định ban đầu
uint8_t tmpHours;
uint8_t Hours;
uint8_t tmpMinutes;
uint8_t Minutes;
boolean set = true; // trạng thái hẹn mặc định mở
boolean MO1;
unsigned long timeRef;

DateTime now; // Giữ thông tin thời gian hiện tại
DHT dht(DHTPIN, DHTTYPE);
void setup()

{
    pinMode(BUZZER_PIN, OUTPUT); // Buzzer pin tín hiệu mở
    digitalWrite(BUZZER_PIN, 0);
    pinMode(rainSensor, INPUT); /* Đặt chân cảm biến mưa là INPUT, vì tín
hiệu sẽ được truyền đến cho Arduino */
    timeRef = millis();
    // Khởi tạo các chương trình LCD và RTC
    lcd.begin(16, 2);
    Wire.begin();
    RTC.begin();
    state = MH_CHINH; // Trạng thái đầu của hệ thống

    // RTC.adjust(DateTime(__DATE__, __TIME__));
    lcd.createChar(4, doC); //tạo hiển thị trên LCD độ C

```

```

    lcd.createChar(5, mua);
    lcd.createChar(6, chuong);
    lcd.createChar(7, nang);

    dht.begin();
}

// Vòn lắp các hệ thống
void loop()
{
    timeRef = millis();
    // Các hệ thống khởi tạo chương trình
    switch (state)

    {
        case MH_CHINH:

            showTime();

            break;

        case TG_CAI_ON:

            showTime();

            checkMOTime();

```

```

        break;

    case HT_TG_CAI:

        showMOTime();

        break;

    case CAI_GIO:

        setMOHours();

        if ( state != CAI_PHUT ) break;

    case CAI_PHUT:

        setMOMinutes();

        break;
}

// Chờ khoảng 1s cho các lần bấm nút
// Nếu 1 nút được nhấn nó sẽ chạy cho tới khi được nhả sau đó mới chuyển
đổi trạng thái
while ( (unsigned long)(millis() - timeRef) < 970 )
{
    if ( (button = lcd.button()) != KEYPAD_NONE )
    {

```

```

while ( lcd.button() != KEYPAD_NONE );
transition(button);
break;
}
}
}

```

// Các hệ thống được hiển thị khi kích hoạt các nút ấn

//sau đó chuyển đổi trạng thái khi được lập trình

```
void transition(uint8_t trigger)
```

```

{
int value = digitalRead(rainSensor);
float h = dht.readHumidity();
float t = dht.readTemperature();
switch (state)
{
case MH_CHINH:
if ( trigger == KEYPAD_LEFT ) state = HT_TG_CAI;
else if ( trigger == KEYPAD_RIGHT ) { set = true; state =
TG_CAI_ON; }
else if ( trigger == KEYPAD_SELECT ) state = CAI_GIO;
break;

case TG_CAI_ON:
if ( trigger == KEYPAD_LEFT ) state = HT_TG_CAI;
else if ( trigger == KEYPAD_RIGHT ) { set = false; state =
MH_CHINH; }
else if ( trigger == KEYPAD_SELECT ) state = CAI_GIO;

```

```

        else if ( trigger == TIME_MET ) {if(value == 1 )
{analogWrite(BUZZER_PIN, 220);delay(30000);analogWrite(BUZZER_PIN,
0);state = MH_CHINH;}

                                else if (t>30 //
h<40                                ){analogWrite(BUZZER_PIN,
220);delay(60000);analogWrite(BUZZER_PIN, 0);state = MH_CHINH;}

                                else if (h>95) {analogWrite(BUZZER_PIN,
0);state = MH_CHINH;}

                                else if(value
==0){analogWrite(BUZZER_PIN, 0);state = MH_CHINH;}

                                }

```

break;

case HT_TG_CAI:

```

if ( trigger == TIME_OUT ) { if ( !set ) state = MH_CHINH;
else state = TG_CAI_ON; }

```

break;

case CAI_GIO:

```

if ( trigger == KEYPAD_SELECT ) state = CAI_PHUT;
else if ( trigger == TIME_OUT ) { if ( !set ) state = MH_CHINH;
else state = TG_CAI_ON; }

```

break;

case CAI_PHUT:

```

if ( trigger == KEYPAD_SELECT ) { set = true; state =
TG_CAI_ON; }
else if ( trigger == TIME_OUT ) { if ( !set ) state = MH_CHINH;

```



```

else state = TG_CAI_ON; }

break;

}

}

// Mảng hiển thị chính trên LCD
void showTime()
{
    now = RTC.now();
    const char* dayName[] = { "CN", "T2", "T3", "T4", "T5", "T6", "T7" };
    const char* monthName[] = { "1", "2", "3", "4", "5", "6", "7", "8", "9",
    "10", "11", "12" };
    lcd.clear();
    lcd.print(String(dayName[now.dayOfTheWeek()]) + " " +
        (now.day() < 10 ? "0" : "") + now.day() + "." +
        monthName[now.month()-1] + "." + now.year());
    lcd.setCursor(0,1);
    lcd.print((now.hour() < 10 ? "0" : "") + String(now.hour()) + ":" +
        (now.minute() < 10 ? "0" : "") + now.minute() + ":" +
        (now.second() < 10 ? "0" : "") + now.second() );
    lcd.setCursor(8,1);
    if (set == 1 )
    lcd.write(6);

    // Đọc nhiệt độ độ ẩm mất khoảng 250ms
    // Đọc cảm biến có thể lên tới 2s (do khả năng thay đổi giá trị cũ rất chậm)
    {

```

```

float h = dht.readHumidity();
float t = dht.readTemperature();
//F. int sensorReading = analogRead(A0);
// kiểm tra giá trị của sensor nếu sai báo failed
if (isnan(t) || isnan(h)) {
    lcd.setCursor(10,1);
    lcd.println("Failed !");
} else {
    lcd.setCursor(10,1);
    lcd.print(round(h));
    lcd.print("%");

    lcd.setCursor(13,1);
    lcd.print(round(t));
    //lcd.println(" *C");
    lcd.write(4);
}
}

//cảm biến mưa
{
    int value = digitalRead(rainSensor); //Đọc tín hiệu cảm biến mưa
    if (value == LOW) {transition(OFF); // Cảm biến đang mưa
        lcd.setCursor(15,0);
        lcd.write(5);;
    } else {
        lcd.setCursor(15,0);
        lcd.write(7);
    }
}

```

```
}  
}  
}
```

// Hiển thị thời gian cài đặt

// Thời gian hiển thị sau 3s quay lại màn hình chính

void showMOTime()

```
{  
    lcd.clear();  
    lcd.print("Thời gian cài");  
    lcd.setCursor(0,1);  
    lcd.print(String("Gio: ") + ( MOHours < 9 ? "0" : "" ) + MOHours +  
              " Phut: " + ( MOMinutes < 9 ? "0" : "" ) + MOMinutes);  
    delay(3000);  
    transition(TIME_OUT);  
}
```

// Kiểm tra thời gian hẹn đã được đáp ứng

// Nếu đáp ứng n sẽ chuyển trạng thái tiếp theo

void checkMOTime()

```
{  
    if ( now.hour() == MOHours && now.minute() == MOMinutes )  
        transition(TIME_MET);  
}
```

//Cài thời gian báo động cơ chạy

```

void setMOHours()
{
    unsigned long timeRef;
    boolean timeOut = true;

    lcd.clear();
    lcd.print("Cai thoi gian");

    tmpHours = 8;
    timeRef = millis();

    lcd.setCursor(0,1);
    lcd.print("cai gio: 8");

    while ( (unsigned long)(millis() - timeRef) < 5000 )
    {
        uint8_t button = lcd.button();
        if ( button == KEYPAD_UP )
        {
            tmpHours = tmpHours < 23 ? tmpHours + 1 : tmpHours;
            lcd.setCursor(11,1);
            lcd.print(" ");
            lcd.setCursor(11,1);
            if ( tmpHours < 10 ) lcd.print(" ");
            lcd.print(tmpHours);
            timeRef = millis();
        }
        else if ( button == KEYPAD_DOWN )

```

```

{
    tmpHours = tmpHours > 0 ? tmpHours - 1 : tmpHours;

    lcd.setCursor(11,1);
    lcd.print(" ");
    lcd.setCursor(11,1);

    if ( tmpHours < 10 ) lcd.print(" ");
    lcd.print(tmpHours);
    timeRef = millis();
}

else if ( button == KEYPAD_SELECT )
{
    while ( lcd.button() != KEYPAD_NONE );
    timeOut = false;
    break;
}
delay(150);
}

if ( !timeOut ) transition(KEYPAD_SELECT);
else transition(TIME_OUT);
}

// Quá trình cài thời gian lần 2 để kết thúc và hẹn giờ
// Nếu không nhận được thời gian cài 5s sau sẽ về màn hình chính
// Nếu cài phút kết thúc bắt đầu hẹn giờ

```

```

void setMOMinutes()
{
    unsigned long timeRef;
    boolean timeOut = true;
    uint8_t tmpMinutes = 0;

    lcd.clear();
    lcd.print("Cai thoi gian");
    timeRef = millis();
    lcd.setCursor(0,1);
    lcd.print("Cai phut: 0");

    while ( (unsigned long)(millis() - timeRef) < 5000 )
    {
        uint8_t button = lcd.button();
        if ( button == KEYPAD_UP )
        {
            tmpMinutes = tmpMinutes < 59 ? tmpMinutes + 1 : tmpMinutes;
            lcd.setCursor(13,1);
            lcd.print(" ");
            lcd.setCursor(13,1);

            if ( tmpMinutes < 10 ) lcd.print(" ");
            lcd.print(tmpMinutes);
            timeRef = millis();
        }
        else if ( button == KEYPAD_DOWN )

```

```

{
    tmpMinutes = tmpMinutes > 0 ? tmpMinutes - 1 : tmpMinutes;
    lcd.setCursor(13,1);
    lcd.print(" ");
    lcd.setCursor(13,1);

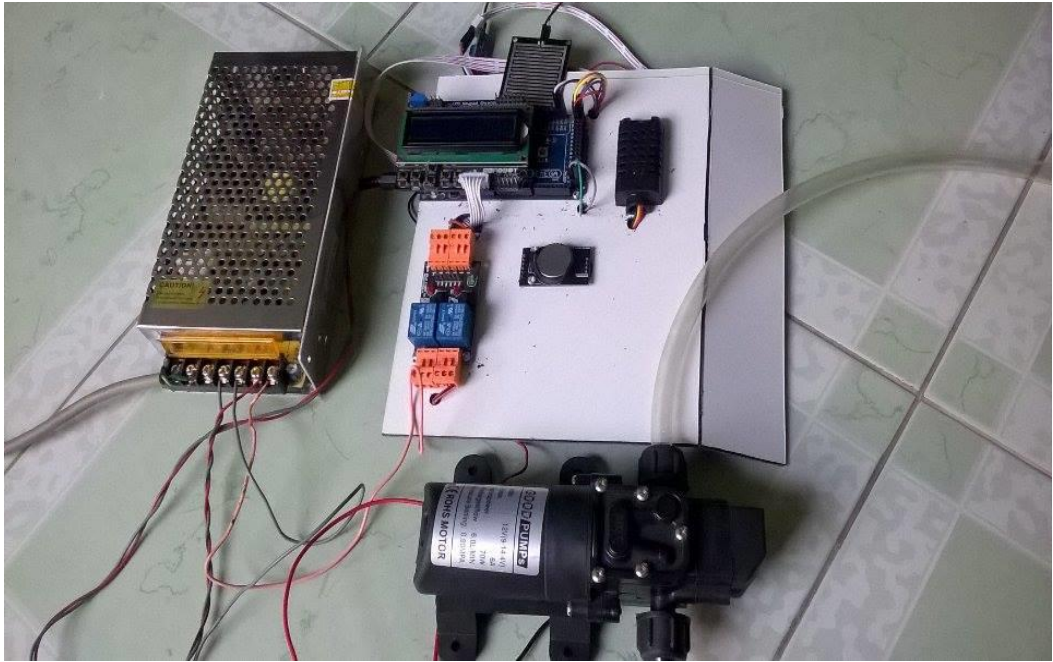
    if ( tmpMinutes < 10 ) lcd.print(" ");
    lcd.print(tmpMinutes);
    timeRef = millis();
}

else if ( button == KEYPAD_SELECT )
{
    while ( lcd.button() != KEYPAD_NONE );
    timeOut = false;
    break;
}
delay(150);
}

if ( !timeOut )
{
    MOHours = tmpHours;
    MOMinutes = tmpMinutes;
    transition(KEYPAD_SELECT);
}
else transition(TIME_OUT);
}

```

Kết quả lập trình được thể hiện trên hình 3.13 bên phải là thiết bị điều khiển và động cơ, còn bên trái là nguồn.



Hình 3.5 : Hình ảnh mạch sau khi đã nạp code và cấp nguồn.

KẾT LUẬN

Qua thời gian làm đồ án tốt nghiệp với nội dung: **Thiết kế hệ thống tưới cây tự động** em đã thiết kế và xây dựng được hệ thống tưới cây tự động gồm:

- Thiết kế đo nhiệt độ ($^{\circ}\text{C}$), độ ẩm của thời tiết dựa trên cơ sở đó để điều khiển động cơ hoạt động. VD: trong thời tiết nhiệt độ trên 30°C với điều kiện đó động cơ sẽ hoạt động lâu hơn việc tưới nước sẽ tăng lên đảm bảo cho cây trồng phát triển tốt khi gặp thời tiết xấu.

- Thiết kế cảm biến mưa ở đây tạo khả năng tiết kiệm năng lượng cũng như nguyên liệu tưới không cần thiết với khí hậu xấu. Đảm bảo cho cây tròn cũng như tạo thêm khả năng thông minh cho điều khiển.

- Thiết kế mảng thời gian thực bám sát thời gian thực tế đưa cho người dùng thông tin thời gian, cũng như tạo mảng thời gian điều khiển động cơ tưới cây tự động.

- Thiết kế LCD hiển thị đưa ra các thông tin trên LCD đưa thông tin cho người sử dụng các thông tin để đặt thời gian điều khiển hệ thống.

Trong quá trình thực hiện, lập trình cho mạch đo nhiệt độ, độ ẩm gặp phải nhiều khó khăn khác nhau như: do phải nghiên cứu nhiều tài liệu nước ngoài, datasheets,... dẫn đến nhiều chỗ dịch sai, dịch nhầm dẫn đến áp dụng các hàm, câu lệnh bị sai ý nghĩa, cấu trúc..., trong quá trình viết code gặp phải nhiều lỗi phát sinh mà không tìm ngay ra nguyên nhân cần đầu tư thời gian để giải quyết, nhiều linh kiện rất khó để tìm được thư viện chuẩn để lập trình... Quá trình lắp mạch cũng gặp phải những khó khăn nhất định tuy nhiên em đã cố gắng giải quyết được vấn đề phát sinh để hoàn thành được đề tài.

Do thời gian có hạn việc thiết kế hệ thống của em vẫn còn nhiều sai sót. Hệ thống vẫn chưa tối ưu, việc điều khiển từ xa lấy và phát thông tin chưa thành công. Phát triển các dữ liệu trên máy tính chưa phát huy được hết các tính năng tốt nhất. Em rất mong được sự ủng hộ và giúp đỡ của thầy giáo đề đề tài chúng em thực hiện được hoàn thiện hơn và có thêm nhiều cải tiến đáng kể và ứng dụng tốt hơn vào thực tiễn.

Hướng phát triển của đề tài

1. Đo và thông báo được các thông số về môi trường xung quanh

- Đo được thông số độ ẩm của môi trường xung quanh
- Đo các thông số nhiệt độ của thời tiết địa phương.
- Thông báo trạng thái thời tiết.
- Đưa các thông tin đo lên thẻ nhớ dữ liệu thông báo việc tưới tiêu.

2. Phát triển giao diện giám sát mô phỏng trên phần mềm C#.

- Phát triển lập trình qua cổng COM lập trình trên arduino
- Thêm phần điều khiển động cơ bơm, trực tiếp giao diện C#.
- Có thể up load file dữ liệu nhận trực tiếp từ cổng COM lên giao diện.

3. Đưa đề tài phát triển rộng rãi vào thực tế.

- Trước hết áp dụng trong các mô hình trồng rau, cây, hoa cảnh tại các hộ gia đình.
- Có thể phát triển trong những khu sản xuất rau trên diện rộng.
- Trên thực tế có thể điều khiển từ xa và lấy các thông tin qua máy tính

TÀI LIỆU THAM KHẢO

1. Massimo Banzi (2009), *Getting Started with Arduino*, O'Reilly Media.
2. Michael Margollis and Nicholas Weldin (2009), *Arduino Cookbook*, O'Reilly Media.
3. GS. Phạm Văn Ất (2009), *Kỹ thuật lập trình C cơ sở và nâng cao*, Nhà xuất bản Giao thông vận tải.
4. <http://learning.grobotronics.com/2014/09/arduino-lesson-14-dc-motor-1293/>
5. <http://www.adafruit.com/>
6. <http://arduino.vn/>
7. <http://arduino.cc/>