

ĐỒ ÁN TỐT NGHIỆP

THIẾT KẾ VÀ THI CÔNG MÔ HÌNH NHÀ TRỒNG RAU MẦM

Ngành: **KỸ THUẬT ĐIỆN – ĐIỆN TỬ**

Chuyên ngành: **ĐIỆN CÔNG NGHIỆP**

Giảng viên hướng dẫn : TS. TRẦN VIỆT THẮNG

Sinh viên thực hiện : Đỗ Văn Anh

MSSV : 1515021001

Sinh viên thực hiện : Nguyễn Ngọc Tú

MSSV : 1515021062

Lớp : 15HDC02

TP. Hồ Chí Minh, năm 2017

LỜI CAM ĐOAN

1. Tên đề tài: Thiết kế và thi công mô hình nhà trồng rau mầm

2. Giảng viên hướng dẫn: TS.Trần Viết Thắng

3. Họ và tên sinh viên thực hiện:

Tên: Đỗ Văn Anh MSSV: 1515021001

Tên: Nguyễn Ngọc Tú MSSV: 1515021062

Ngành: Kỹ thuật điện - điện tử.

Chuyên ngành: Điện công nghiệp.

4. Lời cam đoan:

“Chúng tôi xin cam đoan đồ án này là công trình do nhóm chúng tôi tự nghiên cứu và thực hiện với sự hướng dẫn của thầy Trần Viết Thắng

Các dữ liệu, hình ảnh và kết quả hoàn thành trình bày trong đồ án là có thật, tuân thủ đúng nguyên tắc trình bày trong đồ án tốt nghiệp, chưa từng được công bố ở các nghiên cứu khác.

Chúng tôi xin chịu trách nhiệm hoàn toàn về nghiên cứu của nhóm mình.”

Tp. Hồ Chí Minh, ngày tháng năm 2017

Sinh viên thực hiện

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trong quá trình thực hiện chúng em đã có cơ hội được tiếp xúc, thực hành trực tiếp với các kiến thức đã học và môi trường tìm hiểu thực tế bên ngoài. Song song cùng việc đó chúng em đã thu thập được nhiều tư liệu, tích lũy thêm nhiều kiến thức để hoàn thiện đề tài tốt nghiệp này.

Tuy nhiên đây là lần đầu tiên chúng em thực hiện đề tài này nên khó tránh khỏi sai sót, đồng thời trình độ cũng như kinh nghiệm còn hạn chế nên đề tài không tránh khỏi những thiếu sót. Rất mong nhận được sự nhận xét, đóng góp ý kiến từ các Quý thầy cô để đề án này ngày càng hoàn thiện hơn.

Chúng em xin chân thành cảm ơn Nhà trường, khoa Cơ - Điện - Điện tử đã tạo mọi điều kiện cho chúng em trong quá trình thực hiện cũng như hỗ trợ tài liệu, kiến thức giúp chúng em hoàn thành đề tài tốt nghiệp này.

Đặc biệt chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy Trần Viết Thắng đã nhiệt tình hướng dẫn chúng em trong quá trình hoàn thành đề tài tốt nghiệp này.

Kính chúc Quý Thầy Cô thật nhiều sức khỏe, mãi vui tươi và tràn đầy nhiệt huyết trên con đường giáo dục trồng người. Để mỗi thế hệ sinh viên ra trường sẽ trở thành những con người Tri thức – Đạo đức – Sáng tạo. Và sẽ mãi không quên hình ảnh người Thầy Cô tận tụy năm nào.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT	iii
DANH MỤC CÁC HÌNH ẢNH	iv
LỜI MỞ ĐẦU	1
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	2
1.1 ĐẶT VẤN ĐỀ.....	2
1.2 ĐỐI TƯỢNG CÂY TRỒNG	2
1.2.1. Rau mầm là gì?.....	2
1.2.2 Các lợi ích sức khỏe đem lại	3
1.2.3 Nhu cầu thị trường	3
1.2.4 Một số đặc điểm phát triển cơ bản	3
1.2.5 Một vài hình ảnh về đối tượng cây trồng	4
1.3 MỤC TIÊU CỦA ĐỀ TÀI.	5
1.4 PHƯƠNG PHÁP NGHIÊN CỨU.....	5
1.5 GIỚI HẠN ĐỀ TÀI.....	6
1.6 KẾT CẤU CỦA ĐỀ TÀI	6
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	7
2.1 ARDUINO:.....	7
2.1.1 Giới thiệu về Arduino:	7
2.1.2 Lịch sử ra đời	8
2.1.3 Một số loại Arduino trên thị trường	10
2.1.4 Sơ đồ nguyên lý Arduino Mega 2560:	14
2.2 CẢM BIẾN ĐỘ ẨM ĐẤT	14
2.3 CẢM BIẾN NHIỆT ĐỘ, ĐỘ ẨM KHÔNG KHÍ DTH11	16
2.3.1 Giới thiệu về cảm biến DTH11	16
2.3.2 Nguyên lý hoạt động	17
2.4 MÀN HÌNH LCD.....	20
2.4.1 Hình dáng và cấu tạo	20
2.4.2 Chức năng các chân	22
2.4.3 Sơ đồ kết nối LCD và MCU	23
2.5 GIAO TIẾP I2C.....	24
2.5.1 Đặc điểm của giao tiếp I2C	24
2.5.2 Truyền nhận dữ liệu Master – Slave	25
2.6 Module DS1307.....	26

2.6.1. Sử dụng Module thời gian thực DS1307	26
2.7 Module cảm biến ánh sáng BH1750.	28
CHƯƠNG 3: THIẾT KẾ VÀ THI CÔNG MÔ HÌNH THỰC NGHIỆM	29
3.1 LỰA CHỌN PHƯƠNG ÁN THIẾT KẾ.....	29
3.1.1 Các phương án thiết kế	29
3.1.2 Phương án thiết kế chính	30
3.2 THIẾT KẾ MÔ HÌNH:.....	31
3.3 THIẾT KẾ SƠ ĐỒ KHỐI	34
3.4 TÍNH TOÁN THIẾT KẾ	35
3.4.1 Thiết kế từng khối cụ thể	35
3.4.2 Sơ đồ nguyên lý toàn hệ thống:	46
3.4.2 Tổng quan về các chức năng của hệ thống	47
3.4.3 Mô hình thực tế	47
3.5 LƯU ĐỒ GIẢI THUẬT	51
3.5.1 Chương trình chính	51
3.6 MÔ TẢ HOẠT ĐỘNG CỦA CHƯƠNG TRÌNH	56
3.7 CODE CHƯƠNG TRÌNH.....	56
3.8 KẾT QUẢ THỰC NGHIỆM	57
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	59
4.1 KẾT LUẬN	59
4.1.1 Ưu điểm	59
4.1.2 Nhược điểm	59
4.2 HƯỚNG PHÁT TRIỂN	59
TÀI LIỆU THAM KHẢO	60
PHỤ LỤC	61

DANH MỤC CÁC TỪ VIẾT TẮT

ADC	Analog-to-digital converter: Mạch chuyển đổi tương tự ra số.
CPU	Central Processing Unit: Bộ xử lý trung tâm.
DIY	Do It Yourself.
GPIO	General Purpose Input Output: Cổng đầu vào và ra với mục đích cơ bản.
I2C	Inter-Integrated Circuit.
IC	Integrated Circuit: Vi mạch tích hợp.
IDE	Integrated Development Environment.
IT	Information Technology: Công nghệ thông tin.
LCD	Liquid crystal display: Màn hình tinh thể lỏng.
LED	Light Emitting Diode.
MCU	Micro-controller Unit: Khối vi điều khiển.
MISO	Master Input/Slave Output.
MIT	Massachusetts Institute of Technology: Viện Công nghệ Massachusetts.
MOSI	Master Output/Slave Input.
PLC	Programmable Logic Controller: Bộ điều khiển Logic lập trình.
SPI	Serial Peripheral Bus.
SS	Slave Select.
USB	Universal Serial Bus.
USD	United States Dollar.

DANH MỤC CÁC HÌNH ẢNH

Hình 1.1 Khay trồng cây rau mầm	4
Hình 1.2 Rau mầm	4
Hình 1.3 Rau mầm sau khi thu hoạch.	5
Hình 2. 1 Logo Arduino	Error! Bookmark not defined.
Hình 2. 2 Massimo Banzi	Error! Bookmark not defined.
Hình 2. 3 Arduino phải có giá tương đương với một bữa ăn Pizza ở ngoài	Error! Bookmark not defined.
Hình 2. 4 Sơ đồ nguyên lý Mega 2560	Error! Bookmark not defined.
Hình 2. 5 Cảm biến độ ẩm đất	Error! Bookmark not defined.
Hình 2. 6 Sơ đồ kết nối board Mega 2560 với cảm biến độ ẩm đất	Error! Bookmark not defined.
Hình 2. 7 Cảm biến DTH11	Error! Bookmark not defined.
Hình 2. 8 Sơ đồ kết nối của cảm biến DTH11 với Vi xử lý. ..	Error! Bookmark not defined.
Hình 2. 9 Giảm độ tín hiệu giao tiếp cảm biến DTH11.	17
Hình 2. 10 Bit 0.	Error! Bookmark not defined.
Hình 2. 11 Bit 1.	Error! Bookmark not defined.
Hình 2. 12 Sơ đồ kết nối board Mega 2560 với cảm biến DTH11.	Error! Bookmark not defined.
Hình 2. 13 Hình dáng của loại LCD thông dụng.	Error! Bookmark not defined.
Hình 2. 14 Sơ đồ chân của LCD.	Error! Bookmark not defined.
Hình 2. 15 Sơ đồ kết nối chân của LCD với MCU.	Error! Bookmark not defined.
Hình 2. 16 Sơ đồ giao tiếp I2C.	Error! Bookmark not defined.
Hình 2. 17 Sơ đồ kết nối board Mega 2560 với LCD có module I2C.	Error! Bookmark not defined.
Hình 2. 18 Hình ảnh module và nguyên lý DS1307.	Error! Bookmark not defined.
Hình 2. 19 Sơ đồ kết nối board Mega 2560 với module DS1307.	Error! Bookmark not defined.
Hình 2. 20 Sơ đồ kết nối board Mega 2560 với module BH1750.	Error! Bookmark not defined.
Hình 3. 1 Mô hình thiết kế 3D	31
Hình 3. 2 Hình chiếu bằng	32
Hình 3. 3 Hình chiếu đứng	32
Hình 3. 4 Hình chiếu cạnh	33
Hình 3. 5 Sơ đồ khối hệ thống.	34
Hình 3. 6 Ổ cắm 220Vac và nguồn 12Vdc - 30A.	35
Hình 3. 7 sơ đồ kết nối với các khối khác của vi điều khiển.	36

Hình 3. 8 Sơ đồ nguyên lý của cảm biến DHT11.	36
Hình 3. 9 Sơ đồ nguyên lý của Module BH1750.	37
Hình 3. 10 Sơ đồ nguyên lý DS1307.	38
Hình 3. 11 Sơ đồ nguyên lý khối phím nhấn.	39
Hình 3. 12 Sơ đồ nguyên lý khối LCD.	40
Hình 3. 13 Hình ảnh quạt thông gió trong mô hình.	42
Hình 3. 14 Sơ đồ kết nối quạt.	42
Hình 3. 15 Máy bơm mini.	43
Hình 3. 16 Sơ đồ kết nối máy bơm.	43
Hình 3. 17 Sò nóng lạnh và quạt tản nhiệt.	44
Hình 3. 18 Sơ đồ kết nối máy làm lạnh.	45
Hình 3. 19 Đèn chiếu sáng.	45
Hình 3. 20 Sơ đồ kết nối đèn.	46
Hình 3. 21 Sơ đồ nguyên lý hệ thống.	46
Hình 3. 22 Mô hình sau khi thi công.	47
Hình 3. 23 Khối nguồn 12V DC-30A.	48
Hình 3. 24 Mạch vi điều khiển Arduino Mega 2560.	48
Hình 3. 25 Phím ma trận 4X4 điều khiển và LCD hiển thị 20X4.	49
Hình 3. 26 Các cảm biến và bộ phận thực thi của hệ thống.	49
Hình 3. 27 Thiết bị làm lạnh và máy bơm.	50
Hình 3. 28 Lưu đồ giải thuật tổng quan chương trình.	56
Hình 3. 29 Kết quả thực nghiệm.	57

LỜI MỞ ĐẦU

Ngày nay khoa học công nghệ ngày càng phát triển, vi điều khiển AVR và 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 mới 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 vô cùng dễ sử dụng tương thích với ngôn ngữ C và hệ thư viện rất phong phú, được chia sẻ miễn phí. Chính vì những lý do như vậy mà 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 cùng với sự hiểu biết về các thiết bị điện tử, chúng em quyết định thực hiện đề tài: **Thiết kế và thi công mô hình nhà trồng rau mầm**. Với mục đích tìm hiểu thêm về Arduino, làm quen với các thiết bị điện tử khác và nâng cao hiểu biết cho bản thân.

Bên cạnh đó, hiện nay vấn đề thực phẩm bẩn đang xuất hiện tràn lan trên thị trường hơn bao giờ hết. Các thông tin đại chúng cũng không ngừng đưa tin về các vụ ngộ độc thực phẩm, các cơ sở sản xuất thực phẩm bẩn. Về lâu dài, sẽ ảnh hưởng đến sức khỏe và gây ra nhiều gánh nặng cho xã hội. Vì thế trong bối cảnh an toàn thực phẩm hiện nay, nhu cầu của thị trường về các sản phẩm sạch và an toàn là thiết yếu đối với mọi người. Và đó cũng là một trong các lý do chúng em chọn đề tài này.

Do kiến thức còn hạn hẹp, thêm vào đó đây là lần đầu chúng em thực hiện đề tài nên không tránh khỏi những thiếu sót, hạn chế vì thế chúng em rất mong có sự góp ý và nhắc nhở từ các Quý thầy cô để có thể hoàn thành đề tài của mình.

Chúng em xin chân thành cảm ơn thầy Trần Viết Thắng đã giúp đỡ chúng em rất nhiều trong quá trình tìm hiểu, thiết kế và hoàn thành đề tài.

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1 ĐẶT VẤN ĐỀ

Khi cuộc sống con người được nâng cao, những nhu cầu cuộc sống hằng ngày càng cao đòi hỏi phải được hỗ trợ tốt hơn. Và từ những nhu cầu thực tế đó ý tưởng về mô hình nhà vườn trồng cây nông nghiệp sạch được hình thành. Một nhà vườn cung cấp những loại thực phẩm sạch sẽ là một trong những sự hỗ trợ và giúp đỡ cho cuộc sống đầy rẫy những loại cây nông nghiệp mà người trồng không còn chú trọng chất lượng, không còn quan tâm đến người khác mà chỉ vụ lợi cho bản thân.

Ngày nay, với sự phát triển một cách nhanh chóng của ngành điện tử cũng như nhiều ngành khác thì ý tưởng về nhà vườn thông minh không còn vướng bởi rào cản công nghệ. Việc điều khiển nhà vườn thông minh thông qua smartphone hoặc máy tính tạo nên bước ngoặt lớn trong việc điều khiển tự động, không đây một cách linh hoạt, có thể nói sự phát triển không ngừng của những chiếc smartphone đã làm cho công nghệ có thêm bước tiến, việc điều khiển dễ dàng hơn.

Từ ý tưởng đó, nhu cầu về cuộc sống tốt hơn nên việc xây dựng “THIẾT KẾ VÀ THI CÔNG NHÀ TRỒNG RAU MẦM” để đáp ứng phần nào trong cuộc sống của mọi người.

1.2 ĐỐI TƯỢNG CÂY TRỒNG

1.2.1. Rau mầm là gì?

Rau mầm là những cây non mới mọc mầm sống không cần đất. Hầu hết các trường hợp hạt giống tự nhiên đều đủ dinh dưỡng để nảy mầm và ra hai lá mầm nhỏ trước khi cần ánh sáng và đất để phát triển. Rau mầm rất mọng nước, mềm và được sử dụng toàn cây, bao gồm cả lá, thân và rễ mầm sạch. Rau mầm cũng là lựa chọn tuyệt vời làm bánh sandwich và salad.

1.2.2 Các lợi ích sức khỏe đem lại

Rau mầm dễ tiêu hóa, có nhiều vitamin, chất khoáng hữu cơ, axit amin, chất đạm, các enzym có ích, và các chất phytochemical, do các chất này cần thiết để cho cây mới nảy mầm có thể phát triển. Các chất dinh dưỡng này rất quan trọng đối với sức khỏe con người.

1.2.3 Nhu cầu thị trường

Giá:

- Hạt giống rau mầm các loại có giá từ: 16.000 VNĐ – 18.000 VNĐ gói 20g.
- Rau mầm các loại có giá từ: 60.000 VNĐ – 80.000 VNĐ gói 1kg khi cung cấp cho các siêu thị như Metro, Maximart, Co.op mart và đặc biệt là Thế Giới Di Động đang tăng nhanh về số lượng bán hàng xanh trong thời gian gần đây.

1.2.4 Một số đặc điểm phát triển cơ bản

- Thời gian sinh trưởng: 5 – 7 ngày.
- Năng suất: 450 - 650g rau/20g hạt, tương đương 10 – 15 hộp.
- Do thời gian sinh trưởng ngắn nên không cần các loại thuốc.
- Thích nghi được với khí hậu Việt Nam.

1.2.5 Một vài hình ảnh về đối tượng cây trồng



Hình 1.1 Khay trồng cây rau mầm



Hình 1.2 Rau mầm



Hình 1.3 Rau mầm sau khi thu hoạch.

1.3 MỤC TIÊU CỦA ĐỒ ÁN.

Tạo ra một khu vườn có thể:

- Giám sát các thông số: nhiệt độ môi trường, độ ẩm đất và không khí,...
- Điều khiển các hệ thống tưới nước, nhiệt độ,...
- Giá thành hợp lí.

1.4 PHƯƠNG PHÁP NGHIÊN CỨU.

Do đây là một đồ án sản phẩm, nên chúng em đã áp dụng phương pháp nghiên cứu lý thuyết, nghiên cứu thực nghiệm chạy thử và hoàn thiện chương trình.

Ý nghĩa nghiên cứu:

Nếu thực hiện thành công đề tài này sẽ mang lại ý nghĩa về cả thực tiễn và lý luận về cách nuôi trồng cây mầm.

• Ý nghĩa lý luận:

Là tài liệu nghiên cứu, tham khảo nhanh, dễ hiểu, thiết thực cho các bạn sinh viên, những người quan tâm về đề tài này.

• **Ý nghĩa thực tiễn:**

Góp phần giúp cho các bạn sinh viên mới nói chung và các bạn sinh viên khoa Cơ - Điện - Điện tử nói riêng thấy rõ được ý nghĩa thực tế và thêm yêu thích chuyên ngành mình đã chọn.

1.5 GIỚI HẠN CỦA ĐỒ ÁN

- Nghiên cứu cho đối tượng là các loại cây rau mầm.
- Sử dụng công nghệ vi điều khiển Arduino.

1.6 KẾT CẤU CỦA ĐỒ ÁN

Đề tài gồm 4 chương:

Chương 1: Tổng quan về đề tài.

Chương 2: Cơ sở lý thuyết.

Chương 3: Thiết kế và thi công mô hình thực nghiệm.

Chương 4: Kết luận và hướng phát triển đề tài.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 ARDUINO:

Với đề tài này chúng ta sẽ sử dụng Vi điều khiển Arduino để xử lý và điều khiển hệ thống. Ngoài ra, các phương án khác như sử dụng IC số sẽ được trình bày và phân tích rõ hơn ở chương 3.

2.1.1 Giới thiệu về Arduino:

Arduino là một board mạch vi xử lý, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. 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. Một khía cạnh quan trọng của Arduino là các kết nối tiêu chuẩn của nó, 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.



Hình 2. 1 Logo Arduino

Môi trường phát triển tích hợp (IDE) của Arduino là một ứng dụng cross-platform (nền tảng) được viết bằng Java, và từ IDE này sẽ được sử dụng cho Ngôn ngữ lập trình xử lý (Processing programming language) và project Wiring. Nó được thiết kế để dành cho những người mới tập làm quen với lĩnh vực phát triển phần mềm. Một chương trình hoặc code viết cho Arduino được gọi là một Sketch.

Các chương trình Arduino được viết bằng C hoặc C++. Arduino IDE đi kèm với một thư viện phần mềm được gọi là "Wiring", từ project Wiring gốc, có thể giúp các thao tác input/output được dễ dàng hơn.

2.1.2 Lịch sử ra đời

Mới đầu Arduino được tạo ra để giải quyết bài toán: Làm thế nào để sinh viên có thể tạo ra được một thiết bị điện tử một cách nhanh chóng.

Vào năm 2002, Banzi, một giáo sư phần mềm đã được đưa về IDII (Viện Thiết Kế Vật Lý Ivrea, Italia) để thúc đẩy một lĩnh vực còn non trẻ gọi là “máy tính vật lý” với một ngân sách hạn hẹp, thời gian hạn chế và các công cụ rất ít.



Hình 2. 2 Massimo Banzi

Giống như nhiều đồng nghiệp của mình, Banzi dựa vào Stamp BASIC, một vi điều khiển được tạo ra bởi công ty California Parallax mà các kỹ sư đã sử dụng trong khoảng một thập kỷ. Nhưng với Stamp BASIC có hai vấn đề mà Banzi phát hiện ra: nó không có đủ khả năng xử lý cho một số dự án của sinh viên, và nó có giá khoảng 100 USD khá cao so với sinh viên. Ông cũng cần một cái gì đó mà có thể chạy trên các máy tính Macintosh, thứ phổ biến với các sinh viên IDII.

Mẫu thử nghiệm ban đầu, được thực hiện vào năm 2005, là một thiết kế đơn giản, và chưa được gọi là Arduino. Massimo Banzi sử dụng cái tên Arduino vào một năm sau đó.

Để làm bo Arduino, nhóm đã đặt ra một giá cụ thể để Arduino trở nên “thân thiện” với sinh viên và mục tiêu của họ là 30\$. “Nó phải tương đương với một bữa ăn Pizza ở ngoài” Banzì nói.

Với mức giá 30\$ cho một bo Arduino có khoảng 20 ngõ I/O có thể tương tác và điều khiển chừng ấy thiết bị, tính chất nguồn mở từ phần cứng tới phần mềm, giao diện lập trình 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. Arduino đã thực sự gây sóng gió trên thị trường người dùng DIY (Do It Yourself) trên toàn thế giới trong vài năm gần đây.

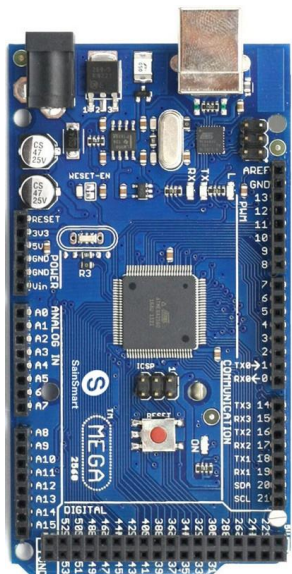




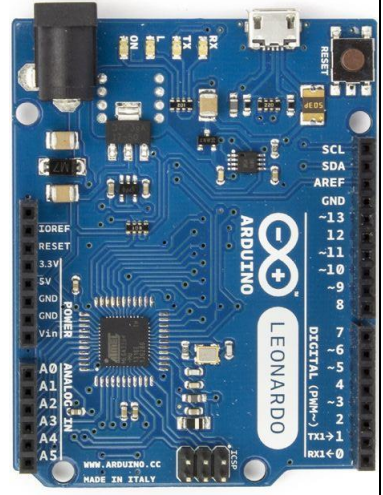
Hình 2. 3 Arduino phải có giá tương đương với một bữa ăn Pizza ở ngoài

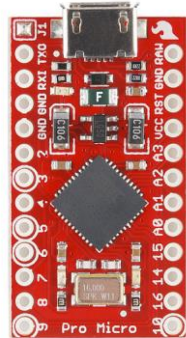
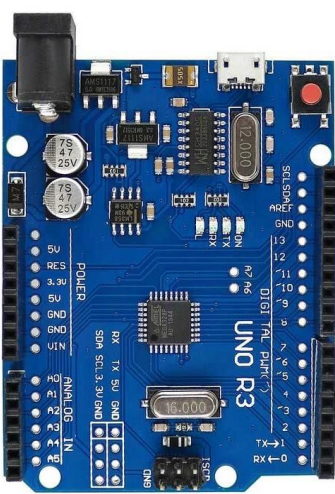
Về cái tên Arduino, đây là tên được đặt theo tên của quán rượu Bar di Re Arduino, nơi mà Bazi và các cộng sự thường lui tới trong quá trình làm nên bo mạch này. Đây là một điều thật sự thú vị và bất ngờ.

2.1.3 Một số loại Arduino trên thị trường

Bảng 2.1 Một số loại Arduino trên thị trường

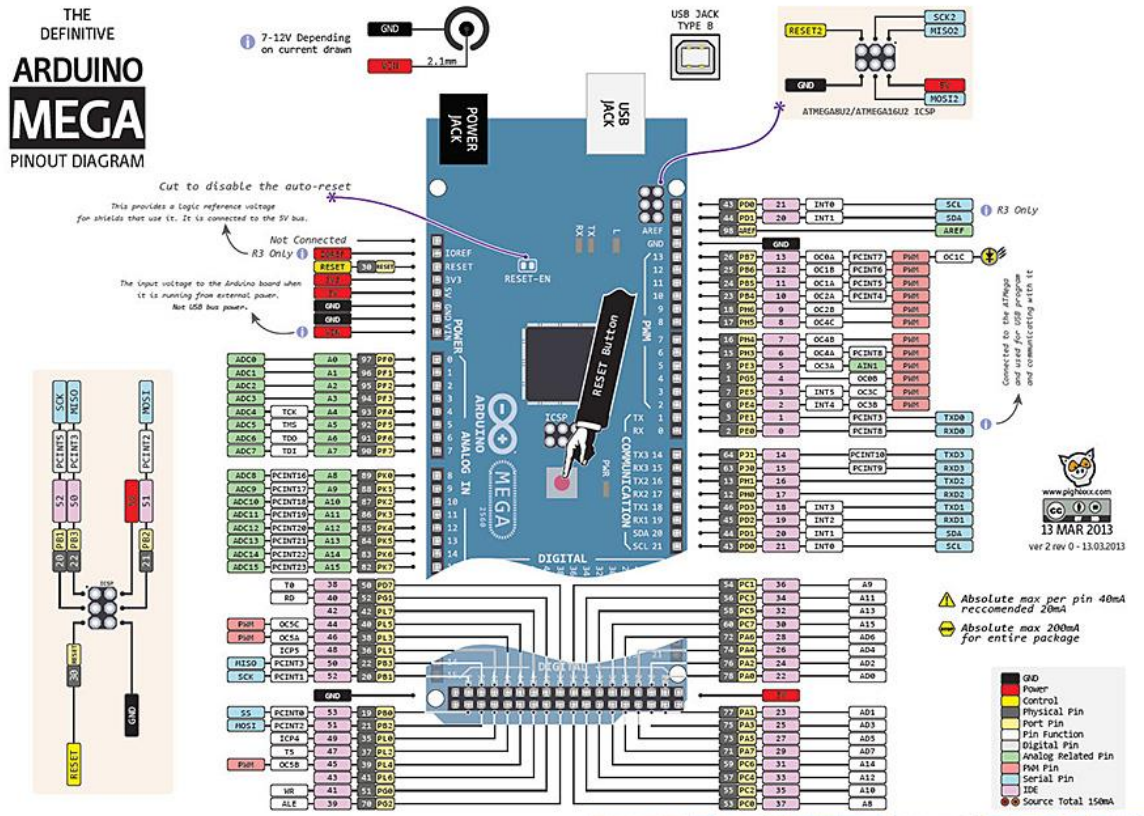
TT	Tên bo Arduino	Thông số kỹ thuật	Hình dáng
	Mega 2560 R3	<p>Vi điều khiển: ATmega2560</p> <p>Điện áp hoạt động: 5V</p> <p>Điện áp ngõ vào DC: 7-12V</p> <p>Số chân Digital: 54 (15 chân PWM)</p> <p>Số chân Analog: 16</p> <p>Bộ nhớ Flash: 256 KB, 8KB sử dụng cho Bootloader</p> <p>SRAM: 8 KB</p> <p>EEPROM: 4 KB</p> <p>Xung clock: 16 MHz</p>	
	Due	<p>Vi điều khiển: AT91SAM3X8E</p> <p>Điện áp hoạt động: 3.3V</p> <p>Điện áp cung cấp: 7-12V</p> <p>Số chân Digital: 54</p> <p>Số chân ngõ vào Analog: 12</p> <p>Số chân ngõ ra Analog: 2</p> <p>Tổng dòng ngõ ra trên các chân I/O: 130mA</p> <p>Bộ nhớ Flash: 512KB</p> <p>SRAM: 96KB</p> <p>Xung clock: 84MHz</p> <p>Kích thước: 101.5 x 53.5 mm</p>	

		Khối lượng: 36g	
	Uno R3	<p>Chip sử dụng: ATmega328</p> <p>Điện áp hoạt động: 5V</p> <p>Điện áp cung cấp: 7-12V</p> <p>Điện áp vào: 6-20V</p> <p>Digital I/O Pins 14 (of which 6 provide PWM output)</p> <p>Analog Input Pins: 6</p> <p>DC Current per I/O Pin: 40 mA</p> <p>DC Current for 3.3V Pin 50mA</p> <p>Flash Memory</p> <p>32 KB (ATmega328) of which 0.5 KB used by bootloader</p> <p>SRAM</p> <p>2 KB (ATmega328) EEPROM</p> <p>1 KB (ATmega328)</p> <p>Clock Speed: 16 MHz</p>	
	Leonardo	<p>Vi điều khiển: ATmega32u4</p> <p>Điện áp hoạt động: 5V</p> <p>Điện thế ngõ vào DC: 7-12V</p> <p>Số chân Digital: 20</p> <p>Số kênh PWM: 7</p> <p>Số kênh vào Analog: 12</p> <p>Bộ nhớ Flash: 32 KB (ATmega32u4), 4KB sử dụng cho Bootloader.</p> <p>SRAM: 2.5 KB (ATmega32u4)</p> <p>EEPROM: 1 KB</p>	

		(ATmega32u4) Xung lock: 16 MHz	
	Pro Micro	Sử dụng chip Atmega32U4 Điện áp cung cấp: 6V ~ 12V DC (chân Raw) Điện áp hoạt động: 5V Hỗ trợ từ Arduino IDE V1.0.1 trở lên Tích hợp cổng USB Micro trên board Tần số hoạt động: 16MHz Chống cấp ngược điện áp Có Led hiển thị nguồn và báo tình trạng Kích thước: 33 x 18 x 6mm	
	Uno R3 (SMD)	Chip điều khiển chính: ATmega328 Nguồn nuôi mạch: 5V Số chân Digital: 14 (hỗ trợ 6 chân PWM) Số chân Analog: 6 Dòng ra trên chân digital: tối đa 40 mA Dòng ra trên chân 5V: 500 mA Dòng ra trên chân 3.3V: 50 mA Dung lượng bộ nhớ Flash: 32 KB (ATmega328) SRAM: 2 KB (ATmega328)	

		EEPROM: 1 KB (ATmega328) Tốc độ: 16 MHz	
--	--	--	--

2.1.4 Sơ đồ nguyên lý Arduino Mega 2560:



Hình 2. 4 Sơ đồ nguyên lý Mega 2560

- Sơ đồ nguyên lý của board Arduino Mega 2560.
- Thẻ số lượng chân kết nối.
- Thẻ hiện chức năng của của từng chân.
- Nguồn VCC, GND và nút Reset của board

2.2 CẢM BIẾN ĐỘ ẨM ĐẤT

Mạch cảm biến độ ẩm của đất gồm 2 bộ phận:

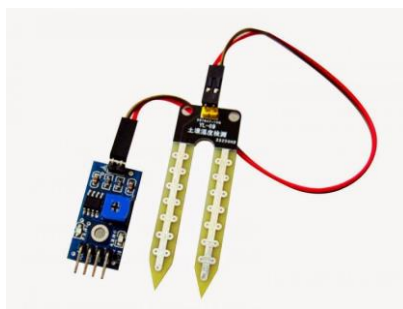
- Bộ phận cảm biến độ ẩm được gắn thẳng xuống đất.
- Bộ phận điều chỉnh độ nhạy nằm phía trên mặt đất.

Mạch cảm biến độ ẩm của đất hoạt động bằng cách so sánh độ ẩm từ phần nằm dưới mặt đất với giá trị định trước (giá trị này thay đổi được thông qua 1 biến trở màu xanh) từ đó phát ra tín hiệu đóng/ngắt rơ le qua chân D0.

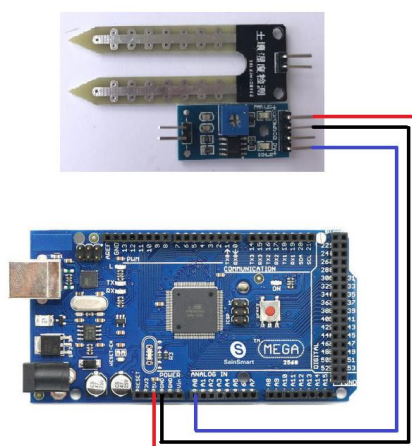
Khi đất khô, chân D0 sẽ được giữ ở mức cao, có thể sử dụng để kích rơ le và chạy máy bơm. Khi đất ẩm, đèn LED màu đỏ sẽ sáng lên, chân D0 sẽ được giữ ở mức thấp, do đó ngắt rơ le.

Chân A0 được sử dụng để kết nối với ADC của vi điều khiển trả về giá trị analog 1024 bit.

Mạch hoạt động với nguồn 5V.



Hình 2. 5 Cảm biến độ ẩm đất

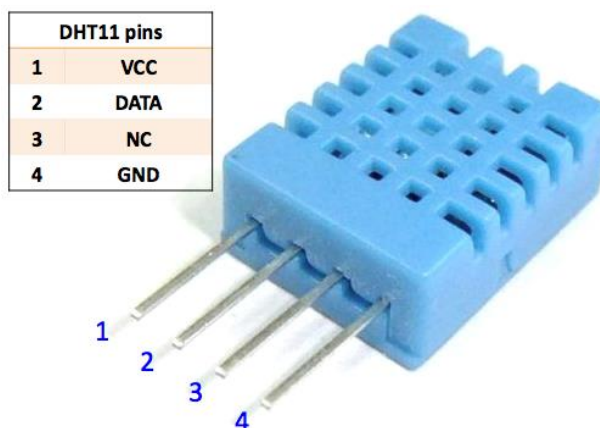


Hình 2. 6 Sơ đồ kết nối board Mega 2560 với cảm biến độ ẩm đất

2.3 CẢM BIẾN NHIỆT ĐỘ, ĐỘ ẨM KHÔNG KHÍ DHT11

2.3.1 Giới thiệu về cảm biến DHT11

DHT11 là một cảm biến nhiệt độ và độ ẩm. Nó ra đời sau và được sử dụng thay thế cho dòng SHT1x ở những nơi không cần độ chính xác cao về nhiệt độ và độ ẩm.



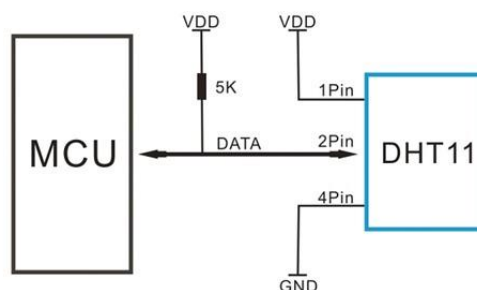
Hình 2. 7 Cảm biến DHT11.

DHT11 có cấu tạo 4 chân như hình 2.8. Nó sử dụng giao tiếp số theo chuẩn 1 dây.

Thông số kỹ thuật

- Đo độ ẩm: 20%-95%.
- Nhiệt độ: 0-50°C.
- Sai số độ ẩm $\pm 5\%$.
- Sai số nhiệt độ: $\pm 2^\circ\text{C}$.

2.3.2 Nguyên lý hoạt động

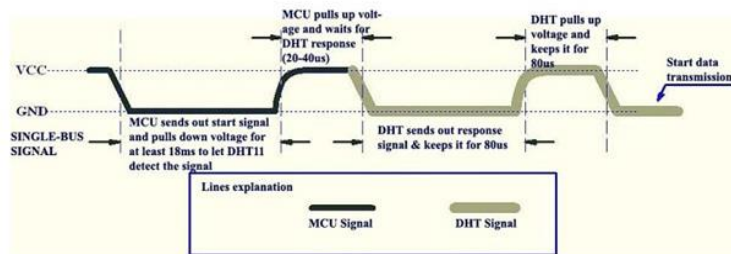


Hình 2. 8 Sơ đồ kết nối của cảm biến DHT11 với Vi xử lý.

Để có thể giao tiếp với DHT11 theo chuẩn 1 chân vi xử lý thực hiện theo 2 bước:

- Gửi tín hiệu muốn đo (Start) tới DHT11, sau đó DHT11 xác nhận lại.
- Khi đã giao tiếp được với DHT11, Cảm biến sẽ gửi lại 5 byte dữ liệu và nhiệt độ đo được.

- **Bước 1:** gửi tín hiệu Start.



Hình 2. 9 Giải đồ tín hiệu giao tiếp cảm biến DTH11.

+ MCU thiết lập chân DATA là Output, kéo chân DATA xuống 0 trong khoảng thời gian >18ms. Trong Code em để 25ms. Khi đó DHT11 sẽ hiểu MCU muốn đo giá trị nhiệt độ và độ ẩm.

+ MCU đưa chân DATA lên 1, sau đó thiết lập lại là chân đầu vào.

Sau khoảng 20-40us, DHT11 sẽ kéo chân DATA xuống thấp. Nếu >40us mà chân DATA ko được kéo xuống thấp nghĩa là ko giao tiếp được với DHT11.

+ Chân DATA sẽ ở mức thấp 80us sau đó nó được DHT11 kéo lên cao trong 80us. Bằng việc giám sát chân DATA, MCU có thể biết được có giao tiếp được với DHT11 ko. Nếu tín hiệu đo được DHT11 lên cao, khi đó hoàn thiện quá trình giao tiếp của MCU với DHT.

- **Bước 2:** đọc giá trị trên DHT11.

DHT11 sẽ trả giá trị nhiệt độ và độ ẩm về dưới dạng 5 byte. Trong đó:

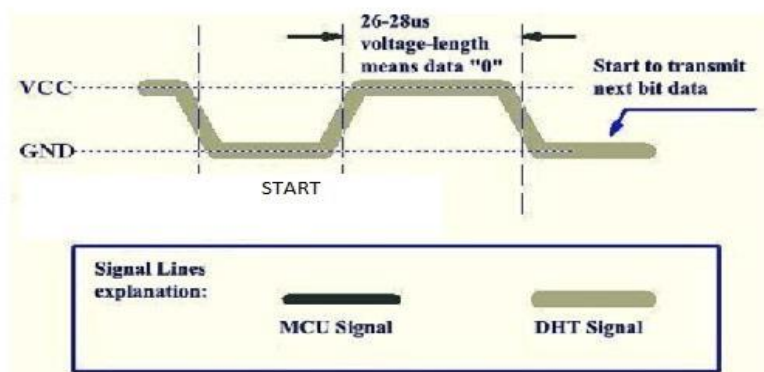
- Byte 1: giá trị phần nguyên của độ ẩm (RH%)
- Byte 2: giá trị phần thập phân của độ ẩm (RH%)
- Byte 3: giá trị phần nguyên của nhiệt độ (TC)
- Byte 4 : giá trị phần thập phân của nhiệt độ (TC)
- Byte 5 : kiểm tra tổng.

Nếu Byte 5 = (8 bit) (Byte1 +Byte2 +Byte3 + Byte4) thì giá trị độ ẩm và nhiệt độ là chính xác, nếu sai thì kết quả đo không có nghĩa.

+ Đọc dữ liệu:

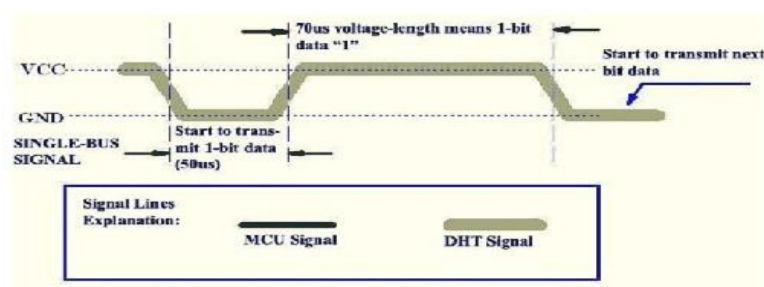
Sau khi giao tiếp được với DHT11, DHT11 sẽ gửi liên tiếp 40 bit 0 hoặc 1 về MCU, tương ứng chia thành 5 byte kết quả của Nhiệt độ và độ ẩm.

- Bit 0:



Hình 2. 10 Bit 0.

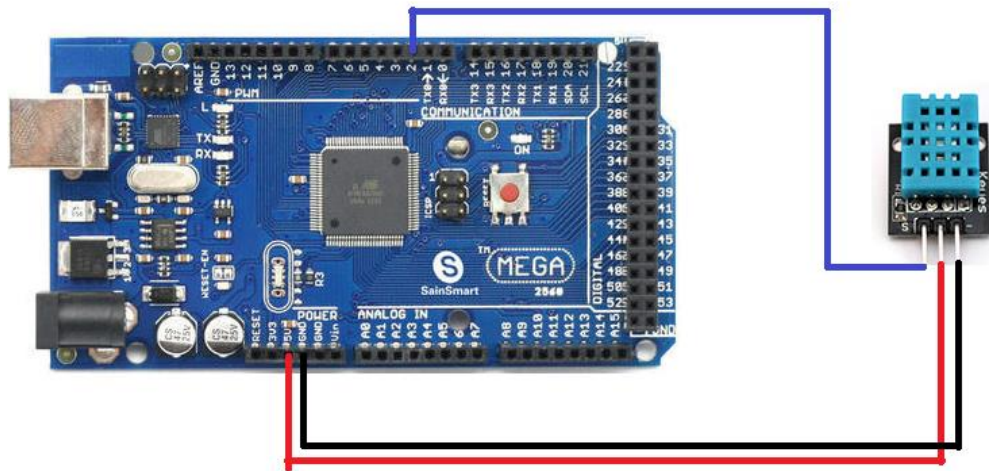
- Bit 1:



Hình 2. 11 Bit 1.

Sau khi tín hiệu được đưa về 0, ta đợi chân DATA của MCU được DHT11 kéo lên 1. Nếu chân DATA là 1 trong khoảng 26-28us thì là 0, còn nếu tồn tại 70us là 1. Do đó trong lập trình ta bắt sườn lên của chân DATA, sau đó delay 50us. Nếu giá trị đo được là 0 thì ta đọc được bit 0, nếu giá trị đo được là 1 thì giá trị đo được là 1.

Cứ như thế ta đọc các bit tiếp theo.



Hình 2. 12 Sơ đồ kết nối board Mega 2560 với cảm biến DTH11.

2.4 MÀN HÌNH LCD

- Ưu điểm: hiển thị dễ dàng, có thể linh động hơn trong việc hiển thị, kết nối đơn giản mạch điện không phức tạp ...
- Nhược điểm: không thu hút được sự chú ý bằng led 7 thanh, giá thành cao ...

2.4.1 Hình dáng và cấu tạo

- Có rất nhiều loại LCD được sử dụng trong nhiều ứng dụng của vi điều khiển. LCD có rất nhiều ưu điểm so với các dạng hiển thị khác: nó có khả năng hiển thị kí tự đa dạng, trực quan (chữ, số, kí tự đồ họa) dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn ít tài nguyên hệ thống ...
- Có rất nhiều loại LCD với nhiều hình dáng và kích thước khác nhau.



Hình 2.13 Hình dáng của loại LCD thông dụng.

Khi sản xuất LCD, nhà sản xuất đã tích hợp chip điều khiển (HD44780) bên trong lớp vỏ và chỉ đưa các chân giao tiếp cần thiết. Các chân này được đánh số thứ tự và đặt tên như hình 2.16:



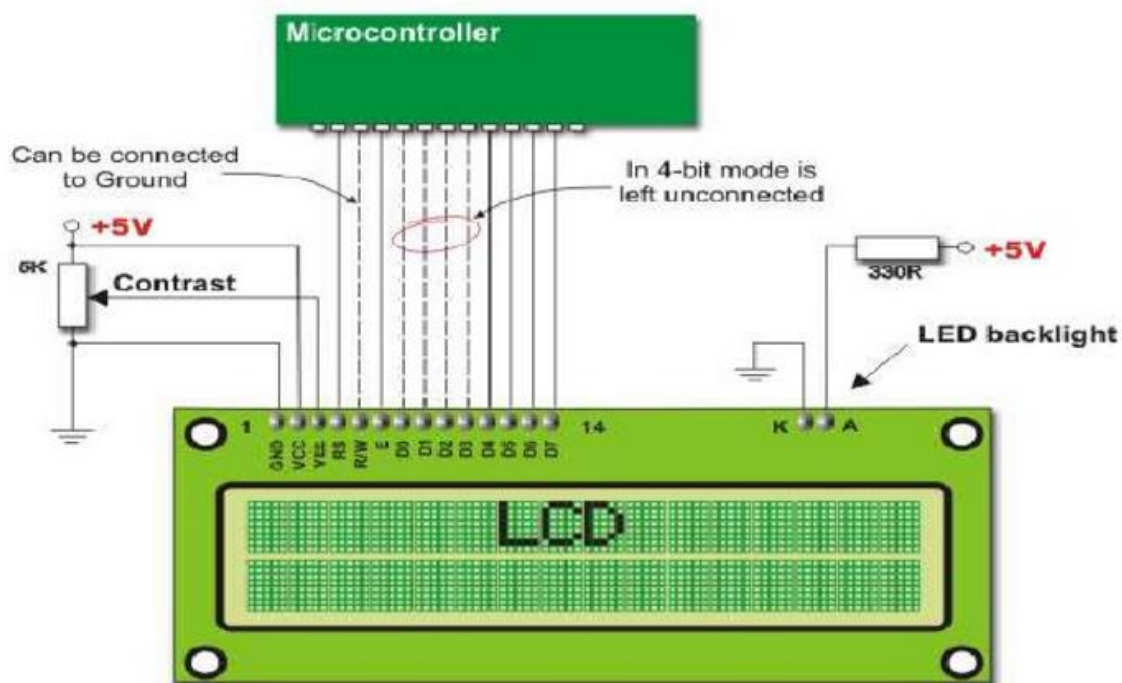
Hình 2.14 Sơ đồ chân của LCD.

2.4.2 Chức năng các chân

Chân	Ký hiệu	Mô tả
1	V _{ss}	Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển.
2	V _{DD}	Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC=5V của mạch điều khiển.
3	V _{EE}	Điều chỉnh độ tương phản của LCD.
4	RS	Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi. + Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read). + Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD.
5	R/W	Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc.
6	E	Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E. + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào(chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E. + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high transition) ở chân E và được LCD giữ ở

		bus đến khi nào chân E xuống mức thấp.
7 - 14	DB0 - DB7	Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này : + Chế độ 8 bit : Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7. + Chế độ 4 bit : Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7.
15	-	Nguồn dương cho đèn nền
16	-	GND cho đèn nền

2.4.3 Sơ đồ kết nối LCD và MCU



Hình 2. 15 Sơ đồ kết nối chân của LCD với MCU.

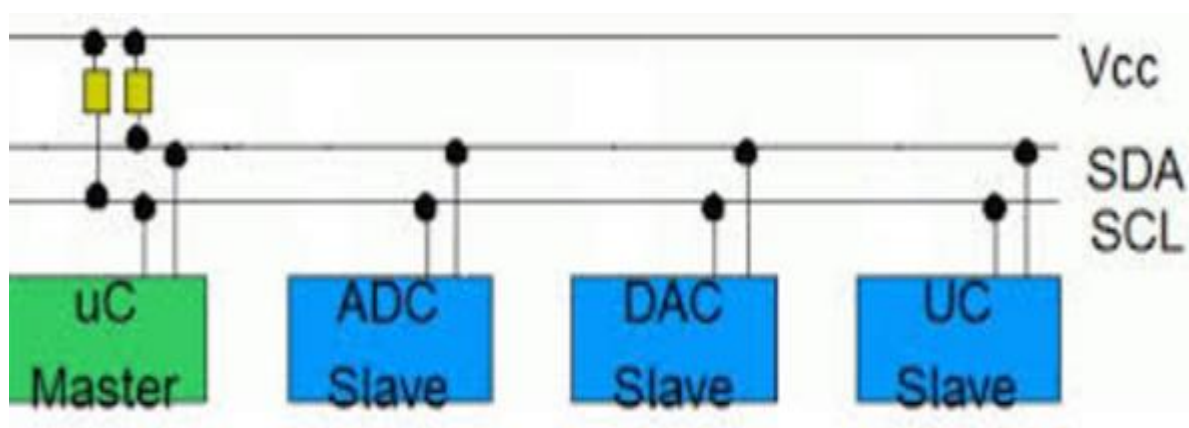
2.5 GIAO TIẾP I2C

I2C là viết tắt của từ tiếng Anh “Inter-Integrated Circuit”, là một loại bus nối tiếp được phát triển bởi hãng sản xuất linh kiện điện tử Philips.

Ban đầu, loại bus này chỉ được dùng trong các linh kiện điện tử của Philips. Sau đó, do tính ưu việt và đơn giản của nó, I2C đã được chuẩn hóa và được dùng rộng rãi trong các mô đun truyền thông nối tiếp của vi mạch tích hợp ngày nay.

2.5.1 Đặc điểm của giao tiếp I2C

Một giao tiếp I2C gồm có 2 dây: Serial Data (SDA) và Serial Clock (SCL). SDA là đường truyền dữ liệu 2 hướng, còn SCL là đường truyền xung đồng hồ và chỉ theo một hướng.



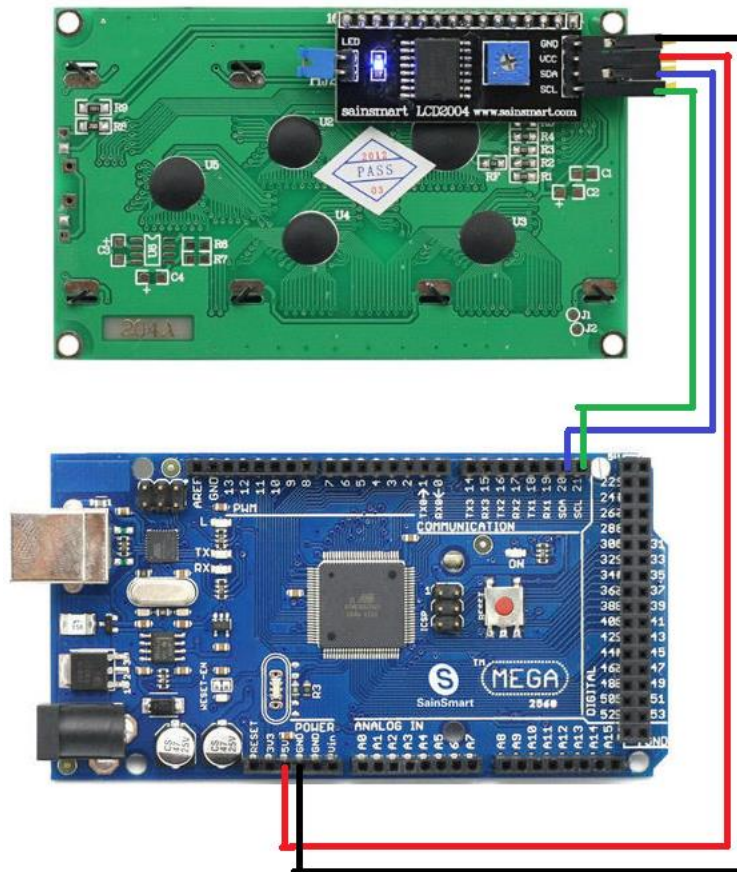
Hình 2. 16 Sơ đồ giao tiếp I2C.

Mỗi dây SDA hay SCL đều được nối với điện áp dương của nguồn cấp thông qua một điện trở kéo lên. Sự cần thiết của các điện trở kéo này là vì chân giao tiếp I2C của các thiết bị ngoại vi thường là dạng cực máng để hở. Giá trị của các điện trở này khác nhau tùy vào từng thiết bị và chuẩn giao tiếp, thường dao động trong khoảng $1\text{ k}\Omega$ đến $4.7\text{ k}\Omega$.

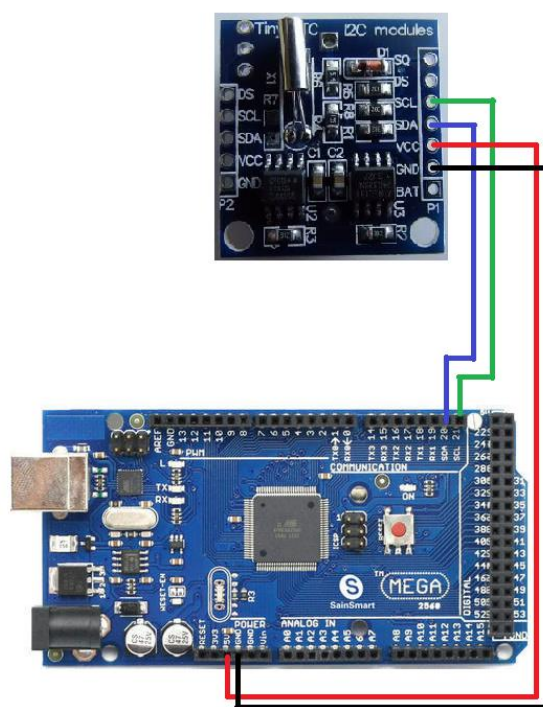
2.5.2 Truyền nhận dữ liệu Master – Slave

Một bus I2C có thể hoạt động ở nhiều chế độ khác nhau:

- Một chủ một tớ.
- Một chủ nhiều tớ.
- Nhiều chủ nhiều tớ.



Hình 2. 17 Sơ đồ kết nối board Mega 2560 với LCD có module I2C.



Hình 2. 19 Sơ đồ kết nối board Mega 2560 với module DS1307.

Đây là các Timekeeper registers của DS1307, chúng ta sẽ dựa vào bảng này để read/write IC DS1307 qua I2C.

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

Việc lấy dữ liệu từ các cảm biến trên module khác cũng tương tự. Như vậy các bạn đã biết cách sử dụng giao tiếp I2C.

2.7 Module cảm biến ánh sáng BH1750.

Cảm Biến Cường Độ Ánh Sáng BH1750 là cảm biến ánh sáng với bộ chuyển đổi ADC 16 bit tích hợp trong chip và có thể xuất ra trực tiếp dữ liệu theo dạng digital. cảm biến không cần bộ tính toán cường độ ánh sáng khác.

BH1750 sử dụng đơn giản và chính xác hơn nhiều lần so với dùng cảm biến quang trở để đo cường độ ánh sáng với dữ liệu thay đổi trên điện áp dẫn đến việc sai số cao. Với cảm biến BH1750 cho dữ liệu đo ra trực tiếp với dạng đơn vị là LUX không cần phải tính toán chuyển đổi thông qua chuẩn truyền I2C.

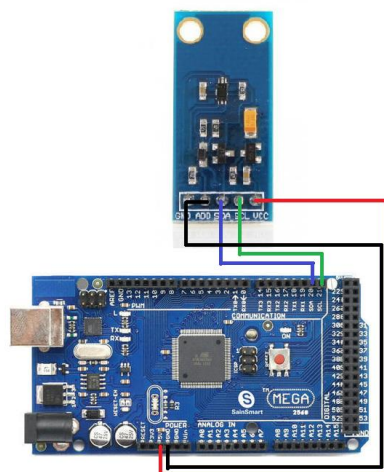


Cường độ được tính như sau:

- + Ban đêm: 0.001 - 0.02 lx.
- + Trời sáng trắng: 0.02 - 0.3 lx
- + Trời mây trong nhà: 5 - 50 lx.
- + Trời mây ngoài trời: 50 - 500 lx.
- + Trời nắng trong nhà: 100- 1000 lx.

Thông số kỹ thuật

- Chuẩn kết nối i2C
- Độ phân giải cao(1 - 65535 lx)
- Tiêu hao nguồn ít.
- Khả năng chống nhiễu sáng ở tần số 50 Hz/60 Hz
- Sự biến đổi ánh sáng nhỏ (+/- 20%)
- Độ ảnh hưởng bởi ánh sáng hồng ngoại rất nhỏ
- Nguồn cung cấp : 3.3V-5V
- Kích thước board : 0.85*0.63*0.13"(21*16*3.3mm)



Hình 2. 20 Sơ đồ kết nối board Mega 2560 với module BH1750.

CHƯƠNG 3: THIẾT KẾ VÀ THI CÔNG MÔ HÌNH THỰC NGHIỆM

3.1 LỰA CHỌN PHƯƠNG ÁN THIẾT KẾ

3.1.1 Các phương án thiết kế

3.1.1.1 Phương án 1: Sử dụng IC số

*** Ưu điểm:**

- Giá thành ngày càng rẻ.
- Kích thước nhỏ.
- Đa dạng về chủng loại
- Độ khả tín cao (tất cả các thành phần được chế tạo cùng lúc và không có những điểm hàn, nối).

- Tăng chất lượng.

- Các linh kiện được phối hợp tốt.

- Tuổi thọ cao.

*** Nhược điểm:**

- Do sử dụng năng lượng nhỏ nên hạn chế tốc độ làm việc.

- Yêu cầu về độ ổn định nguồn cung cấp cao.

- Mạch phức tạp so với các phương án khác.

3.1.1.2 Phương án 2: Sử dụng Vi điều khiển hoặc PLC

*** Ưu điểm:**

- Kích thước nhỏ gọn.
- Đa dạng về chủng loại.
- Khả năng tùy biến chức năng, cũng như xử lý công việc cao.
- Nhiều tài liệu và phần mềm hỗ trợ lập trình viên.
- Đơn giản khi thiết kế mạch.
- Tuổi thọ cao.

*** Nhược điểm:**

- Giá thành vẫn còn khá cao.
- Yêu cầu khả năng lập trình.

3.1.2 Phương án thiết kế chính

Sau khi xem xét các phương án thiết kế hiện có, chúng em quyết định thực hiện phương án 2: Sử dụng Vi điều khiển hoặc PLC. Mà cụ thể với đề tài này, chúng em sẽ sử dụng bo Arduino. Ngoài các ưu, nhược điểm đã được kể trên, với Arduino sẽ có một số ưu, nhược điểm nổi bật khác như sau:

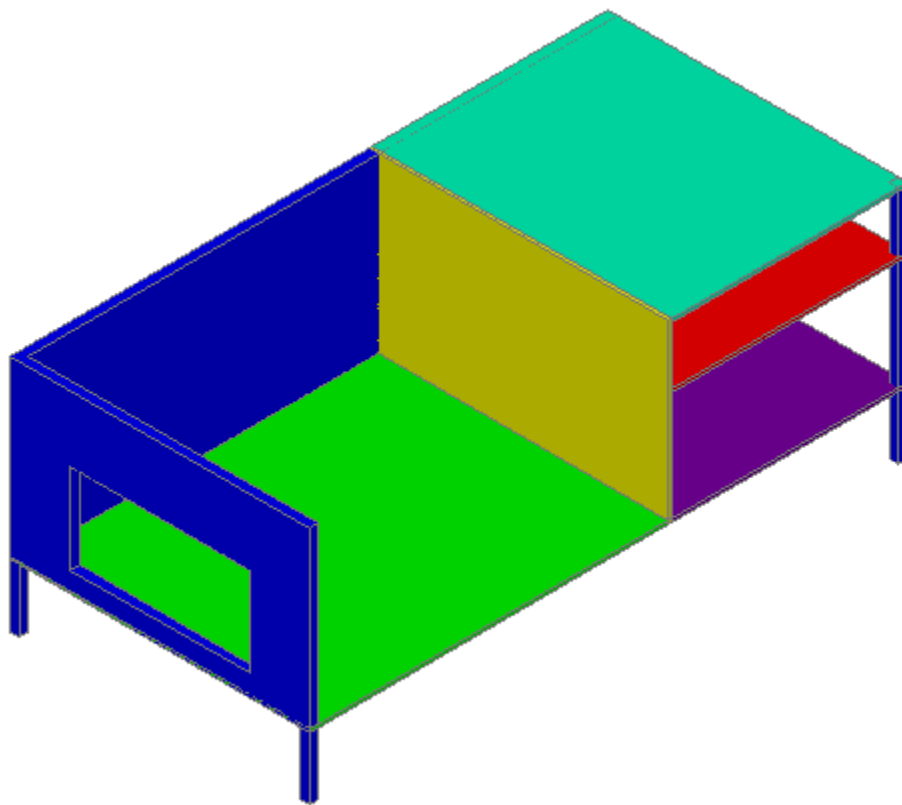
• Ưu điểm:






- Nhiều loại Arduino và các “phụ kiện” cho Arduino.
- Đã được tiêu chuẩn hóa, dễ dàng lắp ráp và lập trình.
- Mã nguồn mở, cộng đồng sử dụng đông đảo.
- Chi phí đầu tư thấp.

• Nhược điểm:

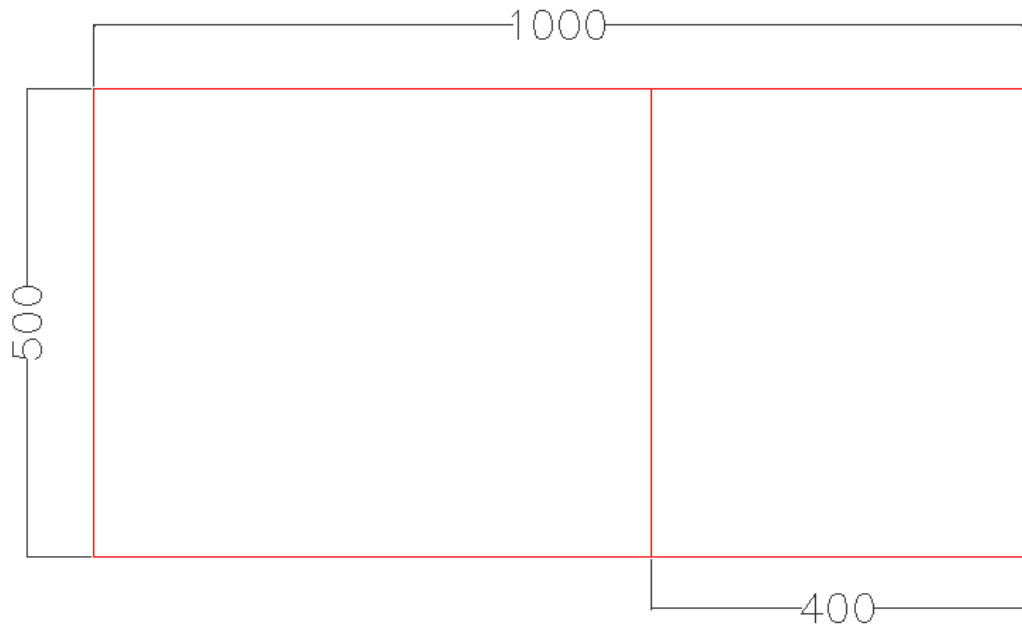
- Khả năng chống chịu với các tác nhân vô sinh của môi trường chưa cao.
- Đang dần bị yếu đi so với các tân binh như Raspberry Pi, ...

3.2 THIẾT KẾ MÔ HÌNH:

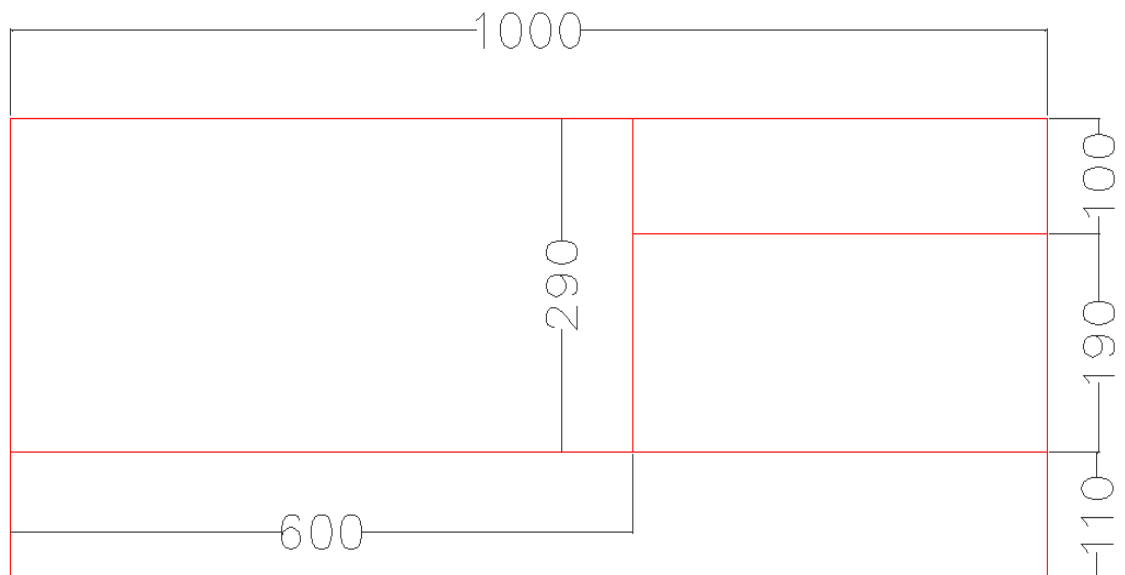


- | | |
|---|----------------------------------|
|  | Khu vực trồng rau |
|  | Bàn phím và LCD |
|  | Khối nguồn và mạch vi điều khiển |
|  | Máy bơm và thiết bị làm lạnh |
|  | Các cảm biến |

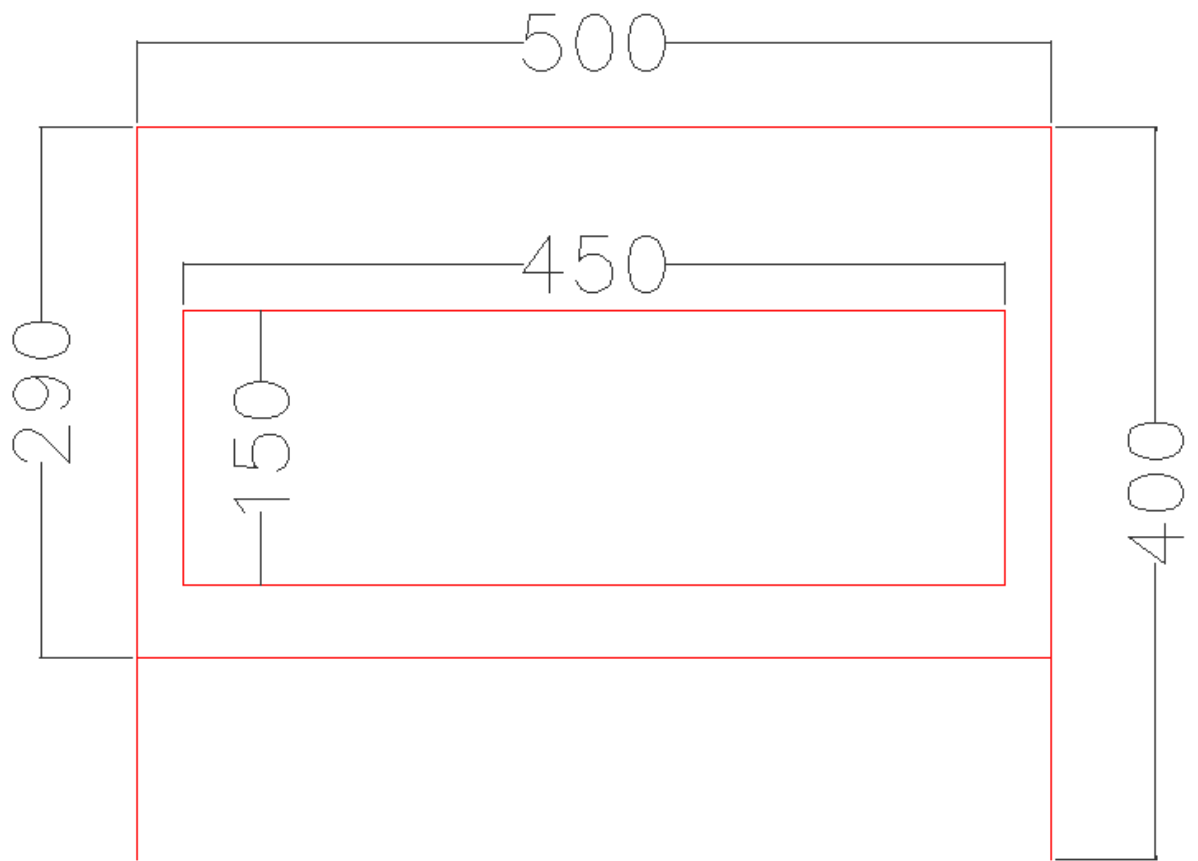
Hình 3. 1 Mô hình thiết kế 3D



Hình 3. 2 Hình chiếu bằng



Hình 3. 3 Hình chiếu đứng.



Hình 3. 4 Hình chiếu cạnh.

3.3 THIẾT KẾ SƠ ĐỒ KHỐI



Hình 3. 5 Sơ đồ khối hệ thống.

- Khối cảm biến: có chức năng chuyển đổi các đại lượng vật lý thành tín hiệu số, cung cấp dữ liệu đầu vào cho khối xử lý.
- Khối bàn phím và Bluetooth: cung cấp cách thức cho người dùng thao tác với hệ thống.
- Khối nguồn: cung cấp năng lượng cho toàn mạch.
- Khối hiển LCD 20x4: Hiển thị các thông tin mà người dùng cần qua tâm.
- Khối thực thi: điều khiển các thiết bị ngoại vi gồm máy bơm, máy lạnh, quạt thông gió, đèn chiếu sáng.

3.4 TÍNH TOÁN THIẾT KẾ

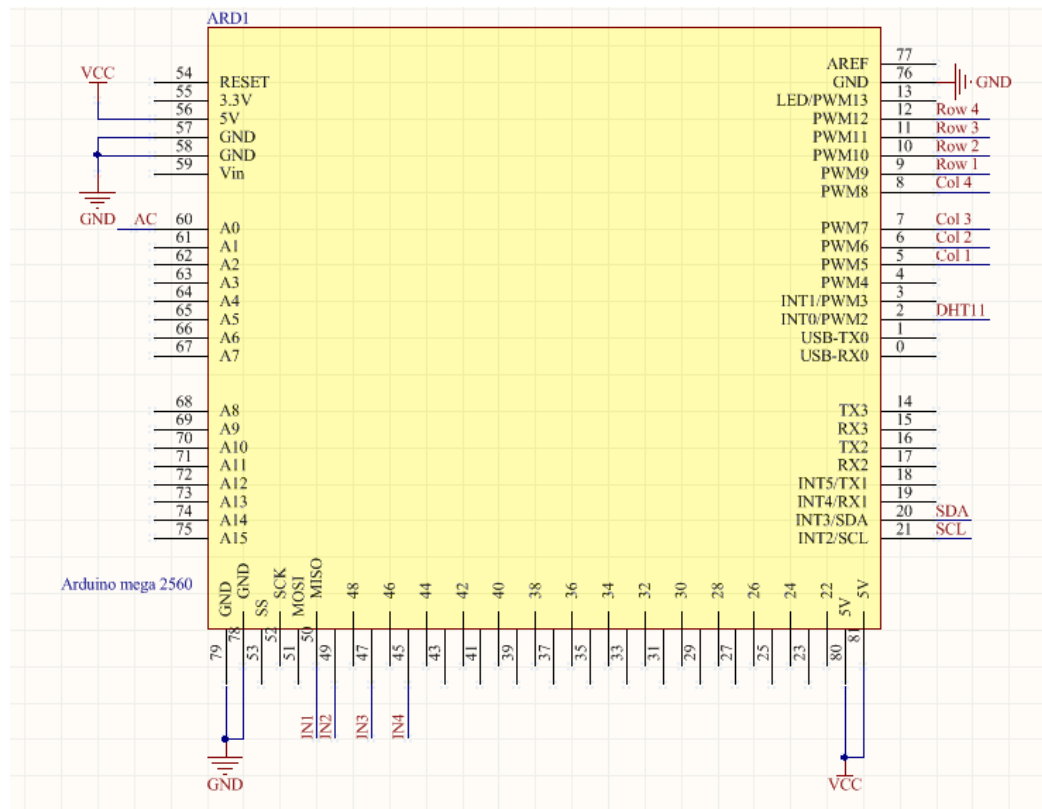
3.4.1 Thiết kế từng khối cụ thể

- Khối Nguồn: Nguồn 12 Vdc – 30A. Và 220 Vac.



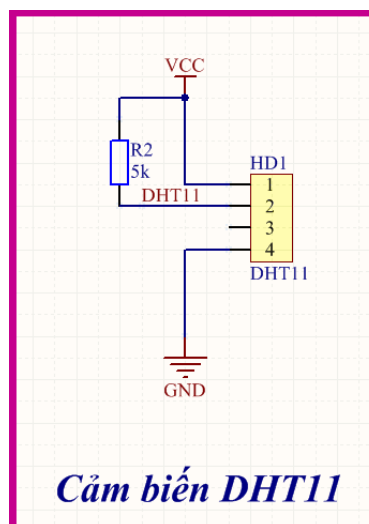
Hình 3. 6 Ổ cắm 220Vac và nguồn 12Vdc - 30A.

- Khối xử lý: mạch sử dụng Arduino (đã được trình bày ở chương 2)



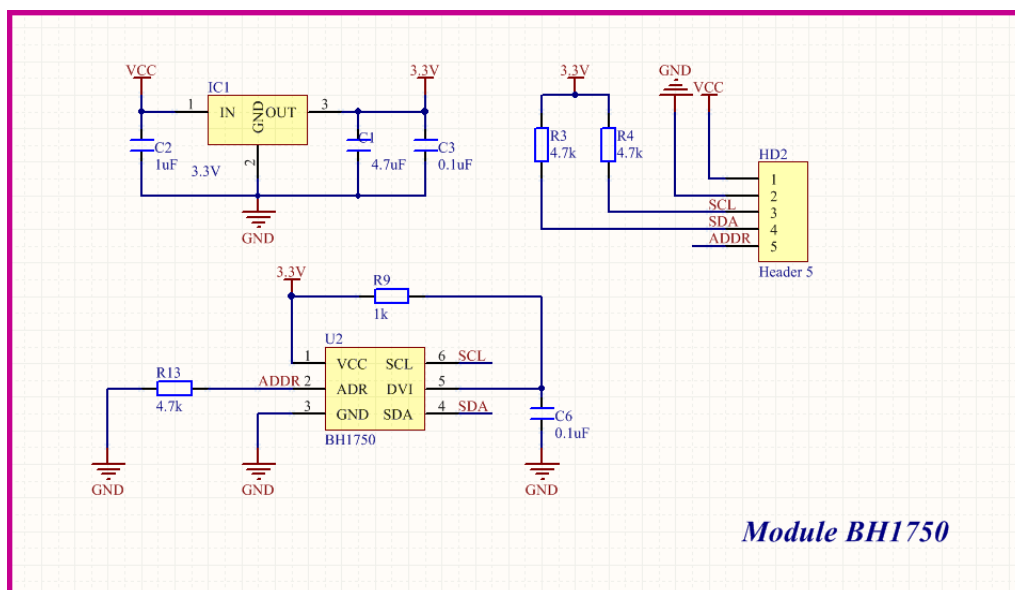
Hình 3. 7 sơ đồ kết nối với các khối khác của vi điều khiển.

- Khối cảm biến: mạch sử dụng cảm biến DHT11, độ ẩm đất, module BH1750, module DS1307. (đã được trình bày ở chương 2)



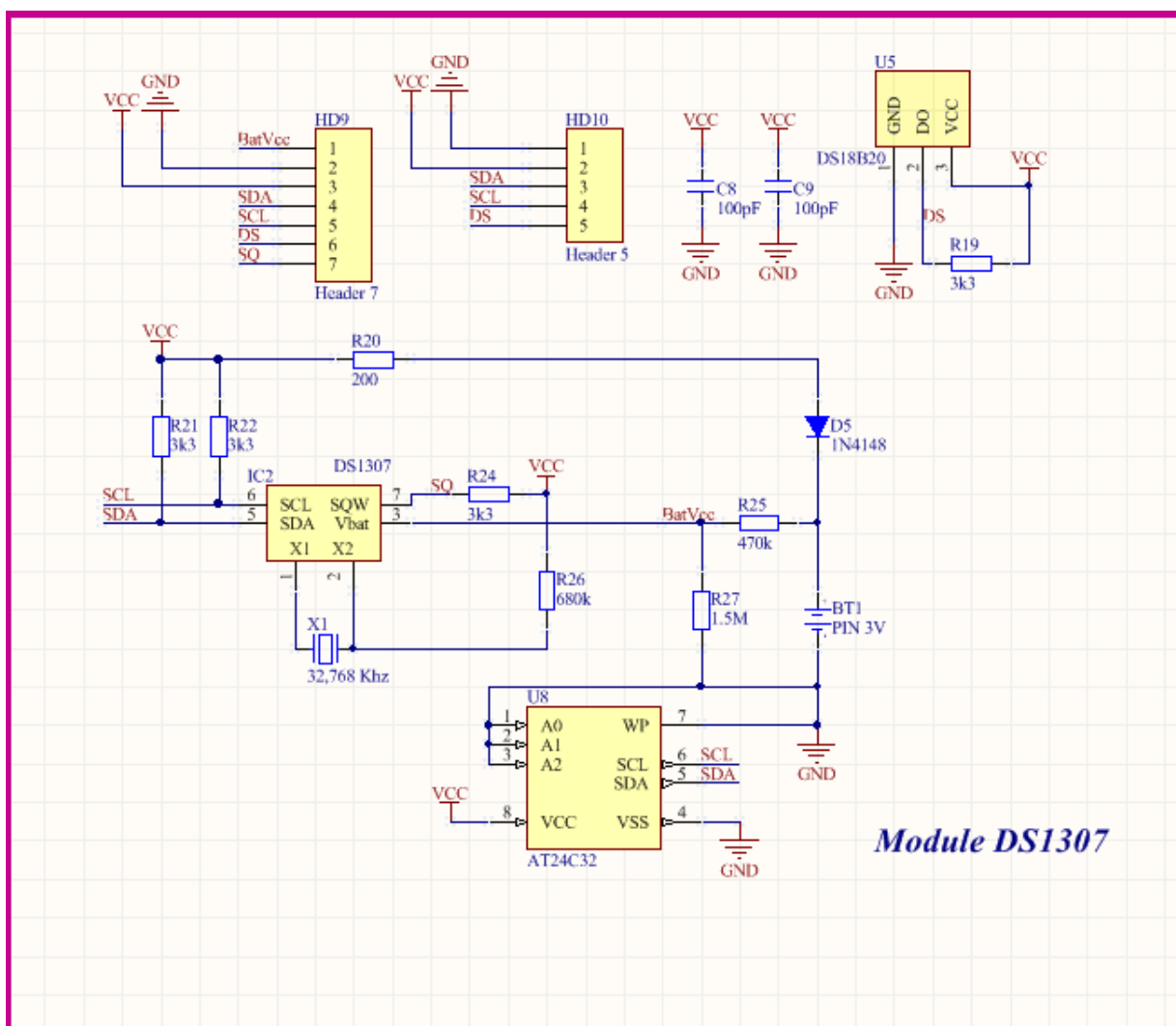
Hình 3. 8 Sơ đồ nguyên lý của cảm biến DHT11.

- Như chương 2 ta đã biết vì cảm biến DHT11 đưa về giá trị điện áp với dải tần số tương ứng với 5 byte. Vì thế ta kết nối chân data của cảm biến với chân 2 của Arduino.



Hình 3. 9 Sơ đồ nguyên lý của Module BH1750.

- Vì cảm biến hoạt động tốt ở điện áp 3.3Vdc nên ta cần hạ điện áp của nguồn từ 5Vdc bằng IC1 còn các tụ C1, C2, C3 có tác dụng lọc nhiễu và ổn định điện áp cho khối. Các điện trở như R3, R4 thì có tác dụng là cho tín hiệu từ cảm biến ổn định hơn. Với R9, R13, C6 là do yêu cầu từ datasheet của cảm biến. Cảm biến giao tiếp bằng I2C với chân 6, 4 của cảm biến kết nối theo thứ tự với chân 21, 20 của vi điều khiển.

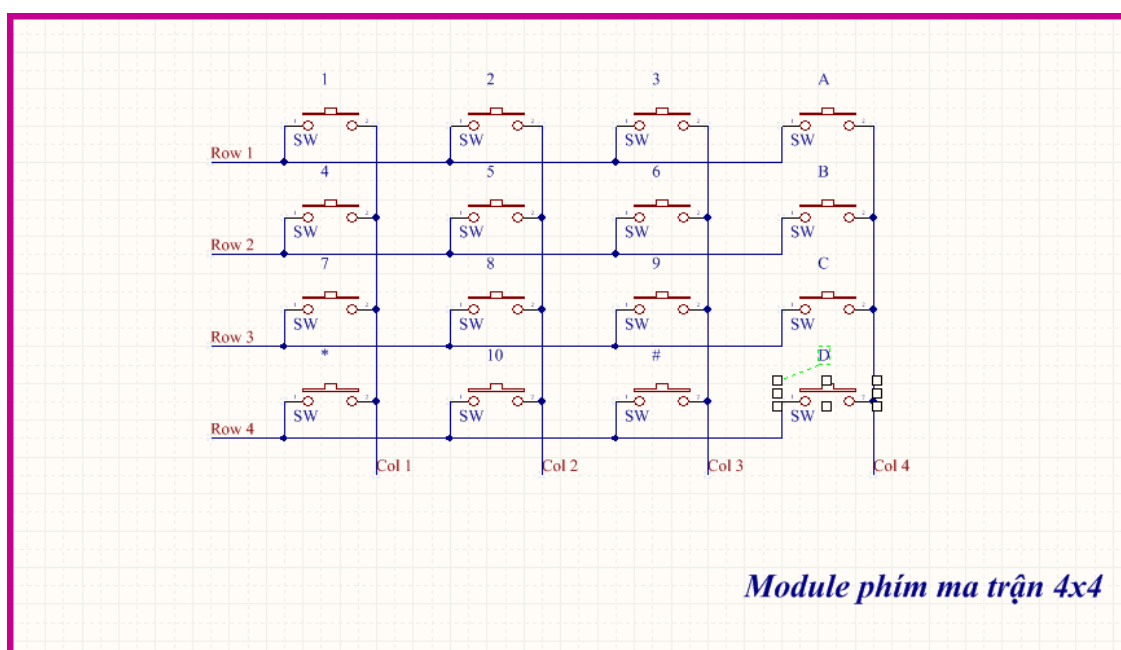


Hình 3. 10 Sơ đồ nguyên lý DS1307.

- Các linh kiện của khối đều dựa vào datasheet của DS1307. Ở đây AT24C32 có nhiệm vụ giao tiếp với board Arduino và lấy dữ liệu của DS1307. Thạch anh X1 đảm nhiệm để DS1307 đếm thời gian. Và pin BT1 có nhiệm vụ cung cấp nguồn cho DS1307 hoạt động liên tục.
- Khối này cũng dùng giao tiếp I2C và kết nối với chân 20, 21 của Arduino.
- Đây là các Timekeeper registers của DS1307, chúng ta sẽ dựa vào bảng này để read/write IC DS1307 qua I2C.

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

• Khối phím nhấn:

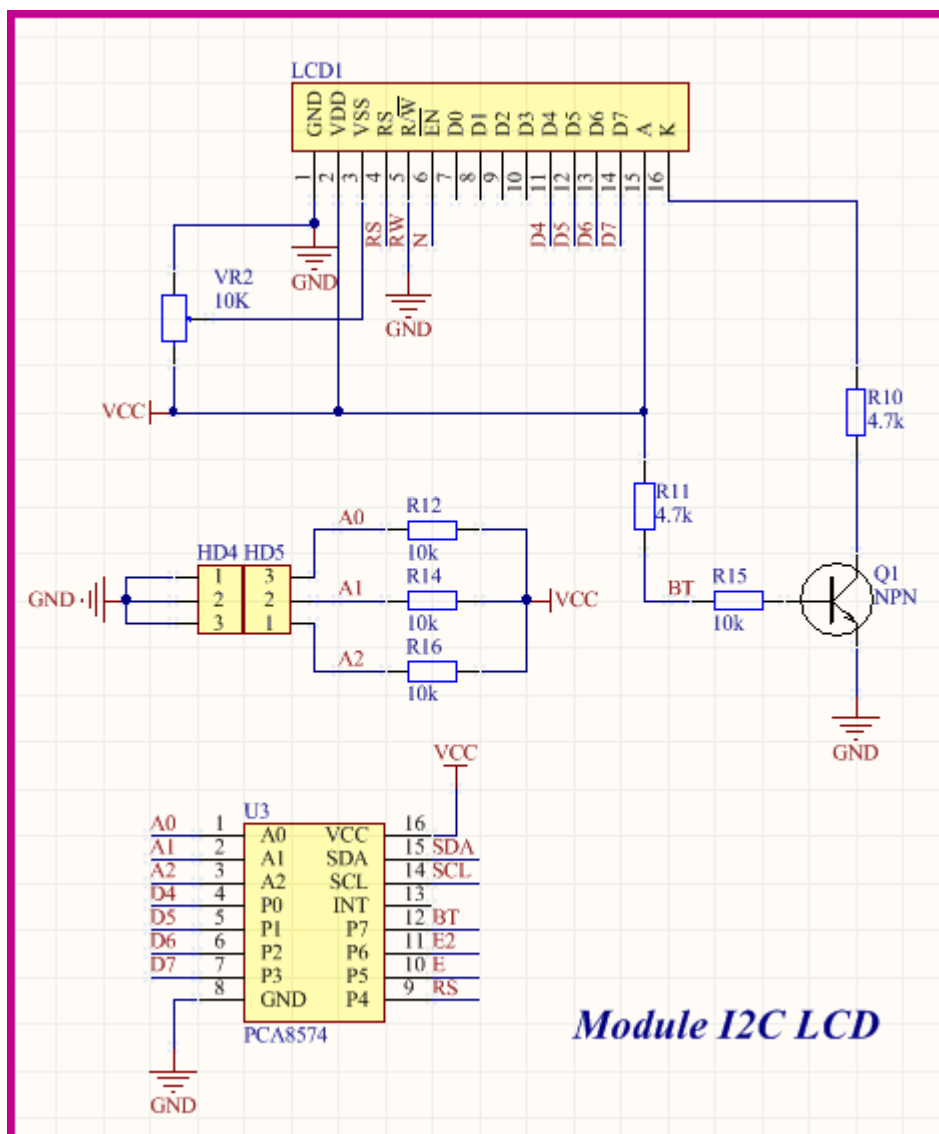


Hình 3. 11 Sơ đồ nguyên lý khối phím nhấn.

- Bình thường chân của vi điều khiển sẽ ở mức cao do được nối lên nguồn thông qua điện trở. Tuy nhiên khi nhấn nút chân vi điều khiển sẽ xuống mức 0. Căn cứ vào đây, ta có thể lập trình cho MCU thực hiện một công việc nào đó khi chân của MCU nối với khối nút nhấn ở mức 0.

- Từ Row 1 - Row 4 kết nối theo thứ tự từ chân 9 – 12 của Arduino và Col 1 – Col 2 kết nối theo thứ tự chân 5 – 8 của Arduino.

- Khối hiển thị: mạch dùng LCD 20x04 (đã được trình bày ở chương 2) để hiển thị các thông tin.



Hình 3. 12 Sơ đồ nguyên lý khối LCD.

- Vì LCD chiếm khá nhiều chân của vi điều khiển em sử dụng module giao tiếp I2C để tiết kiệm chân cho Board Arduino. PCA8574 có nhiệm vụ giao tiếp với Arduino Mega 2560 và điều khiển LCD.
- Khối này của dung giao tiếp I2C và kết nối với chân 20, 21 của vi điều khiển.

• Khối thực thi: Gồm nhiều các thiết bị ngoại vi khác nhau được điều khiển bởi Arduino như các module điều khiển nhiệt độ, độ ẩm không khí, độ ẩm đất, ...

- Tính transistor

Ta có: $R_{\text{cuộn dây}} = 400\Omega$

$$P_{\text{cuộn dây}} < 0.36 \text{ W}$$

$$\Rightarrow I_{\text{cuộn dây}} < \sqrt{\frac{0.36}{400}} = 0.03\text{A}$$

Ta có: $I_C \cdot R_C + V_{CE} = V_{CC}$

Để transistor dẫn bão hòa thì $V_{CE} = 0$

$$\Rightarrow I_C = \frac{V_{CC}}{R_C} = \frac{5}{400} = 0.0125\text{A} = 12.5\text{mA}$$

$$\Rightarrow I_B = \frac{I_C}{\beta} = \frac{12.5}{70} \approx 0.2\text{mA}$$

$$\Rightarrow V_B = 5\text{V} - V_{BE} = 5 - 0.7 = 4.3 \text{ V}$$

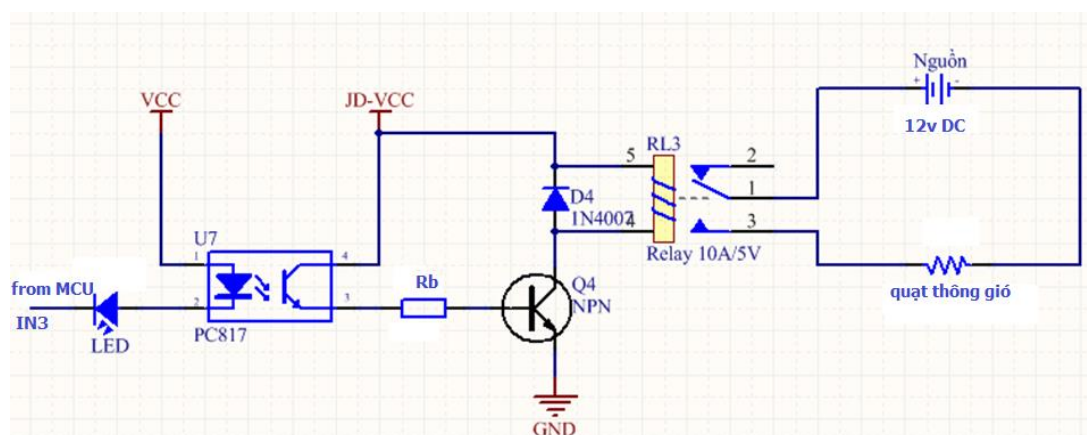
$$\Rightarrow R_B = \frac{V_B}{I_B} = \frac{4.3}{0.2 \times 10^{-2}} = 2150\Omega$$

Vậy ta chọn mua transistor có dòng lớn hơn 12.5mA (chọn C1815)

- Quạt thông gió: Dùng nguồn 12Vdc, 0.3W.



Hình 3. 13 Hình ảnh quạt thông gió trong mô hình.

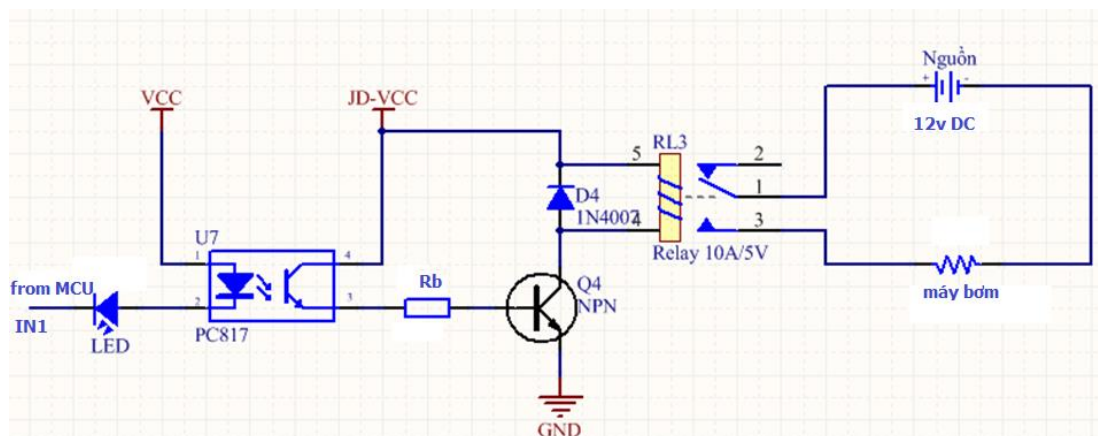


Hình 3. 14 Sơ đồ kết nối quạt.

- Máy bơm : Nguồn 12Vdc 60W.

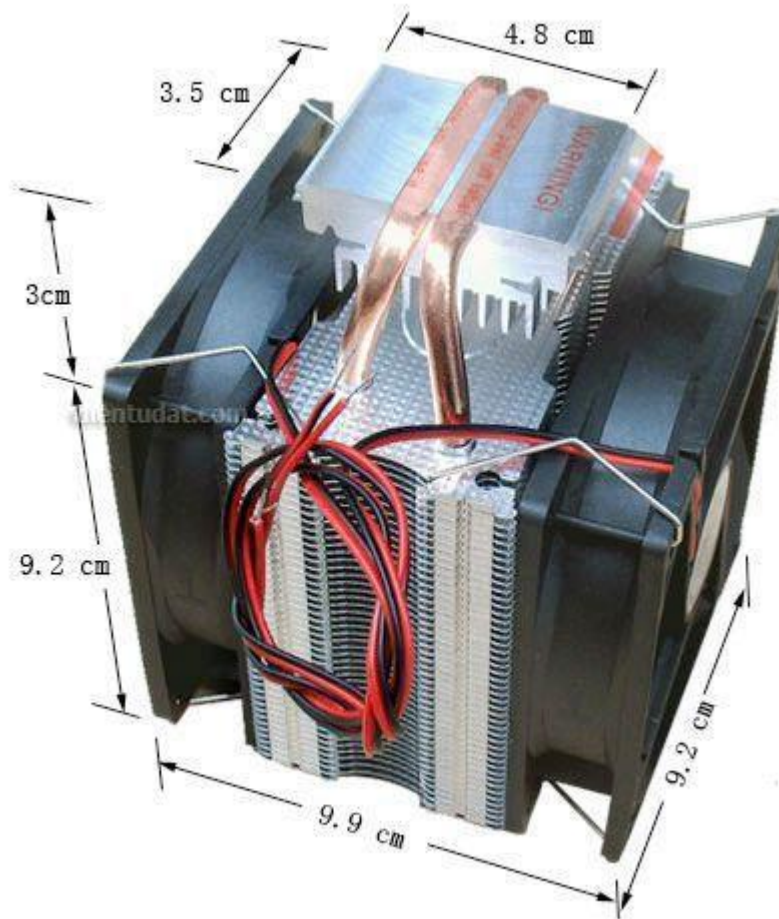


Hình 3. 15 Máy bơm mini

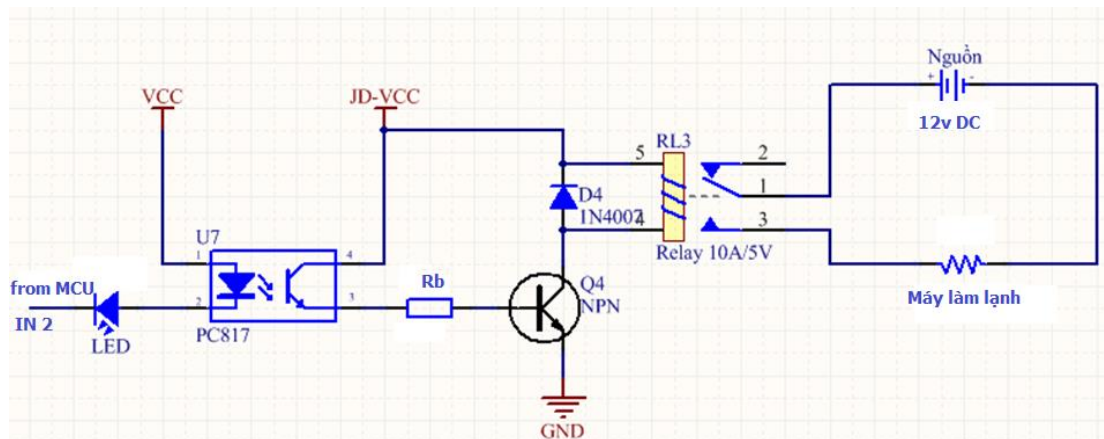


Hình 3. 16 Sơ đồ kết nối máy bơm.

- Máy làm mát: 12Vdc, 44.5W.



Hình 3. 17 Sò nóng lạnh và quạt tản nhiệt.

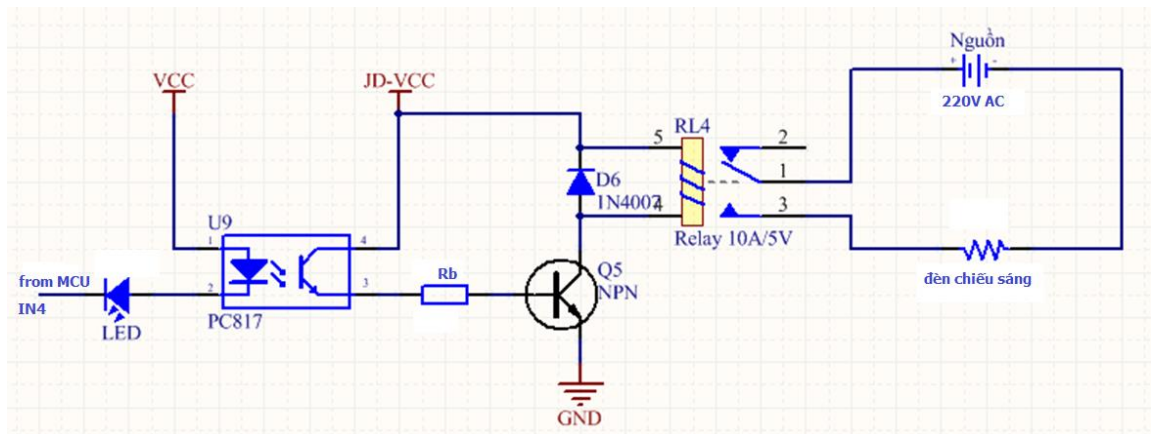


Hình 3. 18 Sơ đồ kết nối máy làm lạnh.

- Đèn chiếu sáng: dụng điện 220Vac.

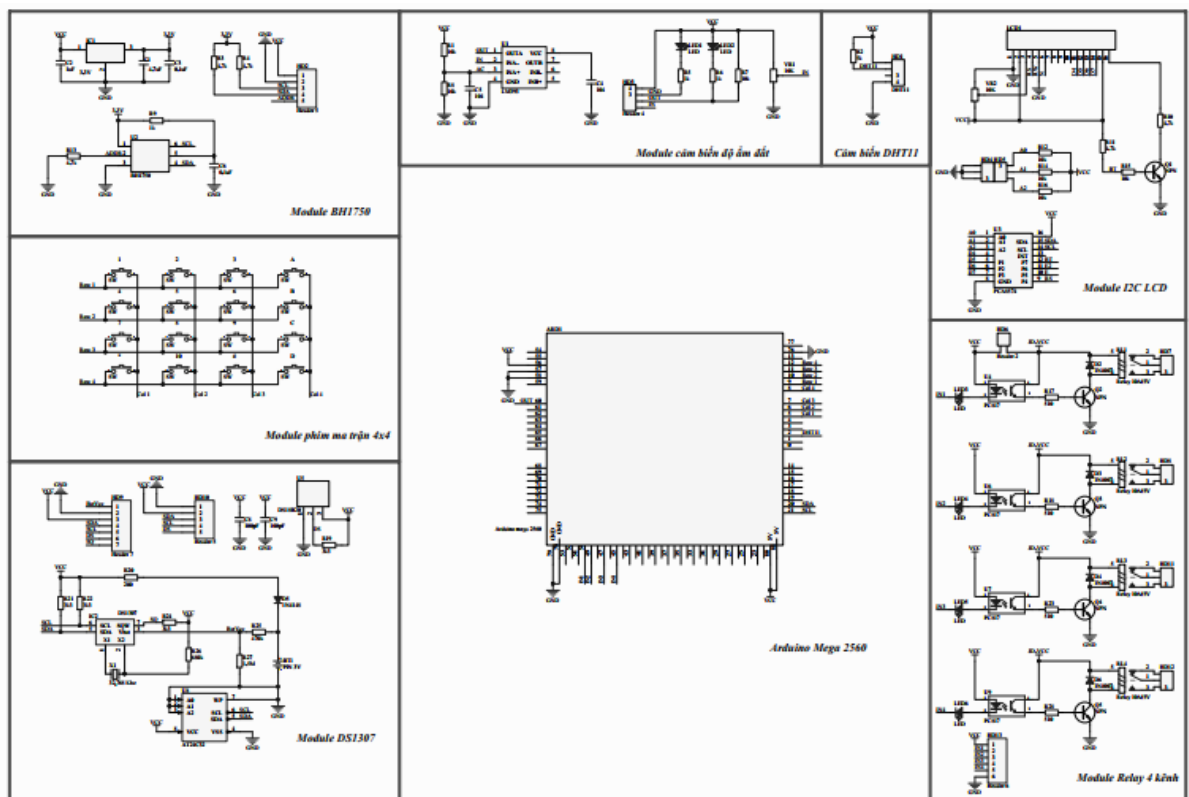


Hình 3. 19 Đèn chiếu sáng.



Hình 3. 20 Sơ đồ kết nối đèn.

3.4.2 Sơ đồ nguyên lý toàn hệ thống:



Hình 3. 21 Sơ đồ nguyên lý hệ thống.

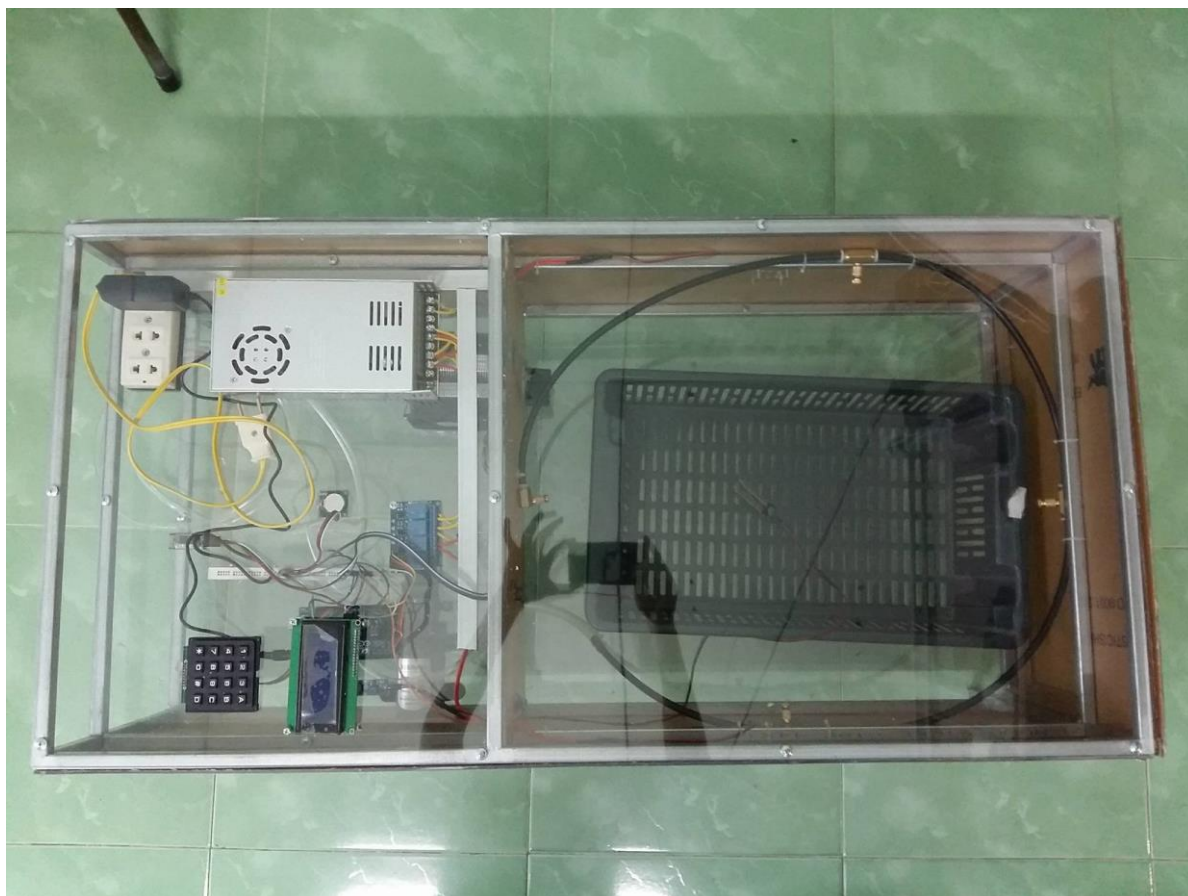
3.4.2 Tổng quan về các chức năng của hệ thống

Mô hình nhà vườn được thiết kế để tạo ra một môi trường sống cho cây trồng sinh trưởng và phát triển mà ở đó chúng ta có thể kiểm soát được các yếu tố vô sinh như ánh sáng, nhiệt độ, độ ẩm, ... Giúp cây trồng có thuận lợi phát triển, hạn chế những chi phối từ điều kiện tự nhiên bên ngoài, giảm trừ sâu bệnh. Từ đó giúp tăng năng suất cây trồng.

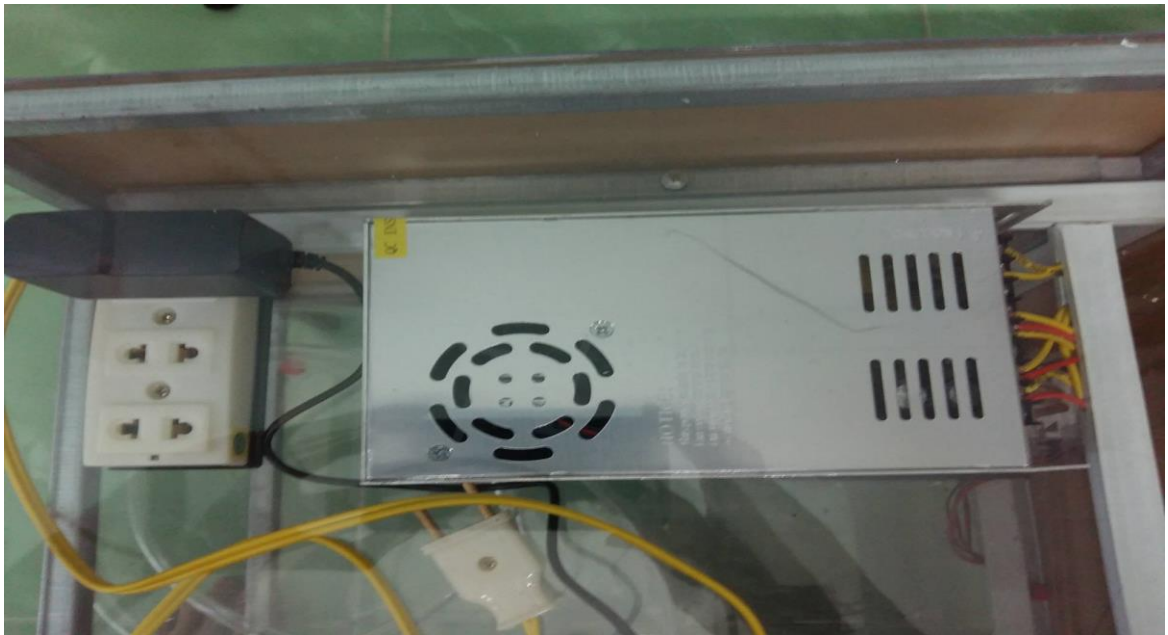
Với hai chế độ là Auto và Manual. Người điều khiển có thể cài đặt và chạy hệ thống với tùy biến sao cho phù hợp với thực tế sử dụng.

Khi sự cố đã được khắc phục xong, chúng ta có thể khởi động lại hệ thống thông qua một nút nhấn vật lý hoặc có thể khởi động lại hệ thống ngay trên thiết bị thông minh có kết nối bluetooth.

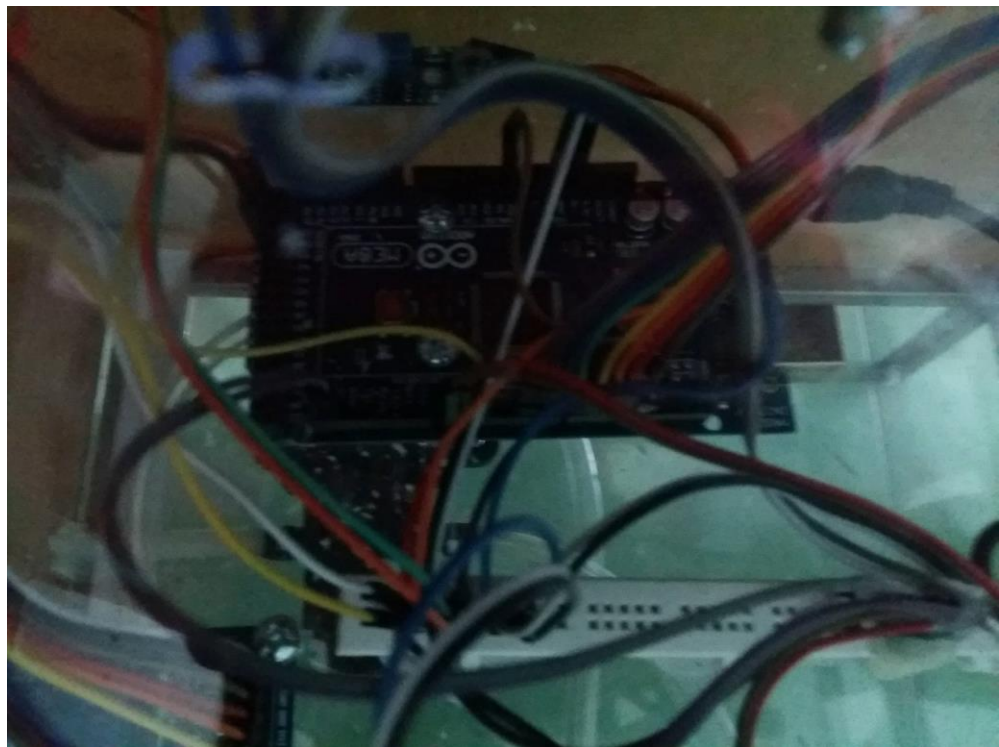
3.4.3 Mô hình thực tế



Hình 3. 22 Mô hình sau khi thi công.



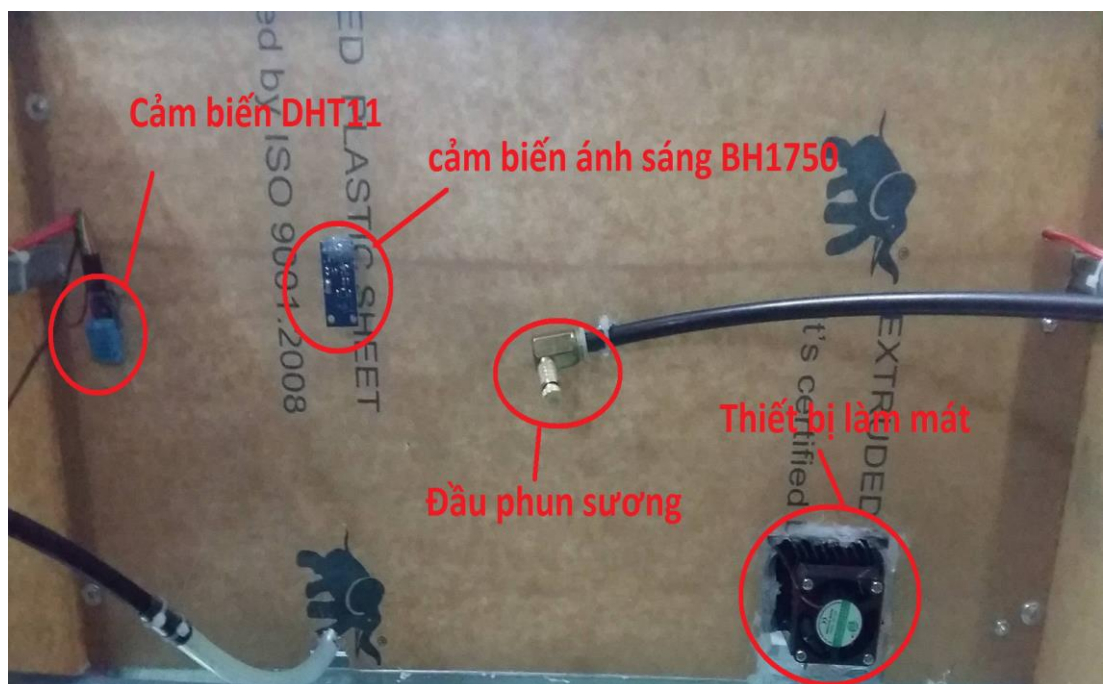
Hình 3. 23 Khối nguồn 12V DC-30A.



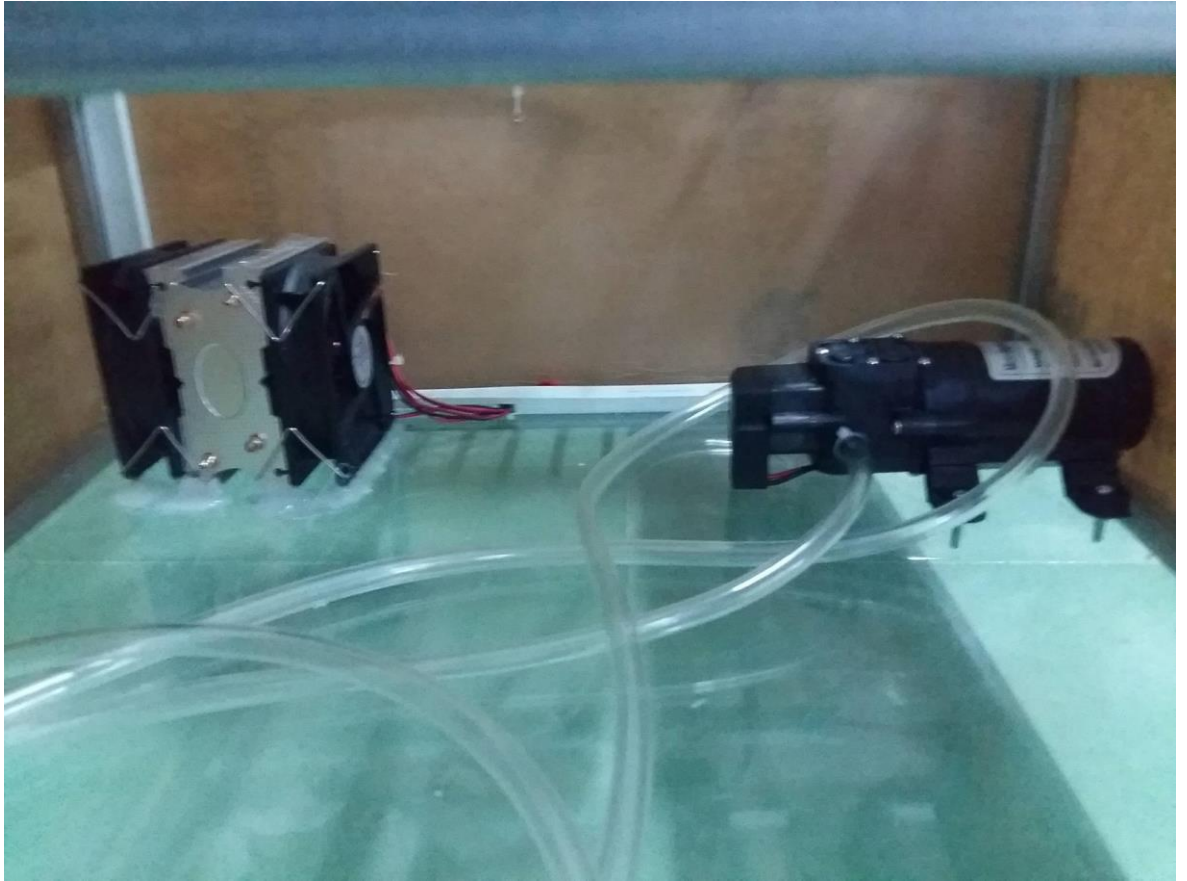
Hình 3. 24 Mạch vi điều khiển Arduino Mega 2560.



Hình 3. 25 Phím ma trận 4x4 điều khiển và LCD hiển thị 20x4.



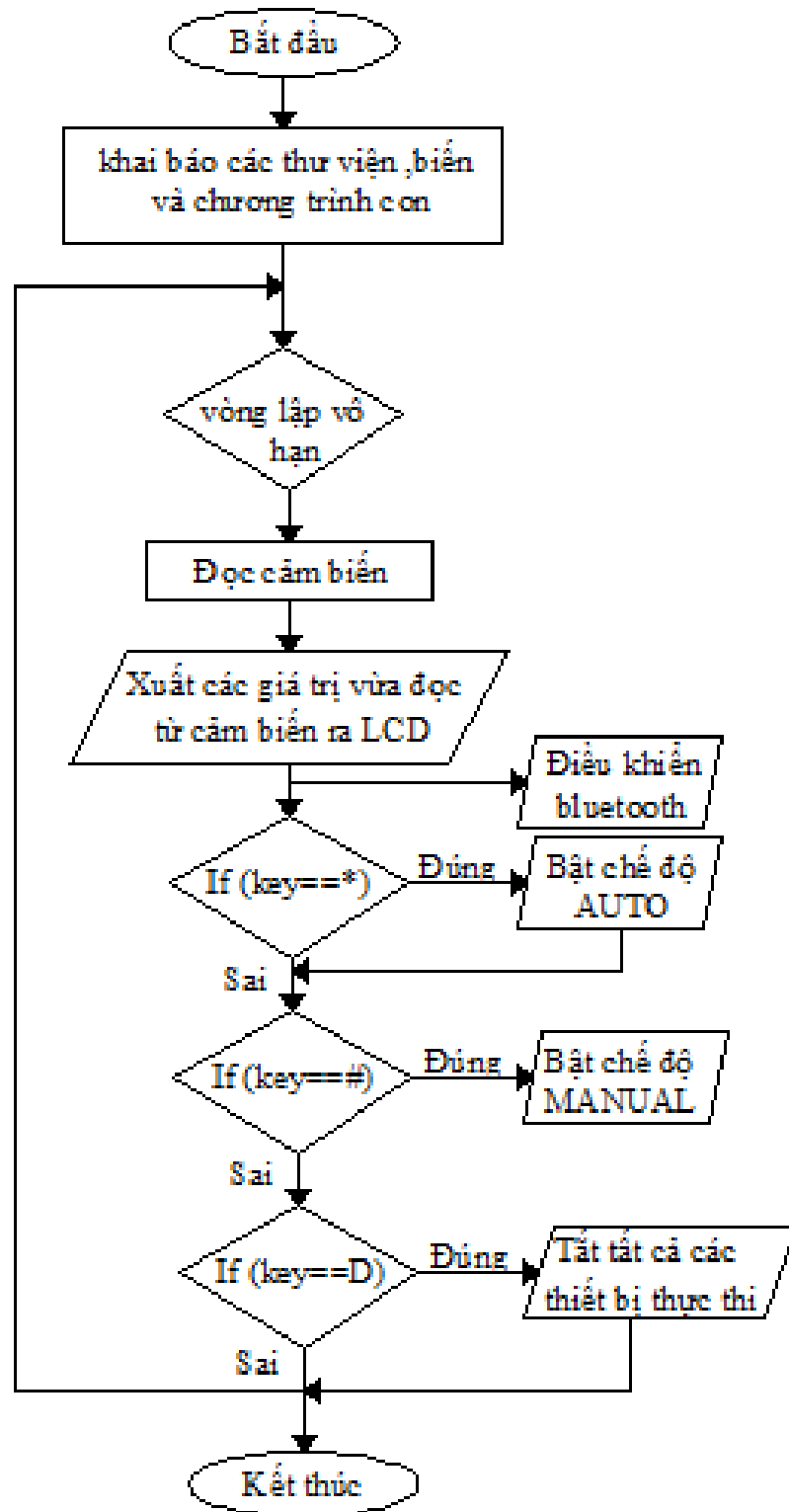
Hình 3. 26 Các cảm biến và bộ phận thực thi của hệ thống.



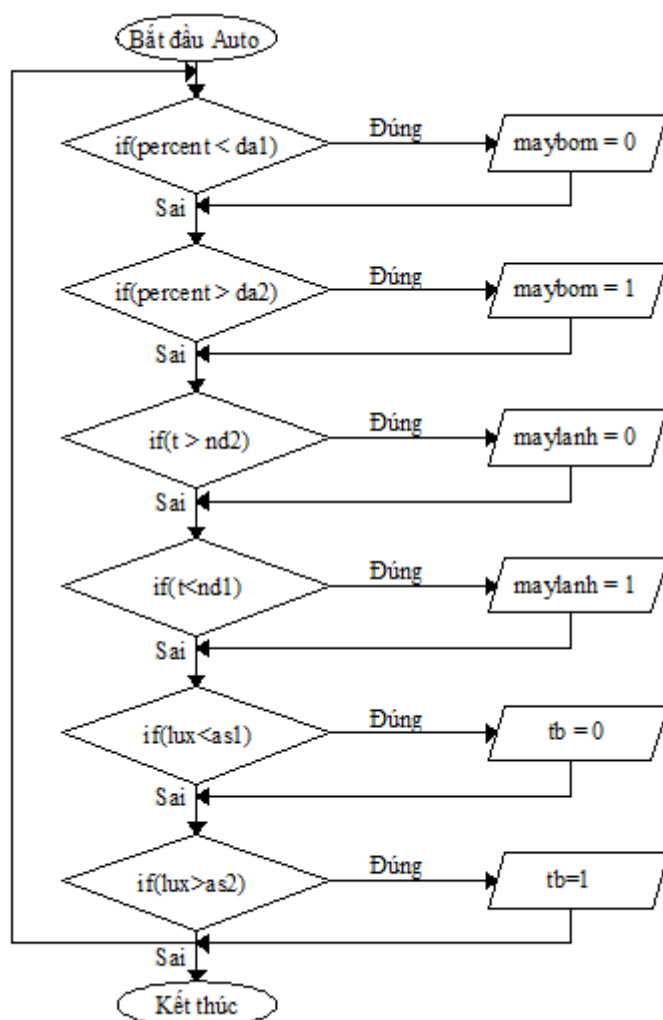
Hình 3. 27 Thiết bị làm lạnh và máy bơm.

3.5 LƯU ĐỒ GIẢI THUẬT

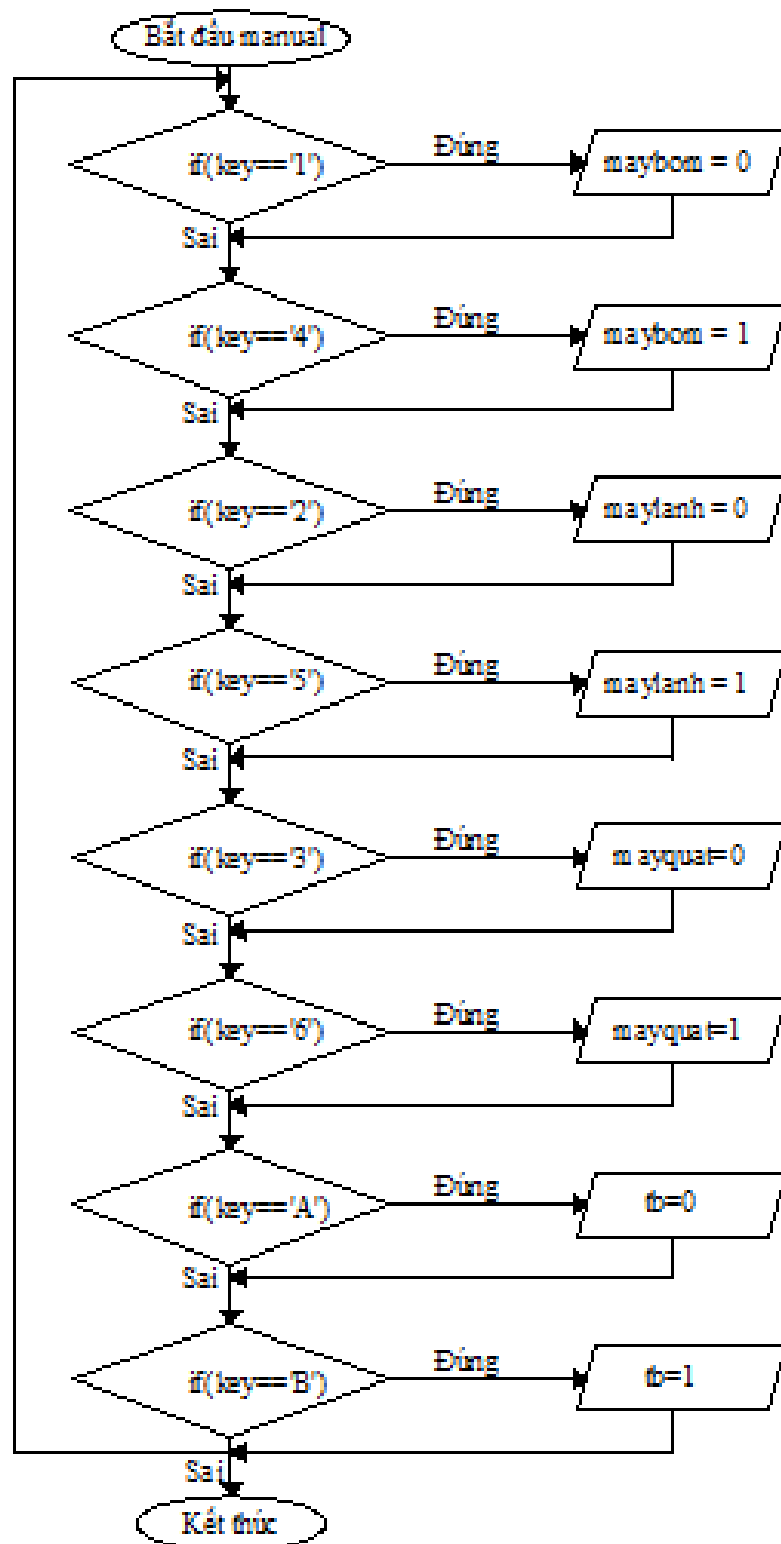
3.5.1 Chương trình chính



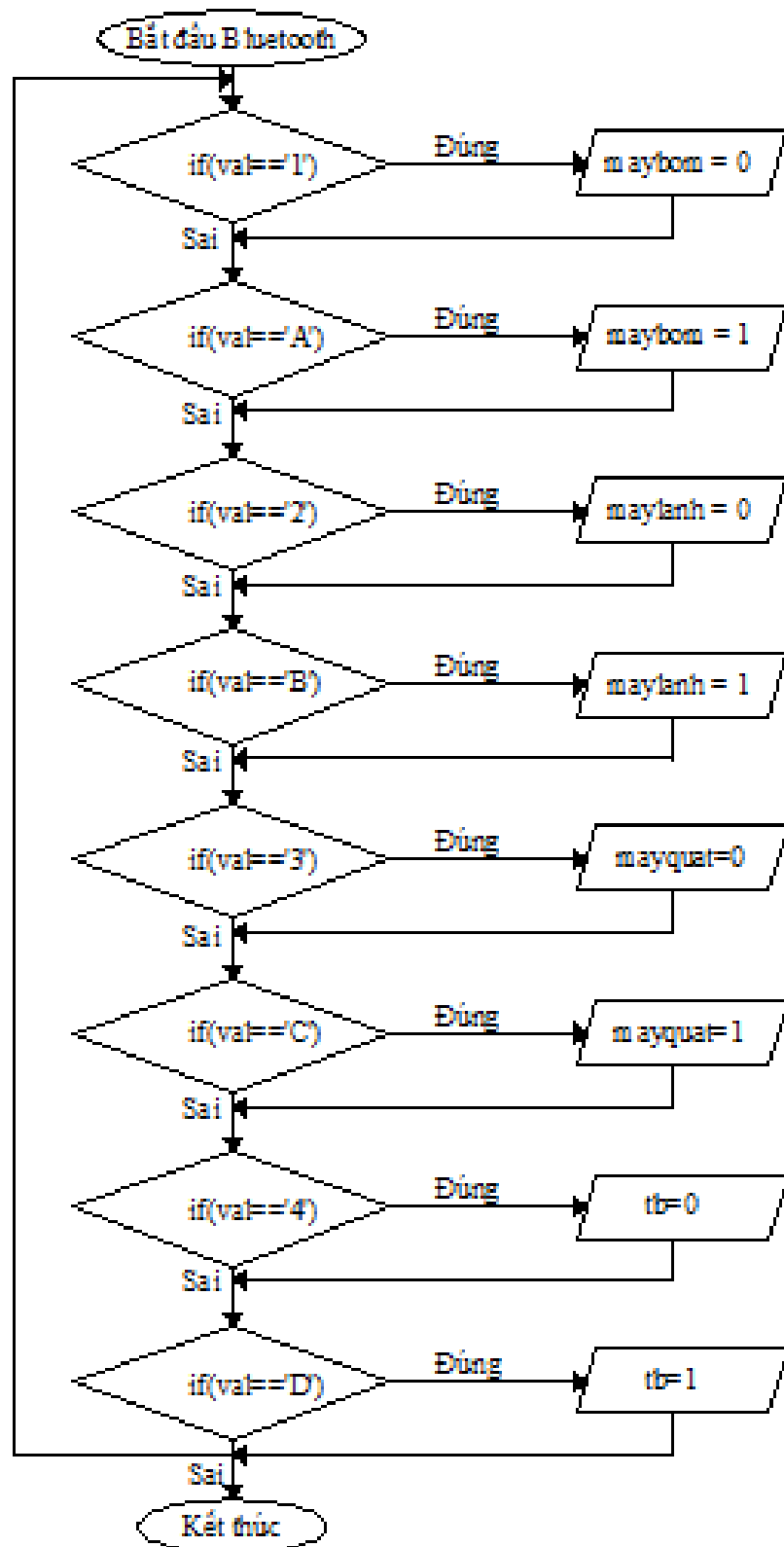
- Đầu tiên là khai báo thư viện, biến và chương trình con.
- Sau đó là đọc giá trị từ cảm biến.
- Xuất ra màn hình LCD.
- Và giá trị nhận từ phím ma trận.
- Nếu key của phím trả về giá trị '*' bật chế độ auto.
- Nếu key = # thì mở chế độ manual:
 - là bàn phím ma trận có biến là key để điều khiển relay khởi động các thiết bị thực thi.
 - Nếu key = D là dừng khẩn cấp tất cả các relay đều hờ các thiết bị ngưng hoạt động.
- Chế độ thông qua Bluetooth:
 - là bằng điện thoại thông qua Bluetooth với biến là val để điều khiển relay khởi động các thiết bị thực thi.



- Nếu độ ẩm đất môi trường nhỏ hơn độ ẩm điều kiện da1 thì máy bơm = 0, thì relay đóng máy bơm khởi động bơm nước.
- Nếu độ ẩm đất môi trường lớn hơn độ ẩm điều kiện da2 thì máy bơm = 1, thì relay hở máy bơm ngưng hoạt động.
- Nếu nhiệt độ môi trường lớn hơn nhiệt độ điều kiện nd2 thì máy lạnh = 0, thì relay đóng sò nóng lạnh và quạt tản nhiệt hoạt động.
- Nếu nhiệt độ môi trường nhỏ hơn nhiệt độ điều kiện nd1 thì máy lạnh = 1, thì relay hở sò nóng lạnh và quạt tản nhiệt ngưng hoạt động.
- Nếu ánh sáng môi trường nhỏ hơn ánh sáng điều kiện as1 thì tb = 0, thì relay đóng lại đèn chiếu sáng hoạt động.
- Nếu ánh sáng môi trường lớn hơn ánh sáng điều kiện as2 thì tb = 1, thì relay hở đèn chiếu sáng ngưng hoạt động.



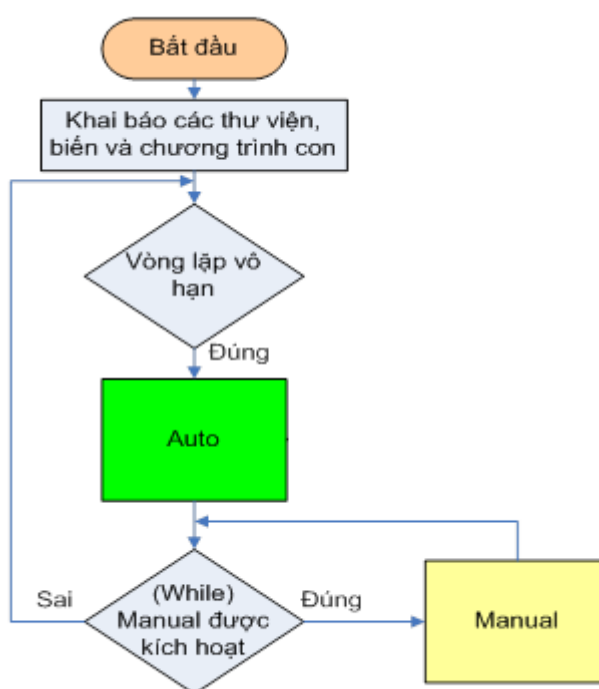
- Điều khiển bằng phím ma trận 4x4



- Điều khiển thiết bị thực thi bằng điện thoại thông qua Bluetooth.

3.6 MÔ TẢ HOẠT ĐỘNG CỦA CHƯƠNG TRÌNH

Như đã đề cập, chương chính chính sẽ có chức năng nhận các tín hiệu từ cảm biến, xử lý và điều khiển các cơ cấu chấp hành, hiển thị các thông số lên màn hình với các chế độ Auto và Manual. Tuy nhiên, do việc bị cắt ra thành nhiều trang khiến cho việc đọc lưu đồ gặp nhiều khó khăn. Với hình 3.28 sẽ cho chúng ta cái nhìn tổng quan về lưu đồ của chương trình. Từ đó, hiểu được nguyên lý hoạt động của hệ thống.



Hình 3. 28 Lưu đồ giải thuật tổng quan chương trình.

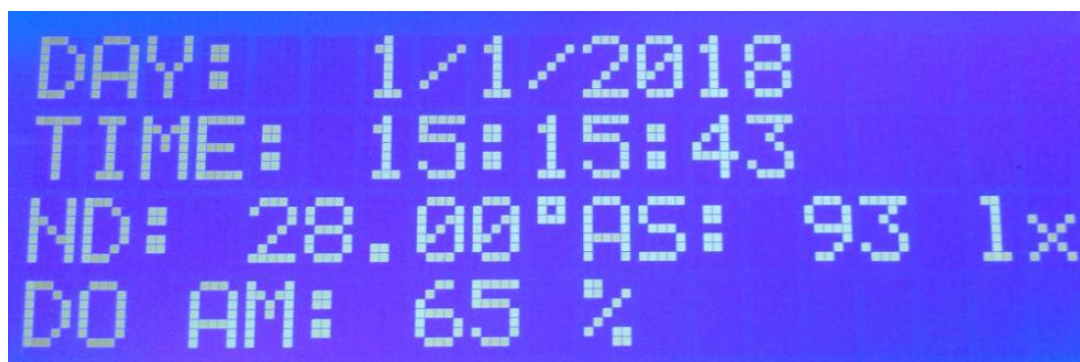
Khi được bắt đầu, chương trình sẽ chạy trong một vòng lặp vô hạn. Các code trong vòng lặp vô hạn này có chức năng giúp cho hệ thống chạy chế độ Auto.

3.7 CODE CHƯƠNG TRÌNH

(Xem phần Phụ lục.)

3.8 KẾT QUẢ THỰC NGHIỆM

- Kết quả thực nghiệm được thu thập để đánh giá trong 120 phút chạy thử mô hình.



Hình 3. 29 Kết quả thực nghiệm.

- Khi sử dụng nguồn từ adapter, module điều khiển trung tâm sinh nhiệt khá nhiều.
- Ở chế độ Auto, các nút nhấn điều khiển máy bơm, máy điều hòa, máy tạo độ ẩm bị vô hiệu hóa.
- Các cảm biến không được thiết kế để hoạt động trong môi trường có tính ăn mòn cao.
- Kết quả thực nghiệm trên mô hình được trình bày trong bảng 3.1 và bảng 3.2:

Bảng 3.1 Yêu cầu thiết kế mô hình nhà trồng rau mầm.

Các thông số cảm biến			Ghi chú
Nhiệt độ	Độ ẩm đất	Ánh sáng	
26° C ÷ 32° C	50% ÷ 60%	70 lux – 90lux	

Bảng 3.2 Kết quả thực nghiệm mô hình nhà trồng rau mầm.

Các thông số cảm biến			Ghi chú
Nhiệt độ	Độ ẩm đất	Ánh sáng	
$28^{\circ}C \div 32^{\circ}C$	$50\% \div 65\%$	70 lux – 93lux	.

Kết luận: Đạt yêu cầu theo mong muốn.

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 KẾT LUẬN

Với đề tài này, mô hình sản phẩm “hệ thống trồng cây rau mầm thông minh” được thiết kế có một số ưu điểm và nhược điểm sau:

4.1.1 Ưu điểm

- Tạo ra môi trường có điều kiện sống phù hợp cho cây trồng.
- Giao diện thân thiện, dễ dàng sử dụng một cách nhanh chóng.
- Chi phí đầu tư thiết kế thấp.

4.1.2 Nhược điểm

- Vẫn còn một số lỗi phát sinh trong quá trình hoạt động.
- Với chi phí đầu tư cạnh tranh, nên các thiết bị chưa thực sự bền bỉ, chống chọi với các tác nhân môi trường lâu dài được.

- Sai số trong quá trình đọc cảm biến.
- Giao diện màn hình LCD khó quan sát từ xa.

4.2 HƯỚNG PHÁT TRIỂN

- Khắp phục các lỗi còn phát sinh.
- Sử dụng các thiết bị có khả năng chống chịu với các tác nhân môi trường tốt hơn.
- Áp dụng công nghệ năng lượng mặt trời, thay thế cho một phần hoặc hoàn toàn nguồn năng lượng từ lưới điện mà hệ thống nhà vườn đang sử dụng.
- Áp dụng điều khiển và thông báo lỗi qua tin nhắn sms.
- Áp dụng điều khiển và giám sát qua mạng internet.

TÀI LIỆU THAM KHẢO

1. Nguyễn Thúy Loan (2013). *Lập trình cơ bản C*. Đại học Công nghệ TP. Hồ Chí Minh
2. Nguyễn Trung Tín (2014). *Giáo trình Arduino cơ bản*. Học viện Hàng không Việt Nam.
3. Phạm Quốc Phương (2015). *Vi điều khiển*. Đại học Công nghệ TP. Hồ Chí Minh.
4. *Thực phẩm sạch – nhu cầu thiết yếu của con người*, 05/2017, <http://rausachsaigon.com.vn/thuc-pham-sach-nhu-cau-thiet-yeu-cua-con-nguoi/>
5. *Viễn cảnh "chết chóc" khi máy móc thay thế hoàn toàn con người*, 05/2017, <http://kenh14.vn/kham-pha/vien-canhh-chet-choc-khi-may-moc-thay-the-hoan-toan-con-nguoi-2015080604593319.chn>
6. *Lịch sử ra đời Arduino*, 05/2017, <https://blackconsole.net/2016/01/17/lich-su-ra-doi-arduino/>
7. *So sánh một số loại Arduino*, 05/2017, <http://www.giaiphaptt.vn/2015/07/so-sanhh-mot-so-loai-arduino.html>
8. *Đọc nhiệt độ - độ ẩm và xuất ra màn hình LCD*, 05/2017, <http://arduino.vn/bai-viet/91-doc-nhieth-do-do-am-va-xuat-ra-man-hinh-lcd>
9. *Cảm biến độ ẩm đất và những ứng dụng hay của nó*, 05/2017, <http://arduino.vn/bai-viet/917-cam-bien-dat-va-nhung-ung-dung-hay-cua-no>

PHỤ LỤC

Code chương trình

```
#include <RTCLib.h> //thời gian thực
#include <SoftwareSerial.h> //bluetooth
#include <DHT.h> //độ ẩm
#include <BH1750.h> //as
#include <Wire.h> //cld
#include <LiquidCrystal_I2C.h> //kết nối i2c

SoftwareSerial mySerial(52, 53); //kết nối bluetooth rx,tx
float t;
uint16_t lux;
uint16_t percent;

LiquidCrystal_I2C lcd(0x27,20,4);
RTC_DS1307 RTC;
BH1750 lightMeter;

const int DHTPIN = 2;
const int DHTTYPE = DHT11;
DHT dht(DHTPIN, DHTTYPE);

byte degree[8] = {
    0B01110,
    0B01010,
    0B01110,
    0B00000,
```

```
0B00000,  
0B00000,  
0B00000,  
0B00000  
};
```

```
const int numRows = 4;    // Tổng số hàng của Keypad  
const int numCols = 4;    // Tổng số cột của Keypad  
const int debounceTime = 10; // Delay mỗi lần nhấn  
const char keys[numRows][numCols] = {  
    {'1','2','3','A'},  
    {'4','5','6','B'},  
    {'7','8','9','C'},  
    {'*','0','#','D'},  
};
```

```
const int colPins[numCols] = {5, 6, 7, 8};  
const int rowPins[numRows] = {9, 10, 11, 12};
```

```
#define maybom 50  
#define maylanh 49  
#define mayquat 47  
#define tb 45
```

```
int nd1 = 25;  
int nd2 = 30;  
int da1 = 50;
```

```
int da2 = 70;  
int as1 = 70;  
int as2 = 100;
```

```
void setup() {
```

```
    pinMode(tb, OUTPUT);  
    pinMode(maybom, OUTPUT);  
    pinMode(maylanh, OUTPUT);  
    pinMode(mayquat, OUTPUT);
```

```
    for (int row = 0; row < numRows; row++){  
        pinMode(rowPins[row],INPUT);    // Hàng là Input  
        digitalWrite(rowPins[row],HIGH);  
    }  
    for (int column = 0; column < numCols; column++){  
        pinMode(colPins[column],OUTPUT);    // Cột là Output  
        digitalWrite(colPins[column],HIGH);  
    }
```

```
    Wire.begin();//khởi tạo thư viện tham gia kết nối i2c.  
    Wire.beginTransmission(0x68);//bắt đầu truyền dữ liệu có địa chỉ 0x68  
    Wire.write(0x07);  
    Wire.write(0x10);  
    Wire.endTransmission();//ngưng quá trình truyền dữ liệu  
    Wire.requestFrom(0x68, 7);
```

```
lcd.begin(20,4); khai báo thư viện dùng lcd 20x4
lcd.backlight();//bật đèn
lcd.createChar(1, degree);//địa chỉ chữ độ
dht.begin();
lightMeter.begin();
mySerial.begin(9600);//khởi tạo kết nối bluetooth
Serial.begin(9600);//kết nối với máy tính
```

```
RTC.begin();
if (! RTC.isrunning()) {
  Serial.println("RTC is NOT running!");
  RTC.adjust(DateTime(__DATE__, __TIME__));
  RTC.adjust(DateTime(2018, 1, 14, 11, 45, 00));
}
```

```
digitalWrite(maybom,HIGH);
digitalWrite(maylanh,HIGH);
digitalWrite(mayquat,HIGH);
digitalWrite(tb,HIGH);
}
```

```
void cb(){
  // DHT11
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(t) || isnan(h))
```

```
{  
  
}  
else  
{  
    lcd.setCursor(0,2);  
    lcd.print("ND: ");  
    lcd.setCursor(4,2);  
    lcd.print(round(t));  
    lcd.write(1);  
    lcd.print("C");  
  
}  
  
// AS  
uint16_t lux = lightMeter.readLightLevel();  
lcd.setCursor(10, 2);  
lcd.print("AS:  ");  
lcd.setCursor(14, 2);  
lcd.print(lux);  
lcd.setCursor(17, 2);  
lcd.print(" lx");  
  
// DAD  
int value = analogRead(A0);  
uint16_t percent = map(value, 1023, 0, 0, 100);  
lcd.setCursor(0, 3);  
lcd.print("DO AM:  ");
```

```
lcd.setCursor(7, 3);  
lcd.print(percent );  
lcd.setCursor(10, 3);  
lcd.print("%");  
}  
  
void tg(){  
    DateTime now = RTC.now();  
    lcd.setCursor(0, 1);  
    lcd.print("TIME:");  
    lcd.setCursor(6, 1);  
    lcd.print(now.hour(), DEC);  
    lcd.print(":");  
    lcd.print(now.minute(), DEC);  
    lcd.print(":");  
    lcd.print(now.second(), DEC);  
    lcd.print(" ");  
    lcd.setCursor(0, 0);  
    lcd.print("DAY:");  
    lcd.setCursor(6, 0);  
    lcd.print(now.day(), DEC);  
    lcd.print("/");  
    lcd.print(now.month(), DEC);  
    lcd.print("/");  
    lcd.print(now.year(), DEC);  
    lcd.print(" ");  
}
```



```
void loop() {
  char key = getKey();
  // bluetooth
  char val;
  if(mySerial.available()>0){
    delay(3);
    val=mySerial.read();
  }
  if(val == '1'){digitalWrite(maybom,LOW);}
  else if(val == '2'){digitalWrite(maylanh,LOW);}
  else if(val == '3'){digitalWrite(mayquat,LOW);}
  else if(val == '4'){digitalWrite(tb,LOW);}
  else if(val == 'A'){digitalWrite(maybom,HIGH);}
  else if(val == 'B'){digitalWrite(maylanh,HIGH);}
  else if(val == 'C'){digitalWrite(mayquat,HIGH);}
  else if(val == 'D'){digitalWrite(tb,HIGH);}
  else if(val == 'I'){
    digitalWrite(maybom,HIGH);
    digitalWrite(maylanh,HIGH);
    digitalWrite(mayquat,HIGH);
    digitalWrite(tb,HIGH);
  }
  cb();
  tg();
  if(key!= 0 || val == '9'){
    while( key == '*' || val == '9'){
      char key = getKey();
      int value = analogRead(A0);
```

```
uint16_t percent = map(value, 1023, 0, 0, 100);
uint16_t lux = lightMeter.readLightLevel();
float t = dht.readTemperature();

tg();
cb();
if(percent < da1) digitalWrite(maybom,LOW);
if(percent > da2) digitalWrite(maybom,HIGH);
if(t > nd2) digitalWrite(maylanh,LOW);
if(t < nd1) digitalWrite(maylanh,HIGH);
if(lux < as1) digitalWrite(tb,LOW);
if(lux > as2) digitalWrite(tb,HIGH);
if(key == '0' || key == 'D' || mySerial.available()>0){
    digitalWrite(maybom,HIGH);
    digitalWrite(maylanh,HIGH);
    digitalWrite(mayquat,HIGH);
    digitalWrite(tb,HIGH);
    break;
}
}

while( key == '#'){
    char key = getKey();
    tg();
    cb();
    if (key == '1')digitalWrite(maybom,LOW);
    if (key == '4')digitalWrite(maybom,HIGH);
    if (key == '2')digitalWrite(maylanh,LOW);
```

```
if (key == '5')digitalWrite(maylanh,HIGH);
if (key == '3')digitalWrite(mayquat,LOW);
if (key == '6')digitalWrite(mayquat,HIGH);
if (key == 'A')digitalWrite(tb,LOW);
if (key == 'B')digitalWrite(tb,HIGH);
if(key == 'D'){
    digitalWrite(maybom,HIGH);
    digitalWrite(maylanh,HIGH);
    digitalWrite(mayquat,HIGH);
    digitalWrite(tb,HIGH);
}
if(key == '0'){
    digitalWrite(maybom,HIGH);
    digitalWrite(maylanh,HIGH);
    digitalWrite(mayquat,HIGH);
    digitalWrite(tb,HIGH);
    break;
}
}
}
}
char getKey(){
    char key = 0;
    for(int column = 0; column < numCols; column++)
    {
        digitalWrite(colPins[column],LOW);
        for(int row = 0; row < numRows; row++)
        {
```

```
if(digitalRead(rowPins[row]) == LOW)
{
    delay(debounceTime);
    while(digitalRead(rowPins[row]) == LOW)
    {
        key = keys[row][column
    }
}
digitalWrite(colPins[column],HIGH);
}
return key;
}
```