

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN ĐIỆN TỬ - VIỄN THÔNG



ĐỒ ÁN MÔN HỌC 1

Đề tài:

MÔ HÌNH NÔNG NGHIỆP THÔNG MINH

SVTH: NGUYỄN VĂN GIỎI

MSSV: 14119013

SVTH: DANH QUANG VŨ

MSSV: 14119068

GVHD: Ths. HUỲNH HOÀNG HÀ

TP.Hồ Chí Minh – tháng 7 năm 2017

LỜI CẢM ƠN

*Nhóm nghiên cứu xin gửi lời cảm ơn sâu sắc và chân thành tới thầy **HUYỀN HOÀNG HÀ** đã tận tình giúp đỡ, chỉ bảo, hướng dẫn nhóm thực hiện trong suốt quá trình thực hiện đề tài này. Trong thời gian làm việc với thầy, nhóm không những tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần làm việc, thái độ nghiên cứu khoa học nghiêm túc, hiệu quả của thầy đây là những điều rất cần thiết cho nhóm trong quá trình học tập và công tác sau này.*

Chân thành cảm ơn đến các bạn đã luôn đồng hành, giúp đỡ để nhóm có thể thực hiện đề tài này thành công.

Người thực hiện
Nguyễn Văn Giới
Danh Quang Vũ

TÓM TẮT

Hiện nay trên thế giới, việc ứng dụng các công nghệ điều khiển vào trong sản xuất là rất nhiều và cần thiết trong các ngành nghề kể cả trong nông nghiệp. Trong nông nghiệp, nhờ ứng dụng các công nghệ điều khiển hiện đại mà năng suất và chất lượng cây trồng tăng lên đáng kể. Với công nghệ trồng rau có sự hỗ trợ của các thiết bị theo dõi và điều khiển đã cho những kết quả ngoài mong đợi như: năng suất cao, chất lượng tốt, sạch, an toàn mà còn có thể trồng những loại cây mà từ trước không phải là truyền thống của vùng miền.

Ở Việt Nam, việc trồng rau ứng dụng công nghệ cũng đang từng bước phát triển nhanh chóng, đem lại lợi ích cao cho người nông dân. Tuy nhiên, do chi phí đầu tư cao nên việc ứng dụng chỉ hạn chế trong các nông trang lớn có khả năng về kinh tế, hoặc chỉ điều khiển ở dạng bán tự động nên vẫn cần nhiều nhân công trong việc điều khiển vì hầu hết các trang thiết bị điều khiển đều phải nhập từ nước ngoài nên giá thành cao. Do đó cần phải có hướng nghiên cứu, thiết kế và chế tạo ra những thiết bị này ngay ở trong nước để giảm chi phí, phù hợp với điều kiện kinh tế của người nông dân và điều kiện môi trường, khí hậu ở Việt Nam.

Vì lý những do này, nhóm đã chọn lựa đề tài “Mô hình nông nghiệp thông minh”.

Trong quá trình thực hiện đề tài không tránh khỏi những sai sót do kiến thức có giới hạn, cũng như tham khảo từ nhiều nguồn tài liệu từ internet, sách, báo... Rất mong nhận được sự đóng góp của quý thầy cô cũng như các bạn để đề tài thực hiện thành công và phát triển hơn nữa.

Chân thành cảm ơn.

Người thực hiện đề tài
Nguyễn Văn Giới
Danh Quang Vũ

MỤC LỤC

CHƯƠNG 1	1
TỔNG QUAN ĐỀ TÀI	1
1.1 Tình hình nghiên cứu hiện nay	1
1.2 Lý do chọn đề tài	1
1.3 Mục tiêu của đề tài	1
1.4 Đối tượng và phạm vi nghiên cứu	2
1.5 Giới hạn đề tài	2
1.6 Bố cục của Đồ án môn học	2
CHƯƠNG 2	4
CƠ SỞ LÝ THUYẾT	4
2.1 Kit ARM STM32F103C8T6	4
2.1.1 Giới thiệu vi xử lý ARM	4
2.1.2 Lịch sử phát triển ARM	5
2.1.3 Giới thiệu về ARM Cortex	6
2.1.4 Các thành phần KIT STM32F103C8T6	8
2.2 DHT11	9
2.2.1 Giới thiệu	9
2.2.2 Thông số kỹ thuật	10
2.2.3 Nguyên lý hoạt động	10
2.3 Module Cảm biến độ ẩm đất	13
2.3.1 Giới thiệu	13
2.3.2 Thông số kỹ thuật	14
2.3.3 Nguyên lý hoạt động	15
2.4 LCD 16x02	16
2.4.1 Giới thiệu	16

2.4.2	Hình dáng và kích thước.....	16
2.4.3	Sơ đồ chân LCD 16x02.....	16
2.4.4	Địa chỉ từng ký tự trên LCD 16x02.....	17
2.4.5	Bộ điều khiển LCD và các vùng nhớ.....	18
2.4.6	Mã ASCII hiển thị LCD 16x02.	19
2.5	Module relay 2 kênh.....	20
2.5.1	Giới thiệu.	20
2.5.2	Cấu tạo relay.	20
2.5.3	Sơ đồ nguyên lý module relay 2 kênh.	21
2.6	Hệ điều hành thời gian thực RTOS.	23
2.6.1	Giới thiệu.	23
2.6.2	Tải thư viện hỗ trợ RTOS.	23
2.7	Trình biên dịch Keil C cho ARM.	24
2.7.1	Lý do lựa chọn.	24
2.7.2	Giới thiệu về Keil C.....	25
CHƯƠNG 3.....		26
PHÂN TÍCH LỰA CHỌN PHẦN CỨNG HỆ THỐNG		26
3.1	Yêu cầu hệ thống.	26
3.2	Sơ đồ khối hệ thống.....	26
3.3	Phân tích và kết nối phần cứng.....	27
3.3.1	Khối Kit ARM STM32F103C8T6.....	27
3.3.2	Khối nguồn.	28
3.3.3	Khối PC.....	29
3.3.4	Khối cảm biến.....	30
3.3.5	Khối điều khiển bằng tay.....	31
3.3.6	Khối hiển thị LCD.	32
3.3.7	Khối relay 2 kênh.....	35

3.4	Sơ đồ kết nối mạch.	36
CHƯƠNG 4.....		38
LƯU ĐỒ GIẢI THUẬT TRONG HỆ THỐNG		38
4.1	Hàm chính.....	38
4.2	Hàm con.....	39
4.2.1	Lưu đồ tạo và hiển thị ký tự ở CGRAM LCD.....	39
4.2.2	Lưu đồ nhấp nháy led C13.....	41
4.2.3	Lưu đồ đọc thông số cảm biến.....	42
4.2.4	Lưu đồ hiển thị thông số lên LCD.	43
4.2.5	Lưu đồ điều khiển relay.	44
CHƯƠNG 5.....		50
KẾT QUẢ VÀ ĐÁNH GIÁ		50
5.1	Sơ đồ thiết kế hệ thống.	50
5.2	Sản phẩm hoàn thiện.....	51
5.3	Đánh giá đề tài.	52
5.3.1	Ưu điểm.	52
5.3.2	Nhược điểm.....	52
CHƯƠNG 6.....		54
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		54
6.1	Kết luận.....	54
6.1.1	Những vấn đề nghiên cứu.....	54
6.1.2	Những vấn đề hoàn thành.	54
6.1.3	Những hạn chế của đề tài.....	54
6.2	Hướng phát triển của đề tài.	54
TÀI LIỆU THAM KHẢO.....		55
PHỤ LỤC.....		56

DANH MỤC TỪ VIẾT TẮT

RTOS:	Real-time Operating System.
LCD:	Liquid Crystal Display.
ARM:	Advanced RISC Machine.
RISC:	Reduced Instruction Set Computer.
CPU:	Central Processing Unit.
CAN:	Controller Area Network.
DMA:	Direct Marketing Association.
USART:	Universal Synchronous/Asynchronous Receiver/Transmitter.
SPI:	Serial Peripheral Interface.
USB:	Universal Serial Bus.
WDT:	Watchdog Timer.
SRAM:	Static Random Access Memory.
Op-amp:	Operational Amplifier.
DDRAM :	Display Data RAM.
CGROM :	Character Generator ROM.
CGRAM :	Character Generator RAM.
ASCII:	American Standard Code for Information Interchange.

DANH MỤC HÌNH

Hình 2. 1 Một số ứng dụng của ARM.....	4
Hình 2. 2 Vi điều khiển ARM STM32F103C8T6.....	6
Hình 2. 3 Các thành phần của KIT STM32F103C8T6.....	8
Hình 2. 4 Hình ảnh thực tế và sơ đồ chân.....	9
Hình 2. 5 Kích thước và khoảng cách chân DHT11	10
Hình 2. 6 Kết nối giữa MCU và DHT11.....	10
Hình 2. 7 Gửi tín hiệu.....	11
Hình 2. 8 Nhận dữ liệu bit ‘0’	12
Hình 2. 9 Nhận dữ liệu bit “1”	12
Hình 2. 10 Hình ảnh thực tế của module.	13
Hình 2. 11 Module chính của cảm biến độ ẩm đất	14
Hình 2. 12 Hình ảnh thực tế và sơ đồ chân LM393.....	15
Hình 2. 13 Sơ đồ nguyên lý của module cảm biến	15
Hình 2. 14 Hình dáng của loại LCD 16x02	16
Hình 2. 15 Sơ đồ khối của bộ điều khiển LCD.....	18
Hình 2. 16 Hình dáng thực của relay	20
Hình 2. 17 Cấu tạo của relay và sơ đồ chân.....	20
Hình 2. 18 Sơ đồ nguyên lý module relay 2 kênh.....	21
Hình 2. 19 Module relay dùng BJT PNP	22
Hình 2. 20 Giao diện trang chủ FreeRTOS.....	23
Hình 2. 21 Nguồn tải thư viện RTOS	24
Hình 2. 22 “FreeRTOSv9.0.0.exe” sau khi extract.....	24
Hình 2. 23 Trang chủ để tải Keil C.	25
Hình 2. 24 Giao diện Keil C.....	25
Hình 3. 1 Sơ đồ khối tổng quát của hệ thống.....	26
Hình 3. 2 Pinout và chức năng từng chân của KIT STM32F103C8T6.	28
Hình 3. 3 Sơ đồ khối nguồn cung cấp.....	28
Hình 3. 4 Datasheet của LM7805C.....	29
Hình 3. 5 Kết nối ST-Link V2 mini với KIT.	30

Hình 3. 6 Kết nối nút nhấn với KIT	31
Hình 3. 7 Dòng điện tại I/O của KIT	32
Hình 3. 8 Ký tự do nhóm nghiên cứu tự tạo tại vùng nhớ CGRAM	32
Hình 3. 9 Các đoạn tạo nên ký tự lớn trên LCD	33
Hình 3. 10 Ký tự tự tạo được lưu trong CGRAM.....	34
Hình 3. 11 Hình ảnh thực tế module relay 2 kênh.....	35
Hình 3. 12 Sơ đồ kết nối chân trong mô hình	36
Hình 4. 1 Lưu đồ hàm chính của chương trình.	38
Hình 4. 2 Lưu đồ tạo ký tự và hiển thị từ CGRAM.....	39
Hình 4. 3 Lưu đồ hiển thị chữ “DA – 1”.....	40
Hình 4. 4 Lưu đồ nhấp nháy led C13.....	41
Hình 4. 5 Đọc thông số thứ 2 cảm biến.	42
Hình 4. 6 Hiển thị thông số lên LCD.....	43
Hình 4. 7 Lưu đồ điều khiển relay.	44
Hình 4. 8 Lưu đồ điều khiển động cơ.	45
Hình 4. 9 Lưu đồ điều khiển đèn tự động.....	46
Hình 4. 10 Lưu đồ điều khiển động cơ bằng tay.....	47
Hình 4. 11 Lưu đồ điều khiển đèn bằng tay.....	48
Hình 5. 1 Sơ đồ layout thiết kế trên Proteus	50
Hình 5. 2 Sơ đồ mạch in.....	50
Hình 5. 3 Sản phẩm mạch in	51
Hình 5. 4 Mặt trước của sản phẩm.....	51
Hình 5. 5 Mặt nhìn từ trên xuống.....	52

DANH MỤC BẢNG

Bảng 2. 1 Các dòng phát triển của ARM	5
Bảng 2. 2 Thông số kỹ thuật ARM STM32F103C8T6	7
Bảng 2. 3 Chức năng lựa chọn Boot trong kit	8
Bảng 2. 4 Chức năng các chân của LCD 16x02	17
Bảng 2. 5 Địa chỉ trên LCD 16x02	17
Bảng 2. 6 Bảng mã ASCII.....	19
Bảng 3. 1 Kết nối giữa mạch nạp và KIT	30
Bảng 3. 2 Kết nối DHT11 với KIT	30
Bảng 3. 3 Sơ đồ kết nối giữa cảm biến độ ẩm đất và KIT	31
Bảng 3. 4 Thiết lập địa chỉ CGRAM	33
Bảng 3. 5 Vùng nhớ CGRAM.....	33
Bảng 3. 6 Sơ đồ nối chân của LCD và KIT.	35
Bảng 3. 7 Kết nối chân của module relay 2 kênh.	36

CHƯƠNG 1

TỔNG QUAN ĐỀ TÀI

1.1 Tình hình nghiên cứu hiện nay.

Hiện nay, xã hội ngày càng phát triển và khoa học kỹ thuật ngày càng tiên tiến hơn thì nhu cầu về kỹ thuật càng được quan tâm và phát triển hơn. Đặc biệt là những công trình nghiên cứu khoa học nhằm thiết kế ra những sản phẩm ứng dụng có ý nghĩa thiết thực vào cuộc sống. Nó có thể thay thế con người, giảm lao động chân tay hoặc giúp con người quan sát, kiểm tra những nơi độc hại, nguy hiểm mà cơ thể con người không thể chịu đựng được.

1.2 Lý do chọn đề tài.

Thấy được tầm quan trọng và nhu cầu thiết yếu của xã hội, đồng thời trong quá trình học tập thì nhóm nghiên cứu được học khá nhiều về kit nhúng và cách hoạt động của các cảm biến. Và bây giờ, thông qua đồ án môn học 1 này, nhóm nghiên cứu muốn tạo ra một sản phẩm sử dụng các tính năng của kit nhúng và các loại cảm biến. Nên nhóm nghiên cứu quyết định chọn đề tài: “*Mô hình nông nghiệp thông minh*”.

1.3 Mục tiêu của đề tài.

Đề tài triển khai nghiên cứu hướng tới mục tiêu sau:

Theo dõi nhiệt độ, độ ẩm không khí thông qua cảm biến dht11 và độ ẩm đất qua cảm biến độ ẩm đất.

Điều chỉnh hệ thống thông qua 3 nút nhấn: nút chuyển chế độ điều khiển tự động và điều khiển bằng tay, nút nhấn điều khiển relay đèn và nút nhấn điều khiển relay bơm nước.

Nếu nút điều khiển chế độ online/offline ở chế độ online thì hệ thống điều khiển đèn và bơm nước thông qua 2 relay sẽ dựa vào giá trị cài đặt tự động trong chương trình để hoạt động phù hợp với loại cây mà chủ trang trại đang canh tác. Nếu nút điều khiển chế độ online/offline ở chế độ offline thì hệ thống điều khiển đèn và bơm nước thông qua 2 relay sẽ dựa vào trạng thái của 2 nút nhấn còn lại mà hoạt động, bất chấp giá trị được đặt trước.

Tất cả các nhiệm vụ: đọc cảm biến, điều khiển nút nhấn, hiển thị thông số đều hoạt động song song thông qua RTOS.

Và tất nhiên, tất cả trạng thái và thông số đều được hiển thị trên màn hình LCD 16x02.

1.4 Đối tượng và phạm vi nghiên cứu.

Đối tượng:

Kit ARM STM32F103C8T6.

Module relay 2 kênh tích cực mức cao.

Nút nhấn điều khiển.

Cảm biến dht11, cảm biến độ ẩm đất.

Phạm vi nghiên cứu.

Nghiên cứu trong lĩnh vực lập trình nhúng trên STM32F103C8T6.

Nghiên cứu phần mềm hỗ trợ điều khiển trên STM32F103C8T6 như Keil C, STM32CubeMX.

Nghiên cứu hệ thống RTOS – hoạt động song song giữa các tác vụ trên ARM.

Cách kết nối vật lý giữa cảm biến, nút nhấn, LCD với STM32F103C8T6.

Tìm hiểu nguyên lý hoạt động của các cảm biến cần sử dụng.

Phương pháp nghiên cứu.

Để giải quyết vấn đề, ta thực hiện phương pháp nghiên cứu như sau:

Nghiên cứu về kit STM32F103C8T6.

Thiết kế board mạch và hộp kết nối với kit để bảo vệ kit trong môi trường ngoài trời.

Nghiên cứu viết code và kết nối với kit thông qua ST-Link V2 mini, chuẩn giao tiếp USB.

Xử lý dữ liệu của các cảm biến để hiển thị lên LCD 16x02.

1.5 Giới hạn đề tài.

Đề tài chỉ tìm hiểu trong phạm vi như sau:

Tìm hiểu cách hoạt động của kit và các task vụ hoạt động song song dựa vào RTOS.

Tìm hiểu các đọc các cảm biến dht11, độ ẩm đất và hiển thị lên LCD 16x02.

Theo dõi và điều khiển hệ thống trực tiếp, chưa hỗ trợ điều khiển và theo dõi từ xa.

1.6 Bố cục của Đồ án môn học.

Nội dung chính của đề tài được trình bày với năm chương như sau:

Chương 1 Tổng quan đề tài.

Trong chương này nêu ra được tình hình nghiên cứu hiện nay, lý do và mục tiêu chọn đề tài, đối tượng và phạm vi nghiên cứu của đề tài, phương pháp nghiên cứu và giới hạn của đề tài.

Chương 2: Cơ sở lý thuyết.

Các lý thuyết chính liên quan đến các thành phần phần cứng và phần mềm cả hệ thống được trình bày như sau:

- STM32F103C8T6.
- Các loại cảm biến được dùng: cảm biến nhiệt độ, độ ẩm: dht11; cảm biến độ ẩm đất.
- Màn hình hiển thị LCD 16x02.
- Module relay điều khiển động cơ và đèn.
- Hệ điều hành thời gian thực – RTOS.
- Trình biên dịch Keil C.

Chương 3: Phân tích lựa chọn hệ thống phần cứng.

Trong chương này, mục đích là thiết kế phần cứng cho hệ thống với những yêu cầu đặt ra với hệ thống. Từ sơ đồ khối tổng quát và sơ đồ khối chi tiết để tiến hành lựa chọn các linh kiện cho các khối.

Phân tích các khối của phần cứng hệ thống:

- Khối KIT STM32F103C8T6.
- Khối PC.
- Khối cảm biến.
- Khối điều khiển bằng tay.
- Khối hiển thị LCD.
- Khối relay 2 kênh.

Xây dựng chương trình trên STM32F103C8T6 để thực hiện quá trình thực hiện song song các hoạt động: đọc cảm biến, hiển thị LCD 16x02, điều khiển relay thông qua nút nhấn.

Chương 4: Lưu đồ giải thuật trong hệ thống.

Chương này trình bày về lưu đồ giải thuật của chương trình chính và của tất cả các chương trình con trong hệ thống.

Giải thích hoạt động.

Chương 5: Kết quả và đánh giá.

Trình bày về kết quả thi công phần cứng, sản phẩm hoàn thiện.

Đánh giá ưu nhược điểm của hệ thống.

Chương 6: Kết luận và hướng phát triển.

Đưa ra các kết luận về những vấn đề mà trong quá trình nghiên cứu đã đạt được, chưa đạt được và những hạn chế của đề tài.

Đưa ra hướng phát triển đề tài trong tương lai.

CHƯƠNG 2

CƠ SỞ LÝ THUYẾT

2.1 Kit ARM STM32F103C8T6.

2.1.1 Giới thiệu vi xử lý ARM.

Cấu trúc của ARM (Acorn RISC Machine) là cấu trúc vi xử lý 32 bit kiểu kiến trúc tập lệnh RISC (thuộc kiến trúc Harvard, có tập lệnh rút gọn) được sử dụng rộng rãi trong ngành thiết kế và lập trình nhúng. Vì có ưu điểm là tiết kiệm năng lượng, các bộ CPU ARM được ứng dụng rộng rãi trong các sản phẩm điện tử nhỏ gọn, vì đối với các sản phẩm này thì việc tiêu tán công suất thấp là mục tiêu thiết kế quan trọng được đặt lên hàng đầu.



Hình 2. 1 Một sơ ứng dụng của ARM

Ngày nay, CPU ARM được tìm thấy khắp mọi nơi trong các sản phẩm thương mại điện tử, thiết bị cầm tay (điện thoại di động, máy tính cầm tay,...) cho đến các thiết bị ngoại vi máy tính (ổ đĩa cứng, bộ định tuyến router,..).

2.1.2 Lịch sử phát triển ARM.

Kiến trúc	Số bit	Tên lõi
ARMv1	32/26	ARM1.
ARMv2	32/26	ARM2, ARM250, ARM3.
ARMv3	32	ARM6, ARM7.
ARMv4	32	ARM8.
ARMv4T	32	ARM7TDMI, ARM9TDMI, SecurCore SC100.
ARMv5TE	32	ARM7EJ, ARM9E, ARM10E.
ARMv6	32	ARM11.
ARMv6-M	32	ARM Cortex-M0, ARM Cortex-M0+, ARM Cortex-M1.
ARMv7-M	32	ARM Cortex-M3, SecurCore SC300.
ARMv7E-M	32	ARM Cortex-M4, ARM Cortex-M7.
ARMv8-M	32	ARM Cortex-M23, ARM Cortex-M33.
ARMv7-R	32	ARM Cortex-R4, ARM Cortex-R5, ARM Cortex-R7.
ARMv7-A	32	ARM Cortex-A5, ARM Cortex-A7, ARM Cortex-A8, ARM Cortex-A9, ARM Cortex-A12, ARM Cortex-A15, ARM Cortex-A17.
ARMv8-A	32	ARM Cortex-A32.
ARMv8-A	64/32	ARM Cortex-A35, ARM Cortex-A53, ARM Cortex-A57, ARM Cortex-A72, ARM Cortex-A73.

Bảng 2. 1 Các dòng phát triển của ARM

ARM được bắt đầu thiết kế từ năm 1983 trong một dự án phát triển của công ty máy tính Acorn.

Nhóm thiết kế hoàn thành việc phát triển mẫu vào năm 1985: ARM1, và vào năm sau, nhóm hoàn thành sản phẩm ‘‘thực’’ gọi là ARM2.

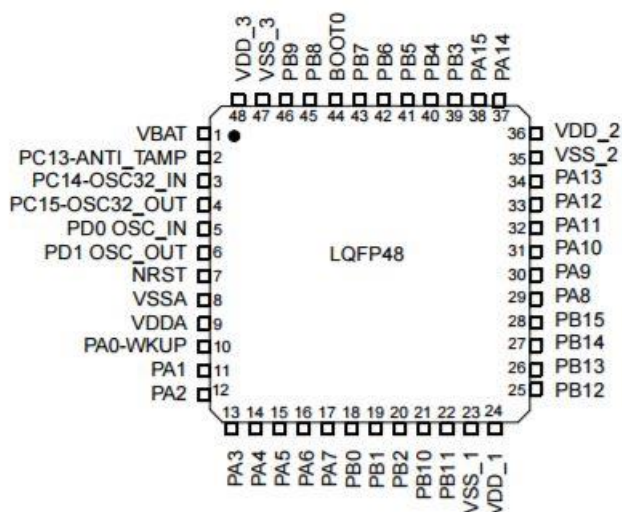
ARM2 có tuyến dữ liệu 32-bit, không gian địa chỉ 26-bit tức cho phép quản lý đến 64 Mbyte địa chỉ và 16 thanh ghi 32-bit. Thế hệ sau, ARM3, được tạo ra với 4KB cache và có chức năng được cải thiện tốt hơn nữa cho các phiên bản tiếp theo.

Vào cuối thập niên 80, hãng Apple Computer hợp tác với Acorn để phát triển các lõi ARM mới và tạo ra ARM6.

Trải qua nhiều thế hệ nhưng lõi ARM gần như không thay đổi kích thước. ARM2 có 30.000 transistors trong khi ARM6 chỉ tăng lên đến 35.000. Ý tưởng của nhà sản xuất lõi ARM là: người sử dụng có thể ghép lõi ARM với một số bộ phận tùy chọn nào đó để

tạo ra một CPU hoàn chỉnh, một loại CPU mà có thể tạo ra trên những nhà máy sản xuất bán dẫn cũ và vẫn tiếp tục tạo ra được sản phẩm với nhiều tính năng mà giá thành vẫn thấp có thể.

2.1.3 Giới thiệu về ARM Cortex.



Hình 2. 2 Vi điều khiển ARM STM32F103C8T6

Dòng ARM Cortex là một bộ xử lý thế hệ mới đưa ra một kiến trúc chuẩn cho nhu cầu đa dạng về công nghệ. Mặc dù vi điều khiển 8-bit rất phổ biến trong quá khứ, nhưng ARM-Cortex đã dần dần thay thế thị trường 8-bit vì giá rẻ hơn và tối ưu về hiệu suất hơn. Để phù hợp với nhu cầu sử dụng, ARM Cortex được chia làm 3 dòng chính:

Cortex-A: bộ xử lý dành cho các ứng dụng cấp cao và phức tạp như hệ điều hành.

Cortex-R: bộ xử lý dành cho các hệ thống đòi hỏi sự khắc khe về đáp ứng thời gian thực.

Cortex-M: bộ xử lý dành cho các dòng vi điều khiển, được thiết kế tối ưu về giá thành.

Dòng ARM STM32 có lõi Cortex-M3. Dòng Cortex-M3 được thiết kế đặc biệt để nâng cao hiệu suất hệ thống, kết hợp với tiêu thụ năng lượng thấp.

STM32F103C8T6 là một trong những kit nhúng được ứng dụng nhiều, có thể lập trình điều khiển ngõ ra theo ý muốn của người lập trình. Ngôn ngữ lập trình điều khiển không quá phức tạp.

Tên gọi	STM32F103C8T6
Mô tả	MCU ARM 64KB FLASH MEM 48-LQFP
Nhà sản xuất	STMicroelectronics

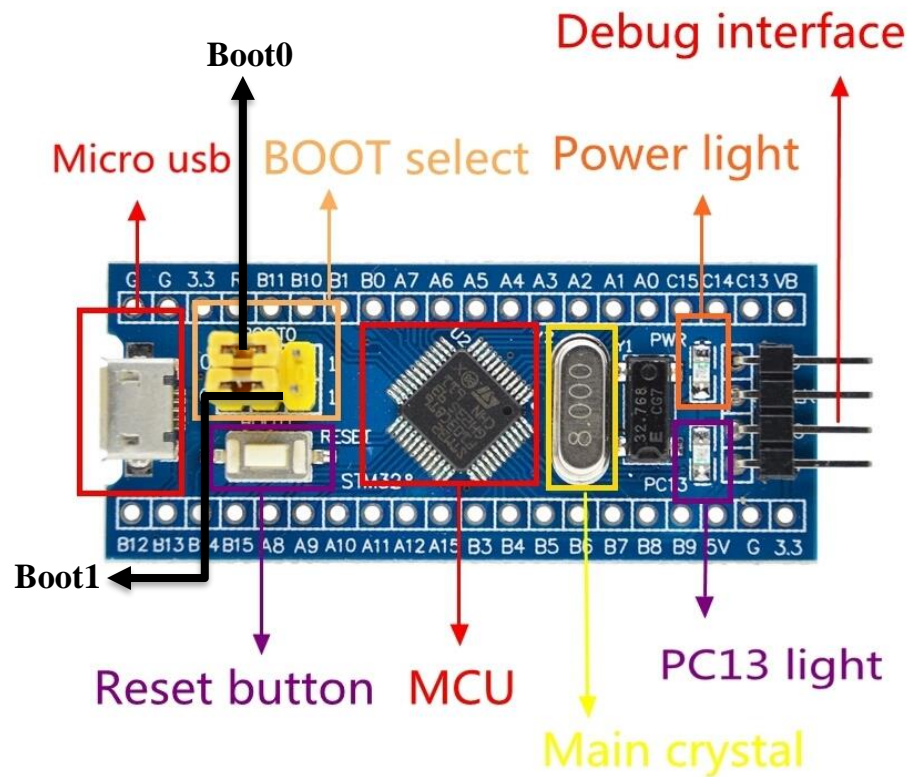
Loại	Mạch tích hợp
Bộ nhớ Flash	64KB (64K x 8)
Bộ nhớ RAM	20K x 8
Số lượng I/O	37
Kiểu đóng gói chân	LQFP48
Tốc độ tối đa	72MHz
Kiểu xung	Dùng xung nội
Kiểu Memory	FLASH
Lõi	ARM® Cortex®-M3
Kích thước Core	32-Bit
Nhiệt độ hoạt động	-40°C ~ 85°C
Kết nối	CAN, I2C, DMA, SPI, UART/USART, USB
Ngoại vi	DMA, động cơ servo PWM, động cơ bước, WDT
Điện áp hoạt động	2 V ~ 3.6 V

Bảng 2. 2 Thông số kỹ thuật ARM STM32F103C8T6

CPU do ST sản xuất dựa trên ARM Cortex-M3 chip xử lý 32-bit STM32F103C8T6 với hình dạng vỏ LQFP48 có 48 chân, có 64 KB bộ nhớ flash, có 20 KB bộ nhớ RAM và tích hợp thêm các ngoại vi 12 bit ADC, DAC, PWM, CAN, SDIO và các nguồn tài nguyên khác.

- Là vi xử lý hiệu suất xử lý 32-bit ARM Cortex-M3 với cấu trúc RISC.
- Tần số hoạt động là 72 MHz, với 1.25 DMIPS/MHZ.
- Kiểu đóng gói LQFP48 với 48 chân ngõ ra, kích thước 7x7 mm.
- Số lượng I/O giao tiếp ngoại vi là 37.
- Kiểu dao động: dùng xung nội, có thể dùng xung ngoại bằng thạch anh.
- Kiểu bộ nhớ: flash, có thể ghi và xóa bằng điện. Không mất dữ liệu khi ngừng cung cấp điện.
- Có thể kết nối với 7 x DMA, điều khiển động cơ servo bằng PWM, động cơ bước và watchdog timer.
- 3 x USARTs, 2x I2C, 4 x Timer, 10 x ADC, 2 x SPI, 1 x CAN, 2 x ADC (12 bit và 16 kênh).
- Hoạt động ở mức điện áp 2 – 3.6v.

2.1.4 Các thành phần KIT STM32F103C8T6.



Hình 2. 3 Các thành phần của KIT STM32F103C8T6

Micro usb: Cấp nguồn 5v cho kit, có mạch chuyển đổi 5v -> 3.3v tích hợp trong kit. Không hỗ trợ tải hay debug.

Boot select:

Boot 1	Boot 0	Chức năng
X	0	Sử dụng bộ nhớ flash.
0	1	Khởi động nạp chương trình qua USART1 vào bộ nhớ hệ thống.
1	1	Sử dụng bộ nhớ SRAM.

Bảng 2. 3 Chức năng lựa chọn Boot trong kit

Khi boot0 = 0, boot1 = 0 hoặc để hở thì KIT dùng bộ nhớ flash để sử dụng.

Khi boot0 = 1, boot1 = 0 thì KIT dùng bộ nhớ hệ thống để nạp code qua USART1.

Khi boot0 = 1, boot1 = 1 thì chương trình được thực hiện trong SRAM.

Power light: Đèn báo nguồn, có thể xem nguồn điện có ổn định hay không.

Debug interface: hỗ trợ mô phỏng, nạp code và gỡ lỗi.

PC13 Light: đèn kết nối với chân C13, tích cực mức cao. Dùng để kiểm tra I/O hoặc thử nghiệm chương trình.

Main Crystal: thạch anh 8 Mhz, cấp xung chính cho KIT tối đa là 72Mhz.

MCU: bộ xử lý trung tâm – STM32F103C8T6.

Reset button: khởi động lại MCU với chương trình đang chạy.

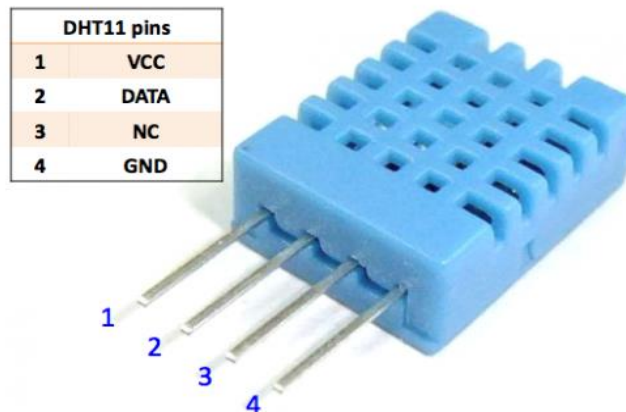
40 chân giao tiếp: bao gồm I/O, nguồn, reset,...

2.2 DHT11.

2.2.1 Giới thiệu.

DHT11 là cảm biến nhiệt độ và độ ẩm.

Là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp onewire (giao tiếp 1 dây truyền dữ liệu duy nhất). Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp có được dữ liệu chính xác mà không cần phải qua bất kỳ tính toán phức tạp nào.



Hình 2. 4 Hình ảnh thực tế và sơ đồ chân.

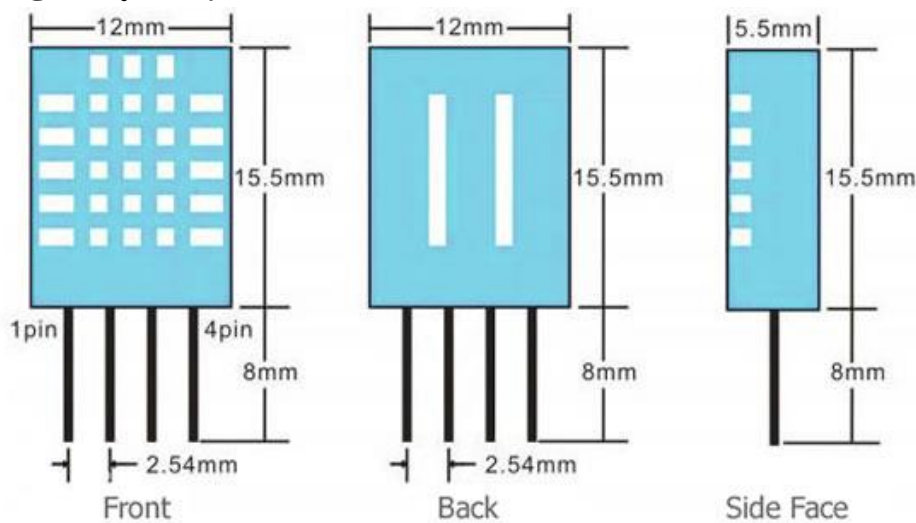
Chân 1 – VCC: cấp nguồn 3-5v.

Chân 2 – Data: lấy dữ liệu kiểu tín hiệu số: 0, 1.

Chân 3 – NC: Not Connect, không được kết nối, nhà thiết kế thêm vào để đúng chuẩn 2.54 mm, và có thể sẽ chắc chắn hơn là 3 chân.

Chân 4 – GND: nối đất.

2.2.2 Thông số kỹ thuật.



Hình 2. 5 Kích thước và khoảng cách chân DHT11

Nguồn cung cấp: 3-5v DC.

Dòng sử dụng: 2.5mA.

Hoạt động tốt ở độ ẩm 20 - 80% với sai số 5%.

Hoạt động tốt ở nhiệt độ 0 tới 50 °C với sai số $\pm 2^{\circ}\text{C}$.

Kích thước 15mm x 12mm x 5.5 mm

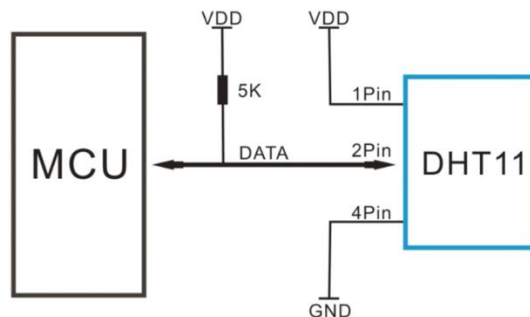
Có 4 chân. Khoảng cách chân 2.54 mm.

2.2.3 Nguyên lý hoạt động.

Để có thể giao tiếp với DHT11 theo chuẩn onewire, vi xử lý cần thực hiện theo 2 bước như sau:

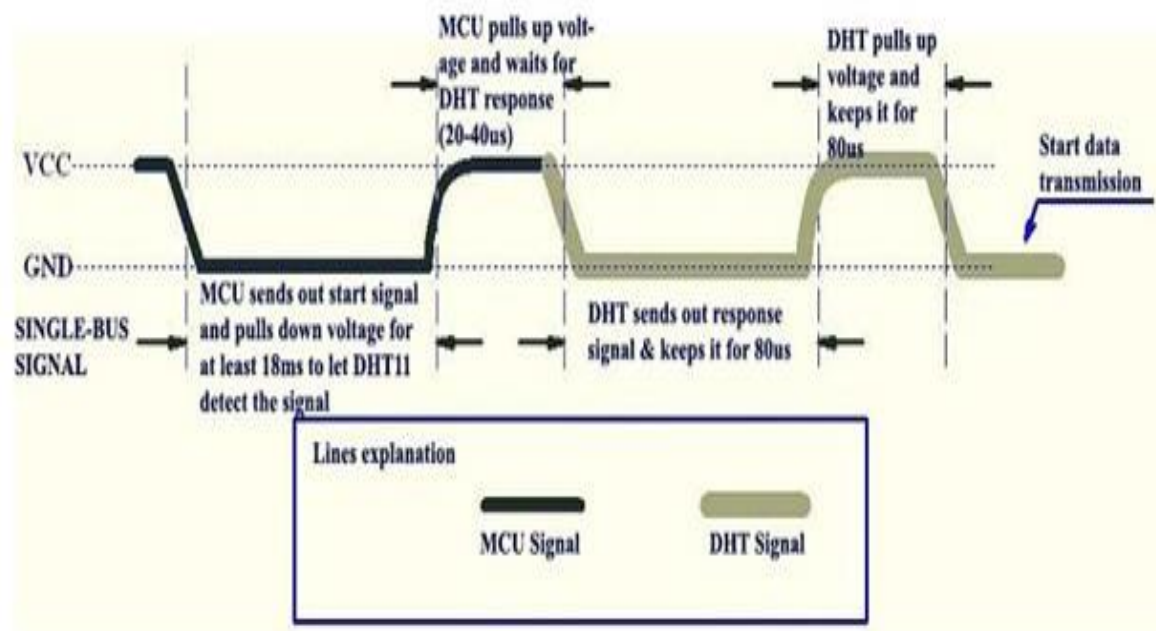
Gửi tín hiệu muốn đo (tín hiệu start) tới DHT11, sau đó DHT11 xác nhận lại.

Khi đã giao tiếp được với DHT11, cảm biến sẽ gửi lại 5 byte dữ liệu bao gồm nhiệt độ, độ ẩm đo được và checksum để kiểm tra.



Hình 2. 6 Kết nối giữa MCU và DHT11

Bước 1: gửi tín hiệu start.



Hình 2. 7 Gửi tín hiệu

MCU thiết lập chân DATA của DHT11 là Output. Sau đó kéo chân DATA xuống mức 0 trong khoảng thời gian lớn hơn 18ms (code để 25ms). Khi đó DHT11 sẽ hiểu MCU muốn đo giá trị nhiệt độ và độ ẩm.

MCU đưa chân DATA lên mức 1, và thiết lập lại chân DATA là chân đầu vào.

Sau khoảng thời gian từ 20-40us, DHT11 sẽ kéo chân DATA xuống mức 0. Nếu lớn hơn 40us mà chân DATA không được kéo xuống thấp nghĩa là không giao tiếp được với DHT11.

Nếu giao tiếp được với DHT11 thì 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 không. Nếu tín hiệu đo được DHT11 kéo lên mức cao, khi đó hoàn thiện quá trình giao tiếp của MCU với DHT11.

Bước 2: đọc giá trị trên DHT11.

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.

DHT11 trả giá trị nhiệt độ và độ ẩm về dưới dạng 5 byte (mỗi byte 8bit). Trong đó:

Byte 1: giá trị phần nguyên của độ ẩm (%).

Byte 2: giá trị phần thập phân của độ ẩm (%).

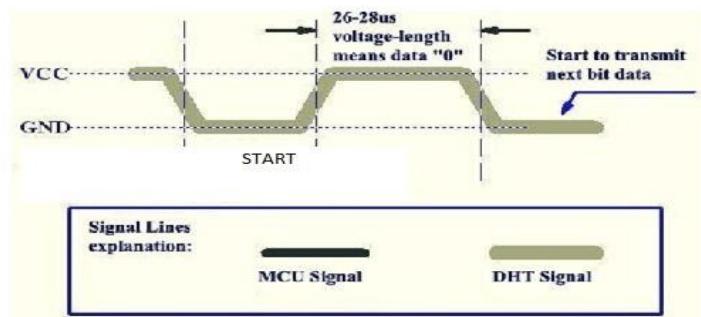
Byte 3: giá trị phần nguyên của nhiệt độ (°C).

Byte 4 : giá trị phần thập phân của nhiệt độ (°C).

Byte 5 : tổng của byte1, byte2, byte3, byte4.

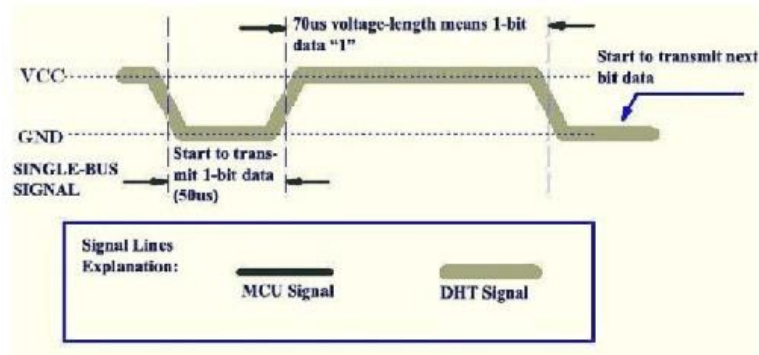
Nếu Byte 5 = (Byte1 +Byte2 +Byte3 + Byte4) thì giá trị độ ẩm và nhiệt độ là chính xác, nếu sai thì kết quả đo không có nghĩa.

Quy trình nhận dữ liệu bit 0:



Hình 2. 8 Nhận dữ liệu bit ‘0’

Quy trình nhận dữ liệu bit 1:



Hình 2. 9 Nhận dữ liệu bit “1”

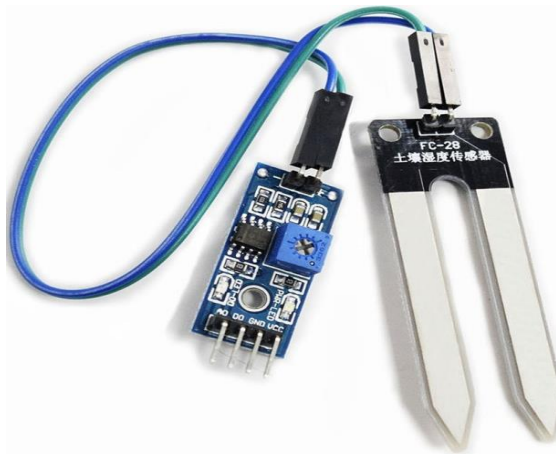
Sau khi tín hiệu được đưa về 0. Dãy chân DATA của MCU được DHT11 kéo lên 1. Nếu chân DATA là 1 trong khoảng 26-28 us thì có giá trị là 0 như *hình 2.8*, còn nếu tồn tại 70us thì có giá trị là 1 như *hình 2.9*.

Do đó trong lập trình, nhóm nghiên cứu bắt sườn lên của chân DATA (bắt đầu kéo lên mức 1), 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 sẽ có kết quả mong muốn.

2.3 Module Cảm biến độ ẩm đất.

2.3.1 Giới thiệu.



Hình 2. 10 Hình ảnh thực tế của module.

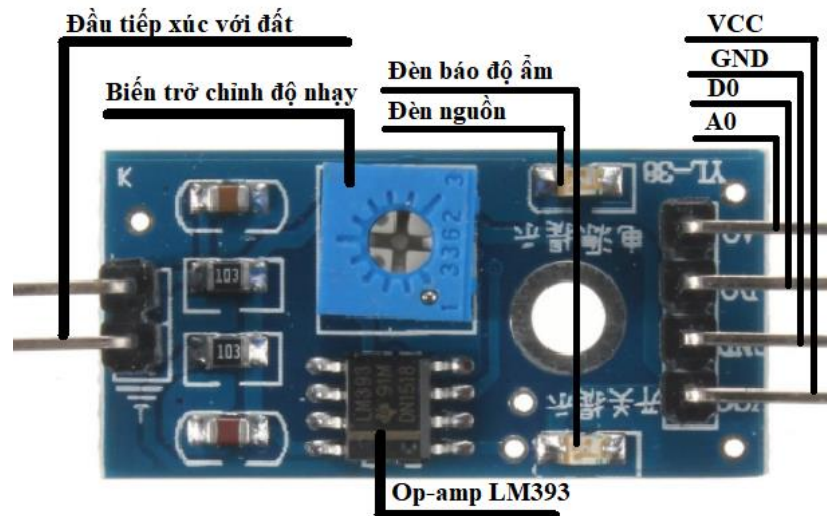
Đây là cảm biến độ ẩm đơn giản, được sử dụng để cảm nhận độ ẩm của đất thông qua các ion có trong nước.

Nó hoạt động dựa vào các ion dẫn điện trong nước để chuyển đổi về mức độ nước có trong đất. Vì vậy số liệu chỉ là ở mức tương đối.

Bình thường đầu ra ở mức thấp, khi đất thiếu nước thì đầu ra sẽ mức cao, độ nhạy có thể điều chỉnh được bằng biến trở.

Mạch có thể sử dụng tưới cây tự động khi không có người quản lý khu vườn trong gia đình hoặc tự động tưới cây khi đất có dấu hiệu thiếu nước mà nhóm đang nghiên cứu.

2.3.2 Thông số kỹ thuật.



Hình 2. 11 Module chính của cảm biến độ ẩm đất

Kích thước module: 1.6cm x 3cm.

Các chân giao tiếp của module:

VCC: 3.3 – 5v.

GND: Nối đất.

D0: Đầu ra tín hiệu số: 0 và 1.

A0: Đầu ra analog (tín hiệu tương tự).

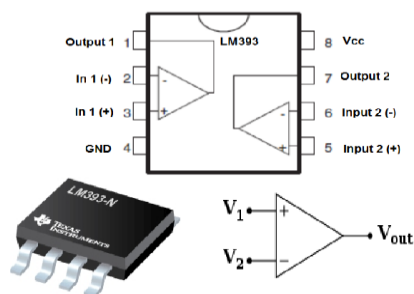
Đầu tiếp xúc đất: kết nối với đầu cắm vào đất để phát hiện độ ẩm của đất. Khi độ ẩm của đất đạt ngưỡng thiết lập, sẽ có tín hiệu ngõ ra tương ứng ở D0 và A0.

Led báo độ ẩm: Sáng khi độ ẩm tại đầu cắm đạt đến ngưỡng thiết lập.

Led báo nguồn: Báo nguồn cung cấp đầu vào.

Biến trở chỉnh độ nhạy: thiết lập độ nhạy của module bằng việc chỉnh biến trở này, làm tăng độ nhạy của đầu cắm với nước trong đất.

Op-amp LM393:



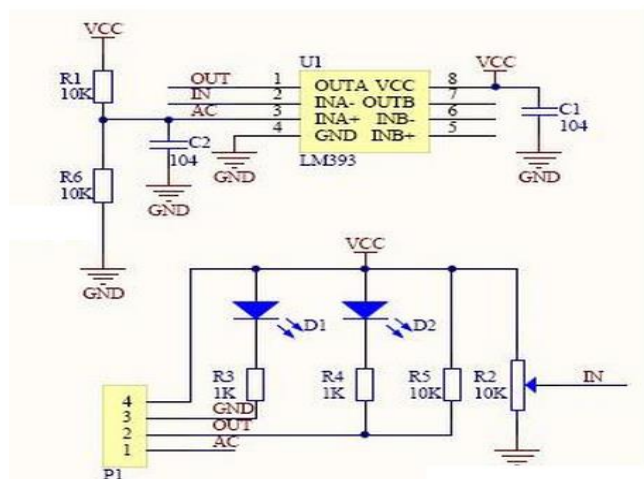
Hình 2. 12 Hình ảnh thực thể và sơ đồ chân LM393

LM393 là vi mạch gồm hai bộ so sánh hoạt động độc lập, hoạt động với cả nguồn cấp đơn hoặc hai nguồn đối xứng. Trong module cảm biến độ ẩm đất thì chỉ sử dụng một mạch so sánh.

Theo biểu diễn trong ký hiệu mạch so sánh, với V_1 ở ngõ vào thuận, thì

- Nếu $V_1 > V_2$, V_{out} là logic 1 (high).
- Nếu $V_1 < V_2$, V_{out} là logic 0 (low).

2.3.3 Nguyên lý hoạt động.



Hình 2. 13 Sơ đồ nguyên lý của module cảm biến

Theo sơ đồ mạch nguyên lý: khi module cảm biến độ ẩm được kích hoạt, khi đó sẽ có sự thay đổi điện áp tại đầu vào của IC LM393. IC này nhận biết có sự thay đổi nó sẽ đưa ra một tín hiệu thấp để báo hiệu và thay đổi sẽ được tính toán để đọc độ ẩm đất.

Đầu cảm của cảm biến được kết nối với “INA-” và “GND”, LM393 sẽ so sánh điện áp của “INA-” và “INA+” để đưa mức điện áp ngõ ra tại “OUT”.

Để việc đo độ ẩm được chính xác, nhóm nghiên cứu chỉ dùng ngõ ra A0, không dùng đến ngõ ra D0.

2.4 LCD 16x02.

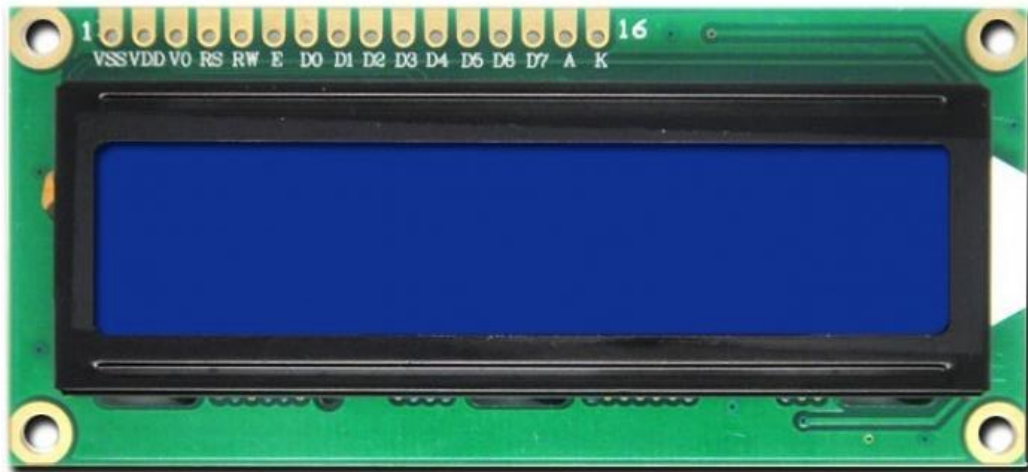
2.4.1 Giới thiệu.

Ngày nay, thiết bị hiển thị LCD (Liquid Crystal Display) được sử dụng trong rất nhiều các 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 và kí tự đồ họa), dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn rất ít tài nguyên hệ thống, giá thành rẻ và dễ giao tiếp, ...

2.4.2 Hình dáng và kích thước.

Có rất nhiều loại LCD với nhiều hình dáng và kích thước khác nhau (20x04, 16x02,...).

Dưới đây, nhóm nghiên cứu xét đến LCD 16x02 và cũng dùng LCD này vào đề tài nghiên cứu của nhóm



Hình 2. 14 Hình dáng của loại LCD 16x02

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, dễ dàng sử dụng.

Màn hình được gắn trên board mạch kết hợp với LED nền.

2.4.3 Sơ đồ chân LCD 16x02

LCD 16x02 có 2 loại là 14 chân và 16 chân, sự khác nhau là các chân nguồn cấp cho đèn nền còn các chân điều khiển thì không thay đổi.

Ở đây, nhóm nghiên cứu xét loại LCD 16 chân bao gồm:

5 chân nguồn: Vss, Vdd, VO, A, K.

3 chân điều khiển: RS, RW, E.

8 chân gửi dữ liệu: D0 đến D7.

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 V _{CC} =5V của mạch điều khiển.
3	V ₀	Điều chỉnh độ tương phản của LCD. Thường nối với biến trở 10k.
4	RS	Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (V _{CC}) để chọn thanh ghi. <ul style="list-style-type: none"> • ‘0’: Lệnh. • ‘1’: Dữ liệu.
5	R/W	Chân chọn chế độ đọc/ghi (Read/Write). <ul style="list-style-type: none"> • Nối R/W với logic “0” để ghi dữ liệu vào LCD. • Hoặc nối với logic “1” để đọc dữ liệu từ LCD.
6	E	Chân cho phép (Enable). Các lệnh được chấp nhận khi có 1 xung cho phép của chân E. <ul style="list-style-type: none"> • Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chấp nhận khi có 1 xung (mức ‘1’ xuống ‘0’) ở chân E. • Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi có 1 xung (mức ‘0’ lên ‘1’) ở chân E và được LCD giữ ở bus đến khi chân E xuống 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 : <ul style="list-style-type: none"> • 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	A	Anot (+) của đèn nền trong LCD
16	K	Katot (-) của đèn nền trong LCD

Bảng 2. 4 Chức năng các chân của LCD 16x02

2.4.4 Địa chỉ từng ký tự trên LCD 16x02.

Địa chỉ	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
Địa chỉ	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

Bảng 2. 5 Địa chỉ trên LCD 16x02

LCD 16x02 có 2 hàng, mỗi hàng 16 ký tự .

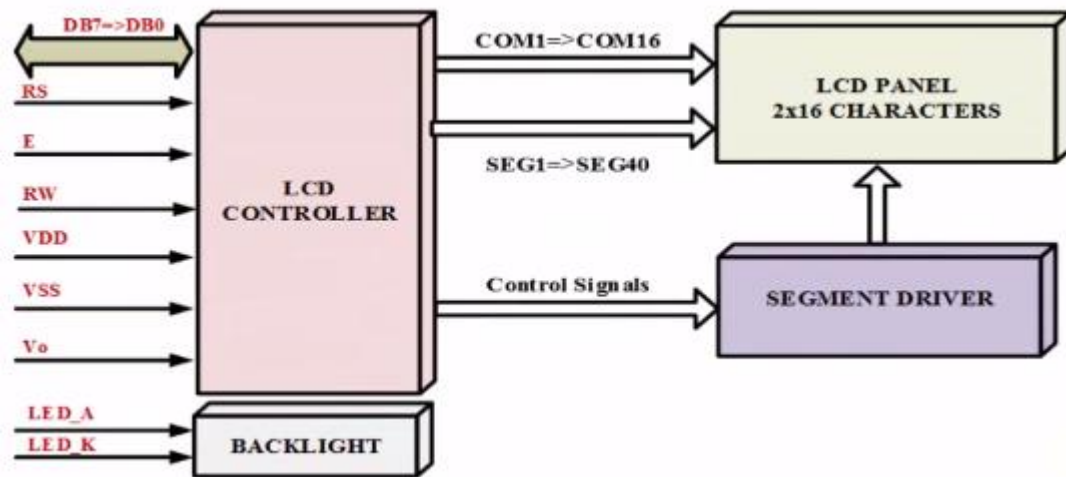
Hàng 1: ký tự tận cùng bên trái có địa chỉ: 0x80, ký tự kế là 0x01, cuối cùng là 0x8F.

Hàng 2: ký tự tận cùng bên trái có địa chỉ: 0xC0, ký tự kế: 0xC1, cuối cùng là 0xCF.

2.4.5 Bộ điều khiển LCD và các vùng nhớ.

Để điều khiển LCD thì có các IC chuyên dụng được tích hợp bên dưới LCD có mã số 447801 đến các IC 447809.

Sơ đồ khối của bộ điều khiển LCD như hình sau:



Hình 2. 15 Sơ đồ khối của bộ điều khiển LCD

Sơ đồ gồm 4 phần:

LCD controller: bộ điều khiển LCD.

LCD panel 2x16 characters: bảng ký tự lưu sẵn trong LCD.

Segment driver: bộ thúc tín hiệu các đoạn trong LCD

Back light: đèn nền.

Bộ điều khiển của LCD có 3 vùng nhớ nội, mỗi vùng có chức năng riêng biệt.

Bộ nhớ DDRAM.

Bộ nhớ chứa dữ liệu để hiển thị (Display Data RAM:DDRAM) lưu trữ những mã ký tự để hiển thị lên màn hình. Mã ký tự lưu trữ trong vùng DDRAM sẽ tham chiếu với từng bitmap ký tự được lưu trữ trong CGROM đã được định nghĩa trước hoặc đặt trong vùng do người sử dụng định nghĩa.

Bộ phát ký tự ROM – CGRAM.

Bộ phát ký tự ROM (Character Generator ROM: CGROM) chứa các kiểu bitmap cho mỗi ký tự được định nghĩa trước mà LCD có thể hiển thị, như được trình bày bằng mã ASCII.

Mã ký tự lưu trong DDRAM cho mỗi vùng ký tự sẽ được tham chiếu đến 1 vị trí trong CGROM.

Bộ phát ký tự RAM – CGRAM.

Bộ phát ký tự RAM (Character Generator RAM: CGRAM) cung cấp vùng nhớ để tạo ra 8 ký tự tùy ý. Mỗi ký tự gồm 5 cột và 8 hàng. Đây là phần sẽ được nhóm nghiên cứu dùng để hiển thị chữ “DA – 1” kích thước lớn, tự tạo, sẽ được nhóm trình bày ở phần sau.

2.4.6 Mã ASCII hiển thị LCD 16x02.

Lower 4 Bits	Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	1	P	`	P				-	9	3	α	p	
xxxx0001	(2)		!	1	A	Q	a	4				。	ア	チ	△	ã	q
xxxx0010	(3)		"	2	B	R	b	r				「	イ	ツ	×	β	θ
xxxx0011	(4)		#	3	C	S	c	s				」	ウ	テ	モ	ε	∞
xxxx0100	(5)		\$	4	D	T	d	t				、	エ	ト	ホ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u				・	オ	ナ	ユ	℃	Ü
xxxx0110	(7)		&	6	F	V	f	v				ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w				ア	キ	ヌ	ラ	g	π
xxxx1000	(1)		(8	H	X	h	x				ィ	ク	ネ	リ	フ	×
xxxx1001	(2))	9	I	Y	i	y				ッ	ケ	ノ	ル	´	¥
xxxx1010	(3)		*	:	J	Z	j	z				エ	コ	ハ	レ	j	〒
xxxx1011	(4)		+	;	K	[k	[オ	サ	ヒ	ロ	×	厶
xxxx1100	(5)		,	<	L	¥	l					ハ	シ	フ	ワ	Φ	⊞
xxxx1101	(6)		-	=	M]	m]				ユ	ズ	ヘ	ン	も	÷
xxxx1110	(7)		.	>	N	^	n	→				ヨ	セ	ホ	°	ñ	
xxxx1111	(8)		/	?	O	_	o	+				ッ	ソ	マ	°	ö	■

Bảng 2. 6 Bảng mã ASCII

LCD sử dụng mã ASCII để hiển thị các ký tự, các số, các ký hiệu,... có tổng cộng 256 ký tự. Mã ASCII như bảng bên trên.

Để hiển thị mã ASCII lên LCD từ giá trị số thập phân thì ta phải chuyển sang số BCD rồi sau đó cộng thêm với 0x30 mới hiển thị lên LCD.

2.5 Module relay 2 kênh.

2.5.1 Giới thiệu.

Relay là một chuyển mạch hoạt động bằng điện. Dòng điện chạy qua cuộn dây của relay tạo ra một từ trường hút lõi sắt non làm thay đổi công tắc chuyển mạch. Dòng điện qua cuộn dây có thể được bật hoặc tắt vì thế relay có hai vị trí chuyển mạch qua lại.

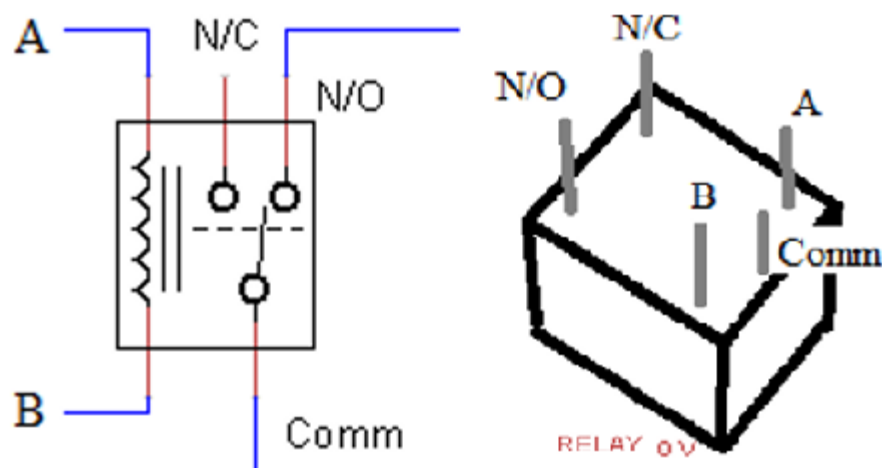


Hình 2. 16 Hình dáng thực của relay

Relay có nhiều loại và nhiều thông số lựa chọn, rất phổ biến trên thị trường.

Nó được sử dụng rất nhiều trong cuộc sống và trong các thiết bị điện tử.

2.5.2 Cấu tạo relay.



Hình 2. 17 Cấu tạo của relay và sơ đồ chân

Cấu tạo relay: gồm 2 phần:

Cuộn hút: tạo ra năng lượng từ trường để hút tiếp điểm về phía mình. Tùy vào điện áp làm việc người ta chia Relay ra:

- DC: 5V, 12V, 24V.
- AC: 110V, 220V.

Cặp tiếp điểm: khi không có từ trường (không cấp điện cho cuộn dây). Tiếp điểm 1 được tiếp xúc với 2 nhờ lực của lò xo. Tiếp điểm thường đóng. Khi có năng lượng từ trường thì tiếp điểm 1 bị hút chuyển sang 3.

Trong Relay có thể có 1 cặp tiếp điểm, 2 cặp tiếp điểm hoặc nhiều hơn.

Sơ đồ chân của relay:

2 chân A,B: cấp nguồn nuôi relay, ở đây nhóm nghiên cứu xét đến relay 5v.

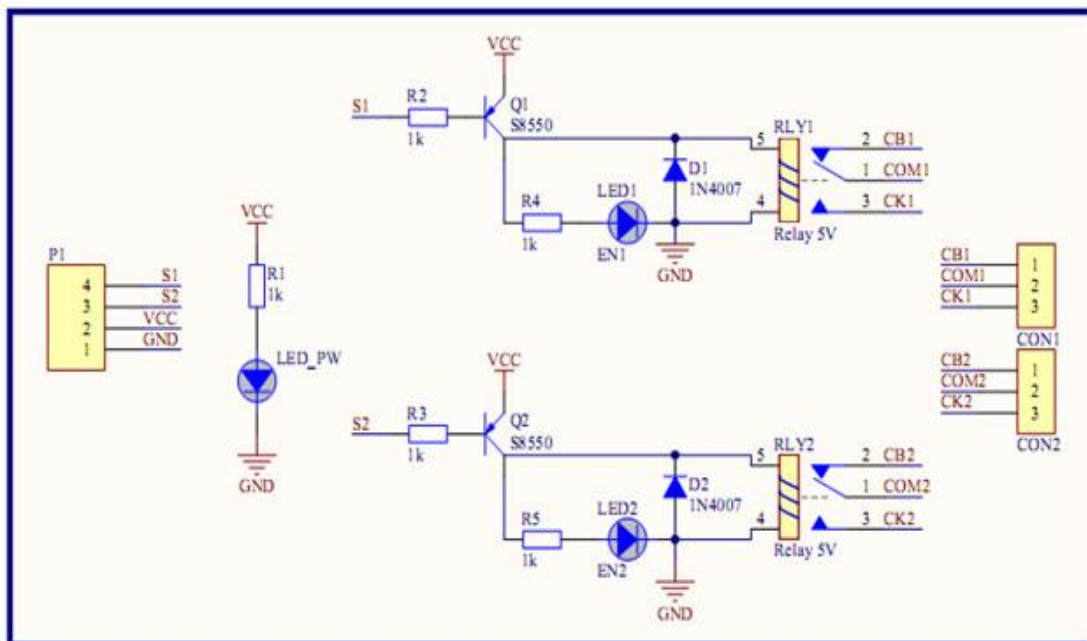
Chân Comm: là chân chung, là nơi kết nối đường cấp nguồn chờ.

Chân N/O, N/C là 2 chân chuyển đổi.

Trong đó: N/C là điểm thường đóng, chân Comm được kết nối với NC khi cuộn dây relay không nhiễm từ (khi 2 đầu cuộn dây không được cấp điện).

N/O: là điểm thường mở, Comm được kết nối với N/O khi cuộn dây relay được từ hóa (được cấp điện).

2.5.3 Sơ đồ nguyên lý module relay 2 kênh.



Hình 2. 18 Sơ đồ nguyên lý module relay 2 kênh

Trong mạch trên là sơ đồ của 2 cặp Relay 5V. Ta phân tích hoạt động của Relay 1. (Sử dụng Transistor PNP S8550).

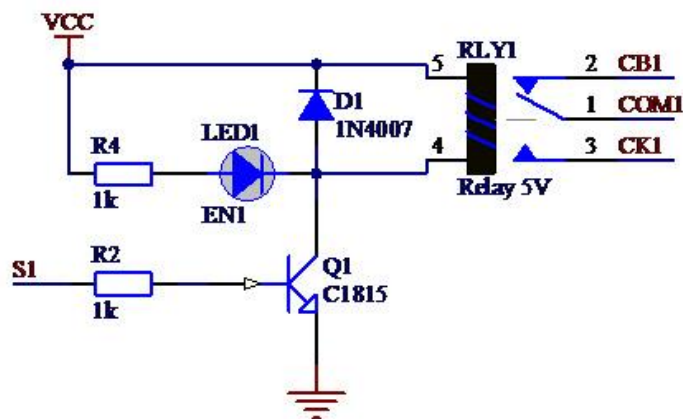
Đầu tiên, tín hiệu S1 từ MCU gửi tới.

Khi $S1 = 1$; Q1 khóa lại, không có dòng chạy qua cuộn hút Relay1. Đèn LED1 tắt. Tiếp điểm 1 với 2 nối với 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 (tắt).

Trong sơ đồ trên nếu sử dụng Relay 12V, 24V thì cần thêm 1 mạch đệm. Do tín hiệu điều khiển 5V, không cùng điện áp với điện áp cấp cho Relay (12V, 24V).



Hình 2. 19 Module relay dùng BJT PNP

Ngoài sử dụng transistor PNP điều khiển, ta có thể sử dụng Transistor NPN (C1815) để điều khiển.

Khi $S1 = 1$; Q1 thông, có dòng điện cấp cho cuộn hút của Relay. Tiếp điểm 1 nối với 3, đèn LED1 sáng.

Khi $S1 = 0$; Q1 tắt, không có dòng qua cuộn hút. Tiếp điểm 1 nối với 2, đèn LED1 tắt.

Với sơ đồ này, có thể thay Relay 5V bằng 12V, 24V (Tương ứng $VCC = 12V, 24V$). Mặc dù điện áp điều khiển $S1 = 0V$ hoặc 5V. Mạch vẫn hoạt động bình thường.

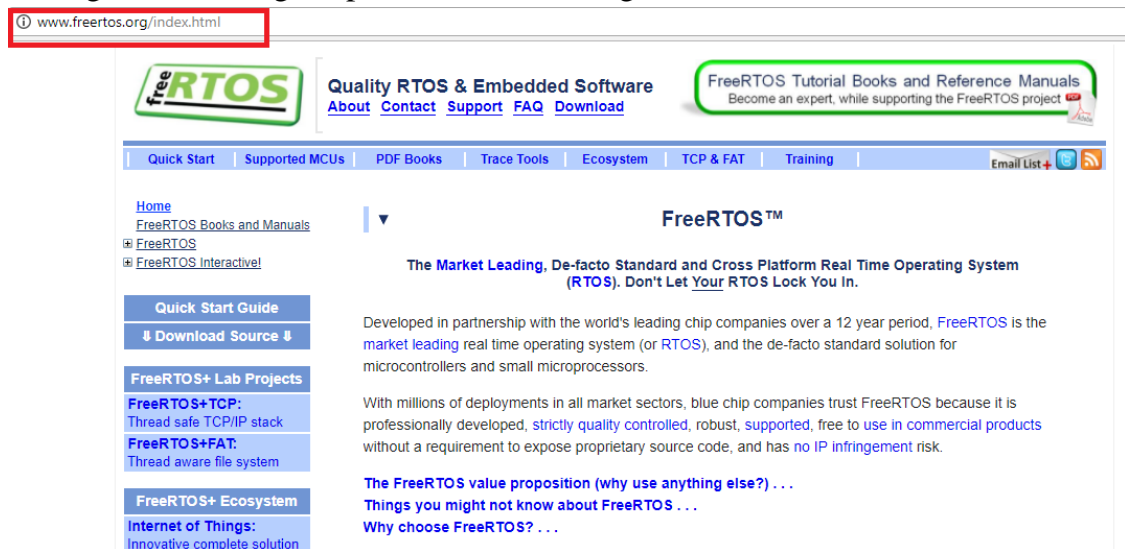
2.6 Hệ điều hành thời gian thực RTOS.

2.6.1 Giới thiệu.

FreeRTOS là một hệ điều hành nhúng thời gian thực (Real Time Operating System) và là cấu trúc mã nguồn mở được phát triển bởi Real Time Engineers Ltd, sáng lập và sở hữu bởi Richard Barry.

Được hỗ trợ rất nhiều dòng vi điều khiển trong đó có dòng STM32. FreeRTOS là một hệ thống nhỏ và đơn giản, hạt nhân bao gồm 3 hoặc 4 tập tin C. Việc ứng dụng một hệ điều hành trong hệ thống sẽ giảm thiểu thời gian code, giảm một số gánh nặng trong giải thuật cho một hệ thống lớn.

Trang chủ hệ thống: <http://www.freertos.org>



Hình 2. 20 Giao diện trang chủ FreeRTOS

FreeRTOS được thiết kế phù hợp cho nhiều hệ thống nhúng nhỏ gọn vì nó chỉ triển khai rất ít các chức năng như: cơ chế quản lý bộ nhớ và tác vụ cơ bản, các hàm API quan trọng cho cơ chế đồng bộ.

Nó không cung cấp sẵn các giao tiếp mạng, drivers hay hệ thống quản lý tệp như những hệ điều hành nhúng cao cấp khác.

Tuy vậy, FreeRTOS có nhiều ưu điểm, hỗ trợ kiến trúc vi điều khiển khác nhau, ngoài ra còn cho phép triển khai các cơ chế điều độ giữa các tiến trình như: queues, counting semaphore,...

2.6.2 Tải thư viện hỗ trợ RTOS.

Tải thư viện RTOS tại link: <http://www.freertos.org/>

Click “Download”.

Click “Download Source Code and Projects”.



Hình 2. 21 Nguồn tải thư viện RTOS

Sau khi tải về và chạy extract file “FreeRTOSv9.0.0.exe”. Ta được:

Local Disk (C:) > Users > NVGIOI > Downloads > Programs > FreeRTOSv9.0.0				
Name	Date modified	Type	Size	
FreeRTOS	5/20/2016 7:25 PM	File folder		
FreeRTOS-Plus	5/20/2016 7:20 PM	File folder		
New - Direct to Task Notifications	1/16/2015 5:56 PM	Internet Shortcut	1 KB	
New - FreeRTOS+ TCP	12/22/2014 2:03 AM	Internet Shortcut	1 KB	
Quick_Start_Guide	9/17/2013 3:17 PM	Internet Shortcut	1 KB	
readme.txt	9/17/2013 3:20 PM	TXT File	2 KB	
Upgrading-to-FreeRTOS-9	2/19/2016 4:32 PM	Internet Shortcut	1 KB	

Hình 2. 22 “FreeRTOSv9.0.0.exe” sau khi extract

Ở đây ta quan tâm đến 2 thư mục “FreeRTOS” và “FreeRTOS-Plus”.

Trong đó:

“FreeRTOS”: chứa các file cấu hình cho RTOS cơ bản. Nhóm nghiên cứu chỉ dùng và áp dụng trên gói này.

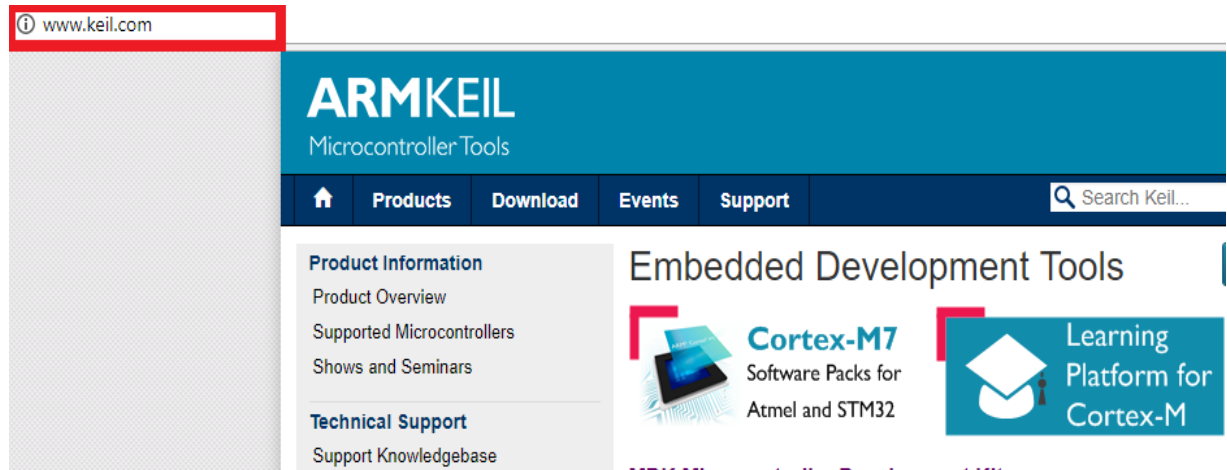
“FreeRTOS-Plus: chứa các file cấu hình cho RTOS nâng cao như: truyền UDP/TCP, SSL, Nabto,...

2.7 Trình biên dịch Keil C cho ARM.

2.7.1 Lý do lựa chọn.

Sự ra đời của một loại vi điều khiển gắn liền với việc phát triển phần mềm ứng dụng cho việc lập trình con vi điều khiển đó. Kèm theo đó là sự xuất hiện của nhiều chương trình soạn thảo và biên dịch ngôn ngữ C cho vi điều khiển: Keil C, HT-PIC, MikoC, CCS...

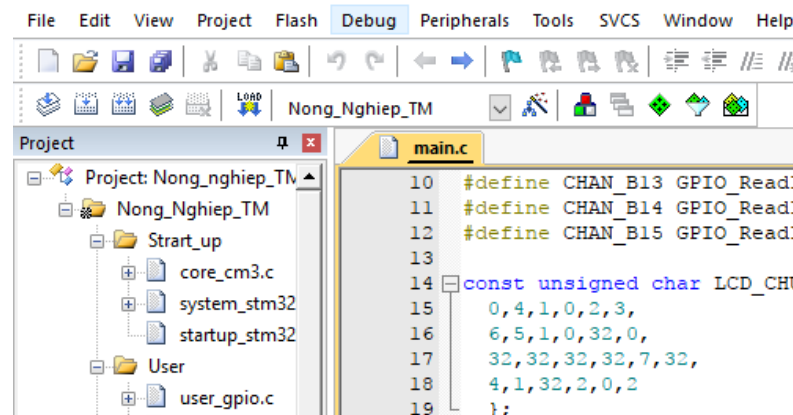
Và Keil C là phần mềm lập trình mạnh cho vi điều khiển ARM.



Hình 2. 23 Trang chủ để tải Keil C.

Việc tải phần mềm Keil C cho ARM là hoàn toàn miễn phí, bên cạnh đó còn có thêm các phần mềm hỗ trợ miễn phí khác: gỡ lỗi, hỗ trợ thư viện,...

2.7.2 Giới thiệu về Keil C.



Hình 2. 24 Giao diện Keil C.

Trình biên dịch Keil C cho ARM được thiết kế đặc biệt để tối ưu hóa phần mềm chạy trên bộ vi xử lý ARM nói chung. Đó là kết quả của 20 năm phát triển cùng với kiến trúc ARM.

Với sự tối ưu hóa cho hiệu suất cao và giảm kích thước mã so với các trình biên dịch khác.

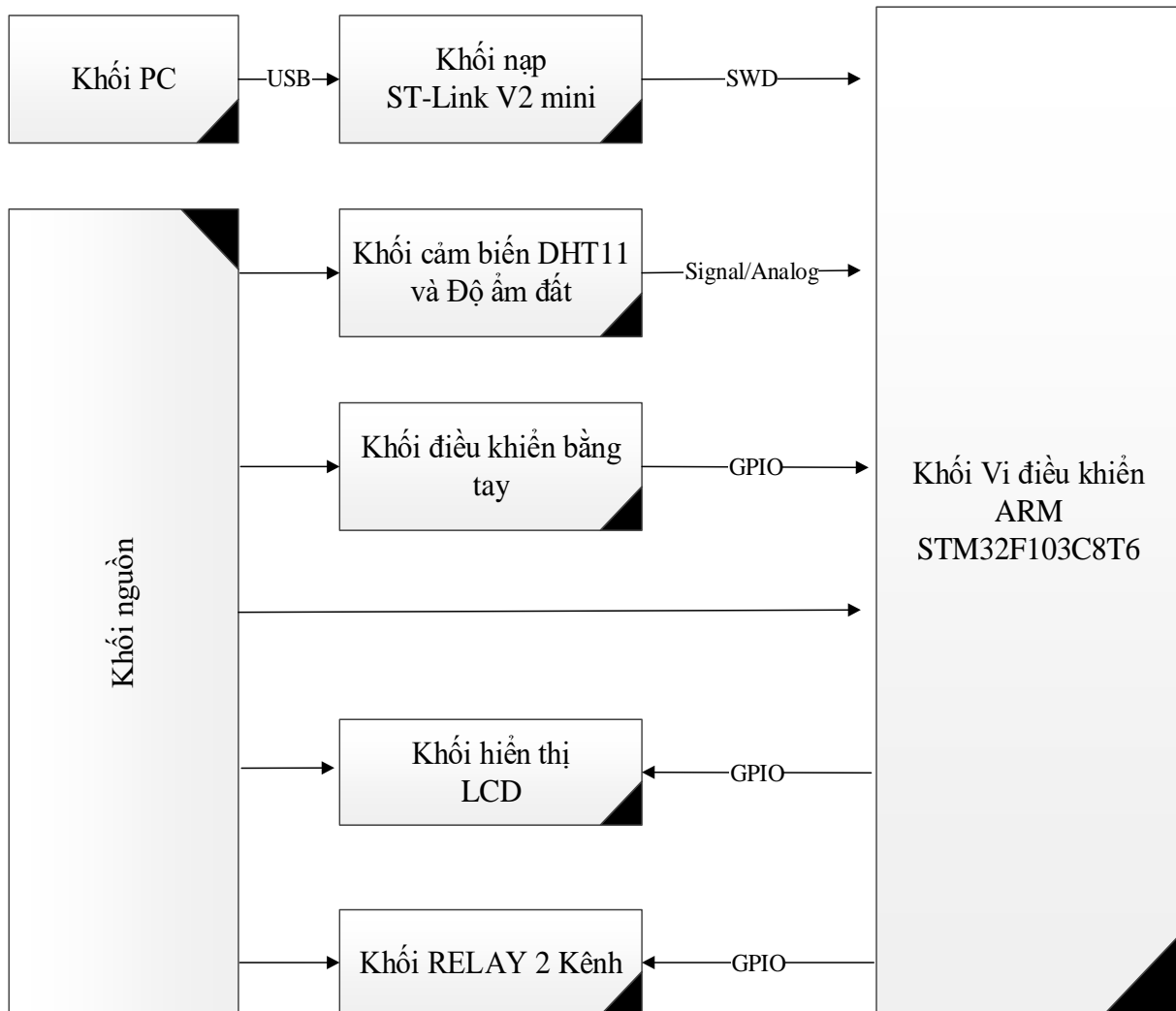
CHƯƠNG 3

PHÂN TÍCH LỰA CHỌN PHẦN CỨNG HỆ THỐNG

3.1 Yêu cầu hệ thống.

Thiết kế một mô hình theo dõi các yếu tố khí hậu của môi trường trồng cây thông qua các cảm biến: nhiệt độ, độ ẩm dht11; cảm biến độ ẩm đất để điều chỉnh sao cho phù hợp, thuận lợi cho sự phát triển của cây bằng relay điều khiển đèn gia nhiệt và relay moto. Các thông số đều hiển thị lên LCD 16x02 để người sử dụng tiện theo dõi.

3.2 Sơ đồ khối hệ thống.



Hình 3. 1 Sơ đồ khối tổng quát của hệ thống.

Chức năng từng khối:

Khối Kit ARM STM32F103C8T6: trung tâm tiếp nhận thông tin, xử lý và điều khiển các thiết bị ngoại vi khác.

Khối nguồn: chuyển đổi dòng điện 12v sang 5v và 3.3v để cấp cho các khối khác trong mạch.

Khối PC: sử dụng phần mềm Keil C để viết và nạp chương trình vào Kit ARM STM32F103C8T6. Phần cứng sử dụng: USB nạp ST-Link V2 mini dùng để nạp chương trình.

Khối cảm biến:

DHT11 thì thu nhận tín hiệu về nhiệt độ, độ ẩm từ môi trường bên ngoài truyền nhận theo chuẩn giao tiếp onewire vào Kit ARM.

Cảm biến độ ẩm đất: thu nhận mức độ nước có trong đất trả về tín hiệu số và tín hiệu tương tự. Ở đây nhóm nghiên cứu chỉ sử dụng tín hiệu tương tự để số liệu chính xác hơn.

Khối điều khiển bằng tay: gồm 3 nút nhấn.

Nút nhấn 1 dùng để chuyển đổi giữa chế độ điều khiển bằng tay hoặc chế độ tự động trong hệ thống.

Hai nút nhấn còn lại dùng để điều chỉnh bằng tay relay 2 kênh để điều khiển đèn gia nhiệt và bơm nước.

Khối hiển thị LCD: dùng để hiển thị các thông số nhiệt độ, độ ẩm, độ ẩm đất và các trạng thái hoạt động của hệ thống.

Khối Relay 2 kênh: dùng để điều khiển đóng ngắt các thiết bị được kết nối theo yêu cầu của người sử dụng hoặc đóng ngắt thông minh dựa trên nhiệt độ, độ ẩm và độ ẩm đất đã được viết trong chương trình hệ thống.

3.3 Phân tích và kết nối phần cứng.

3.3.1 Khối Kit ARM STM32F103C8T6.

STM32F103C8T6 hỗ trợ khá tốt phần cứng với 40 chân và có 32 chân giao tiếp I/O nên khá thoải mái trong việc kết nối giao tiếp với các thiết bị ngoại vi bên ngoài.

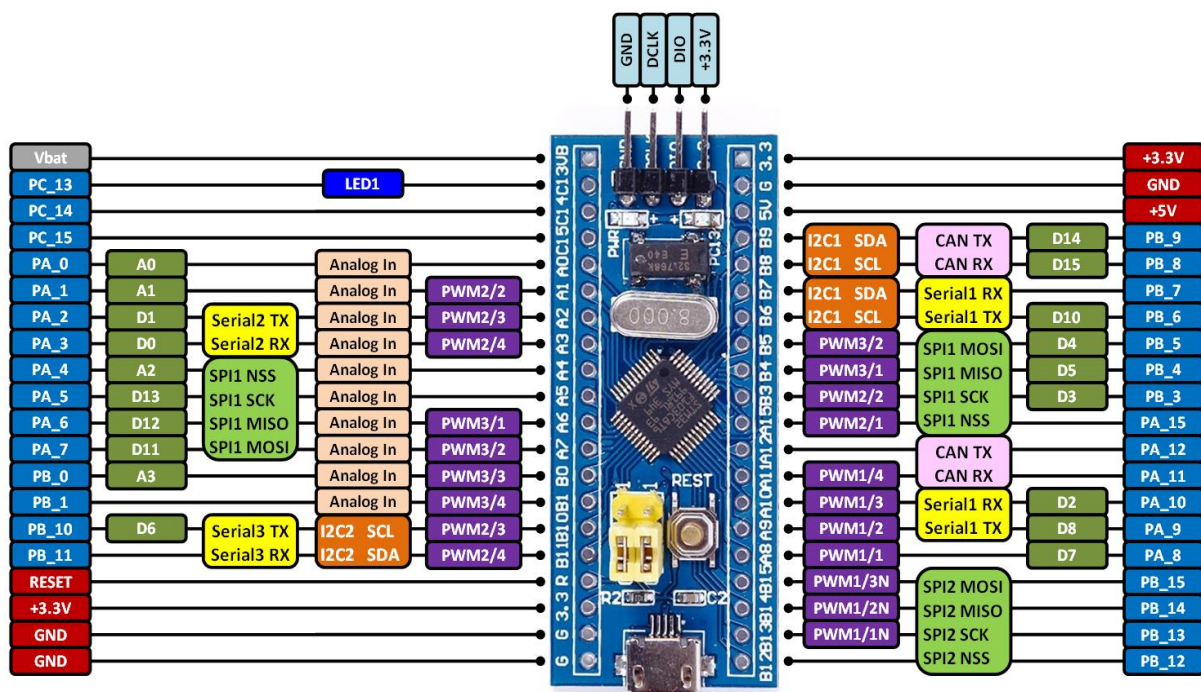
Giá thị trường rất rẻ.

Kích thước nhỏ gọn chỉ 53.34 x 22.86 mm.

Hệ thống thư viện hỗ trợ phong phú.

Dễ dàng cung cấp nguồn 5v qua USB hoặc 3.3v qua cổng nạp.

Hỗ trợ nhiều chuẩn giao tiếp.



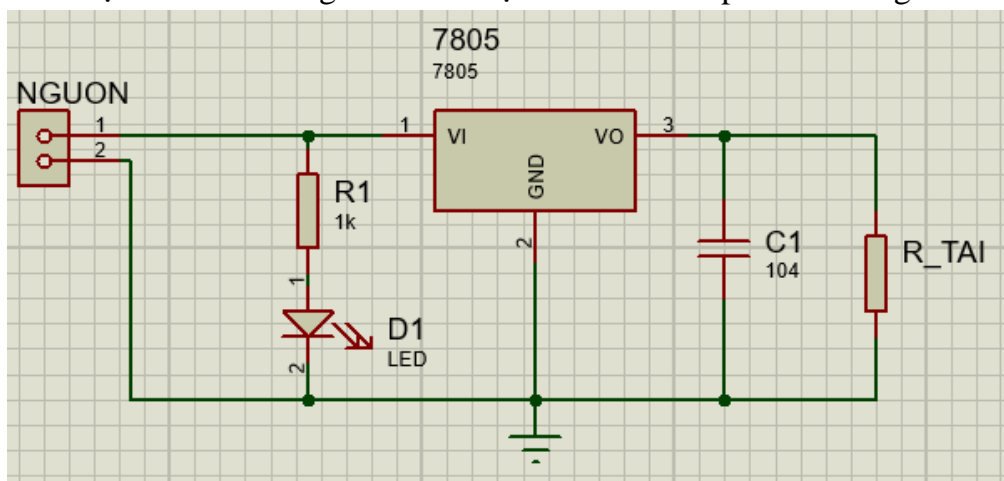
Hình 3. 2 Pinout và chức năng từng chân của KIT STM32F103C8T6.

Sơ đồ chân và chức năng của 40 chân trong KIT. Một chân có thể nhiều chức năng khác nhau, vì thế phải cấu hình đúng với chức năng để kit hoạt động tối ưu nhất.

3.3.2 Khối nguồn.

Trên thị trường có rất nhiều IC ổn áp dương với giá cả hợp lý: 78xx, LM317, LM1805, LM2596..

Tuy nhiên, về yêu cầu, mạch cần nguồn nuôi 5v và 3.3v, dòng yêu cầu nhỏ và nguồn nuôi phải cố định nên nhóm nghiên cứu chọn IC 7805 ổn áp cho khối nguồn của nhóm.



Hình 3. 3 Sơ đồ khối nguồn cung cấp.

Nguồn vào: 9V – 0.5A.

Led báo: Led đực, xanh dương 5 mm.

Điện áp tham chiếu: 3V - 3.2V.

Dòng 5mA – 20 mA.

Tính toán điện trở.

$$R_{\max} = \frac{U_{\text{Nguồn}} - U_{\text{LED}}}{I_{\text{LED}}} = \frac{9-3}{5 \cdot 10^{-3}} = 1200 \, \Omega.$$

$$R_{\min} = \frac{U_{\text{Nguồn}} - U_{\text{LED}}}{I_{\text{LED}}} = \frac{9-3}{20 \cdot 10^{-3}} = 300 \, \Omega.$$

$$R = 300 \sim 1200 \, \Omega.$$

Vì vậy nhóm nghiên cứu chọn trở $R = 1000 \, \Omega = 1 \, \text{K} \, \Omega$.

Ở đây nhóm nghiên cứu không dùng tụ lọc nhiều ở nguồn đầu vào vì nhóm dùng adapter 9V – 0.5A, tụ lọc đã được tích hợp sẵn trong adapter nên nhóm bỏ qua tụ lọc này để mạch được gọn hơn.

IC 7805:

Table 12: Electrical Characteristics Of L7805C (refer to the test circuits, $T_J = 0$ to 125°C , $V_I = 10\text{V}$, $I_O = 500 \, \text{mA}$, $C_I = 0.33 \, \mu\text{F}$, $C_O = 0.1 \, \mu\text{F}$ unless otherwise specified).

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_O	Output Voltage	$T_J = 25^\circ\text{C}$	4.8	5	5.2	V
V_O	Output Voltage	$I_O = 5 \, \text{mA to } 1 \, \text{A}$ $P_O \leq 15\text{W}$ $V_I = 7 \text{ to } 20 \, \text{V}$	4.75	5	5.25	V

Hình 3. 4 Datasheet của LM7805C

Điện áp ngõ vào lý tưởng: 7V – 20V.

Điện áp ngõ ra: 5V.

Điện áp thực ngõ ra dao động từ 4.75V ~ 5.25V.

Tụ lọc $C_O = 0.1 \, \mu\text{F}$. Nên dùng tụ gốm 104 cho mạch.

Ở đây nhóm nghiên cứu không dùng tụ lọc nhiều ở nguồn đầu vào vì nhóm dùng adapter 9V – 0.5A, tụ lọc đã được tích hợp sẵn trong adapter nên nhóm bỏ qua tụ lọc này để mạch được gọn hơn.

Tu C_I : tụ gốm không phân cực 104.

Lọc nhiễu răng cưa, nhọn, hài cao gây nguy hiểm cho mạch.

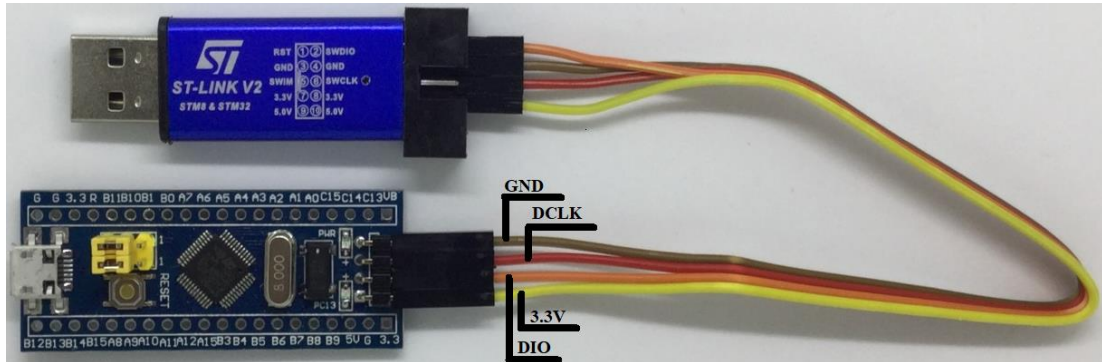
Giá trị: $C = 10 \cdot 10^4 \, \text{pF} = 100000 \, \text{pF} = 100 \, \text{nF} = 0.1 \, \mu\text{F}$.

3.3.3 Khởi PC.

Kết nối với máy tính để thực hiện việc tạo code thông qua phần mềm Keil C mà nhóm chọn, dễ dàng trong việc sửa lỗi.

Ngoài ra nhóm nghiên cứu còn dùng thêm phần mềm mô phỏng STMStudio để xem xung mô phỏng trên kit, đảm bảo việc hoạt động chính xác.

Khối PC kết nối với kit thông qua mạch nạp ST-Link V2 mini nhỏ gọn với 5 chân ra: SWDIO, GND, 3.3V, 5V, SWCLK.



Hình 3. 5 Kết nối ST-Link V2 mini với KIT.

Kết nối:

KIT STM32F103C8T6	ST-Link V2 mini
DIO	SWDIO
DCLK	SWCLK
-	5v
3.3V	3.3V
GND	GND

Bảng 3. 1 Kết nối giữa mạch nạp và KIT

3.3.4 Khối cảm biến.

DHT11.

Là cảm biến rất thông dụng hiện nay với chi phí rất rẻ và dễ lấy dữ liệu thông qua giao tiếp onewire. Không cần phải tính toán quá phức tạp để lấy được dữ liệu vì trong cảm biến đã tích hợp bộ tiền xử lý để làm việc này.

So với DHT22 thì DHT11 cho khoảng đo và độ chính xác kém hơn. Tuy nhiên với môi trường cây trồng thì không yêu cầu quá khắc khe về nhiệt độ nên có thể chấp nhận được.

DHT11	KIT STM32F103C8T6
VCC	NGUỒN 5V
GND	GND
SIGNAL	PIN_B12

Bảng 3. 2 Kết nối DHT11 với KIT

Tín hiệu ngõ ra DHT11 được cấu hình nối vào chân B12 của KIT. Người dùng có thể thay thế chân khác nếu muốn.

Với thông số kỹ thuật:

Nguồn cung cấp: 3-5v DC.

Dòng sử dụng: 2.5mA.

Hoạt động tốt ở độ ẩm 20 - 80% với sai số 5%.

Hoạt động tốt ở nhiệt độ 0 tới 50 °C với sai số $\pm 2^{\circ}\text{C}$.

Nó phù hợp với yêu cầu mà nhóm nghiên cứu đang tiến hành.

Cảm biến độ ẩm đất.

Nhóm nghiên cứu sử dụng cảm biến độ ẩm của đất.

Điện áp làm việc 3.3V – 5V thì đáp ứng tốt yêu cầu.

Kích thước nhỏ, tiết kiệm không gian thiết kế, thuận tiện trong quá trình di chuyển.

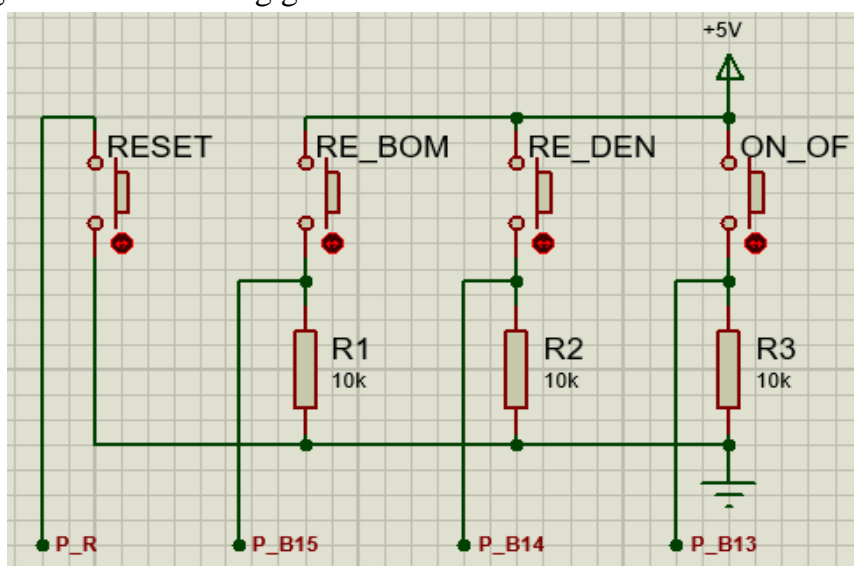
ĐỒ ẨM ĐẤT	KIT STM32F103C8T6
3.3V - 5V	NGUỒN 5V
GND	GND
D0	-
A0	PIN_A0

Bảng 3. 3 Sơ đồ kết nối giữa cảm biến độ ẩm đất và KIT

Nhóm nghiên cứu kết nối như bảng. Ở đầu D0, nhóm không sử dụng nên để trống.

3.3.5 Khởi điều khiển bằng tay.

Sử dụng nút nhấn cơ không giữ để điều khiển.



Hình 3. 6 Kết nối nút nhấn với KIT

Trong mạch này, nhóm nghiên cứu dùng điện trở kéo xuống. Vì:

Khi nhấn nút, chân KIT nhận được mức 1. Thả nút nhấn thì chân KIT sẽ bị thả trôi, nếu một vấn đề nào đó gây nhiễu thì \Rightarrow mạch hoạt động không ổn định. Vì vậy, nhóm thực hiện gắn điện trở 10k để kéo chân KIT xuống GND khi không được nhấn.

Tính toán tìm giá trị điện trở kéo xuống.

Symbol	Ratings	Max.	Unit
I_{VDD}	Total current into V_{DD}/V_{DDA} power lines (source) ⁽¹⁾	150	
I_{VSS}	Total current out of V_{SS} ground lines (sink) ⁽¹⁾	150	
I_{IO}	Output current sunk by any I/O and control pin	25	mA
	Output current source by any I/Os and control pin	-25	
$I_{INJ(PIN)}^{(2)}$	Injected current on five volt tolerant pins ⁽³⁾	-5/+0	
	Injected current on any other pin ⁽⁴⁾	± 5	
$\Sigma I_{INJ(PIN)}$	Total injected current (sum of all I/O and control pins) ⁽⁵⁾	± 25	

Hình 3. 7 Dòng điện tại I/O của KIT

Vì dòng I/O của KIT là 25mA. Ta có: $I = \frac{V}{R} = \frac{5V}{10000\Omega} = 5 \cdot 10^{-4} A = 0.5mA$.

Như vậy, khi nhả nút thì dòng 0.5mA này sẽ đi xuống GND và chân Input sẽ được nối GND, đảm bảo an toàn cho KIT. Giá trị điện trở này trong các mạch điện tử thường là 4k7 hoặc 10k. Có điện trở kéo xuống nhằm loại bỏ hiện tượng trôi nổi điện áp ở ngõ vào. Giá trị điện trở ở đây nhằm tránh hiện tượng ngắn mạch, đảm bảo an toàn cho mạch.

3.3.6 Khởi hiển thị LCD.

Nhóm nghiên cứu sử dụng LCD 16x02 để hiển thị các thông số cho mô hình. Thiết kế nhỏ gọn, giao tiếp dễ dàng, đáp ứng yêu cầu của hệ thống. Có hỗ trợ vùng nhớ tạo ký tự do người dùng định nghĩa.



Hình 3. 8 Ký tự do nhóm nghiên cứu tự tạo tại vùng nhớ CGRAM

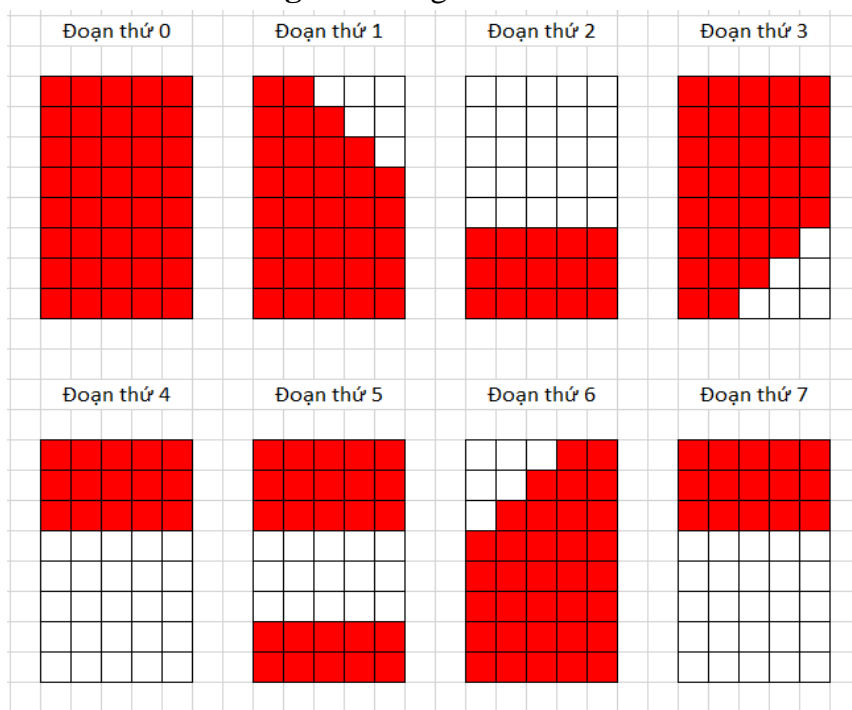
D7	D6	D5	D4	D3	D2	D1	D0	HEX	
0	1	A5	A4	A3	A2	A1	A0		
0	1	0	0	0	0	0	0	0x40	Địa chỉ đầu
0	1		
0	1	1	1	1	1	1	1	0x7F	Địa chỉ cuối

Bảng 3. 4 Thiết lập địa chỉ CGRAM

Vùng nhớ CGRAM này có địa chỉ từ 0x40 đến 0x7F cho phép lưu 64 byte dữ liệu. Có tất cả 8 vùng nhớ từ 0 – 7. Mỗi vùng lưu được 8 byte.

Vùng	ĐC cuối	B7	B6	B5	B4	B3	B2	B1	B0	ĐC đầu	ĐC ký tự
0	0x47	Vùng nhớ 8 byte để lưu ký tự tự tạo								0x40	0
1	0x4F									0x48	1
2	0x57									0x50	2
3	0x5F									0x58	3
4	0x67									0x60	4
5	0x6F									0x68	5
6	0x77									0x70	6
7	0x7F									0x78	7

Bảng 3. 5 Vùng nhớ CGRAM



Hình 3. 9 Các đoạn tạo nên ký tự lớn trên LCD

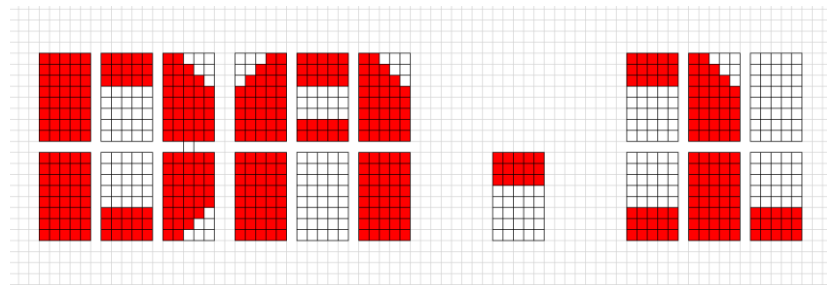
Mỗi một ký tự là một ma trận điểm 5x8, muốn điểm nào sáng thì bit ô đó bằng 1.

Với mỗi chữ cái hiển thị được tạo thành từ 6 đoạn. Nếu đoạn nào không hiển thị thì thay bằng số “32”. Các đoạn để tạo nên các chữ cái hiển thị như trên *hình 3.10*.

Mỗi đoạn có mỗi số đi kèm là trình tự khai báo trong mảng. Mã hex tương ứng là:

```
0x1f,0x1f,0x1f,0x1f,0x1f,0x1f,0x1f,0x1f, //THU 0
0x18,0x1c,0x1e,0x1f,0x1f,0x1f,0x1f,0x1f, //THU 1
0x00,0x00,0x00,0x00,0x00,0x1f,0x1f,0x1f, //THU 2
0x1f,0x1f,0x1f,0x1f,0x1f,0x1e,0x1c,0x18, //THU 3
0x1f,0x1f,0x1f,0x00,0x00,0x00,0x00,0x00, //THU 4
0x1f,0x1f,0x1f,0x00,0x00,0x00,0x1f,0x1f, //THU 5
0x03,0x07,0x0f,0x1f,0x1f,0x1f,0x1f,0x1f, //THU 6
0x1f,0x1f,0x1f,0x00,0x00,0x00,0x00,0x00; //THU 7
```

Để sáng các chữ có nghĩa thì ta kết hợp các đoạn để được các chữ. Mỗi chữ sẽ chiếm 6 vị trí như sau.



Hình 3.10 Ký tự tự tạo được lưu trong CGRAM

Mỗi ký tự lớn sáng thì sử dụng 6 ký tự thường, khi tính thì 3 ký tự hàng 1 trước rồi đến 3 ký tự hàng 2.

Muốn sáng chữ “D” thì các đoạn 0,4,1,0,2,3 sáng.

Muốn sáng chữ “A” thì các đoạn 6,5,1,0,32,0 sáng.

Muốn sáng “-” thì các đoạn 32,32,32,32,7,32 sáng.

Muốn sáng số “1” thì các đoạn 4,1,32,2,0,2 sáng.

CÓ 1 số led hiển thị khoảng trắng có mã là 32.

Sơ đồ kết nối chân LCD với KIT:

LCD16x2	KIT STM32F103C8T6
VSS	GND
VCC	NGUỒN 5V
VEE	CHÂN P _{out} BIẾN TRỞ

RS	PIN_A1
R/W	PIN_A2
E	PIN_A3
D0 – D3	-
D4	PIN_A4
D5	PIN_A5
D6	PIN_A6
D7	PIN_A7
A	NGUỒN 5V
K	GND

Bảng 3. 6 Sơ đồ nối chân của LCD và KIT.

Dùng module giao tiếp là 4bit nên D0-D3 nhóm nghiên cứu không nối dây.

3.3.7 Khối relay 2 kênh.

Căn cứ vào điện áp ngõ ra chân I/O của KIT mức cao là 5v.

Điện áp hoạt động là 5V DC.

Dòng điện là 200 mA/1 relay.

Chịu được hiệu điện thế 250V AC – 10A.

Vì vậy, nhóm nghiên cứu lựa chọn relay module 2 kênh này để làm mô hình hệ thống.



Hình 3. 11 Hình ảnh thực tế module relay 2 kênh

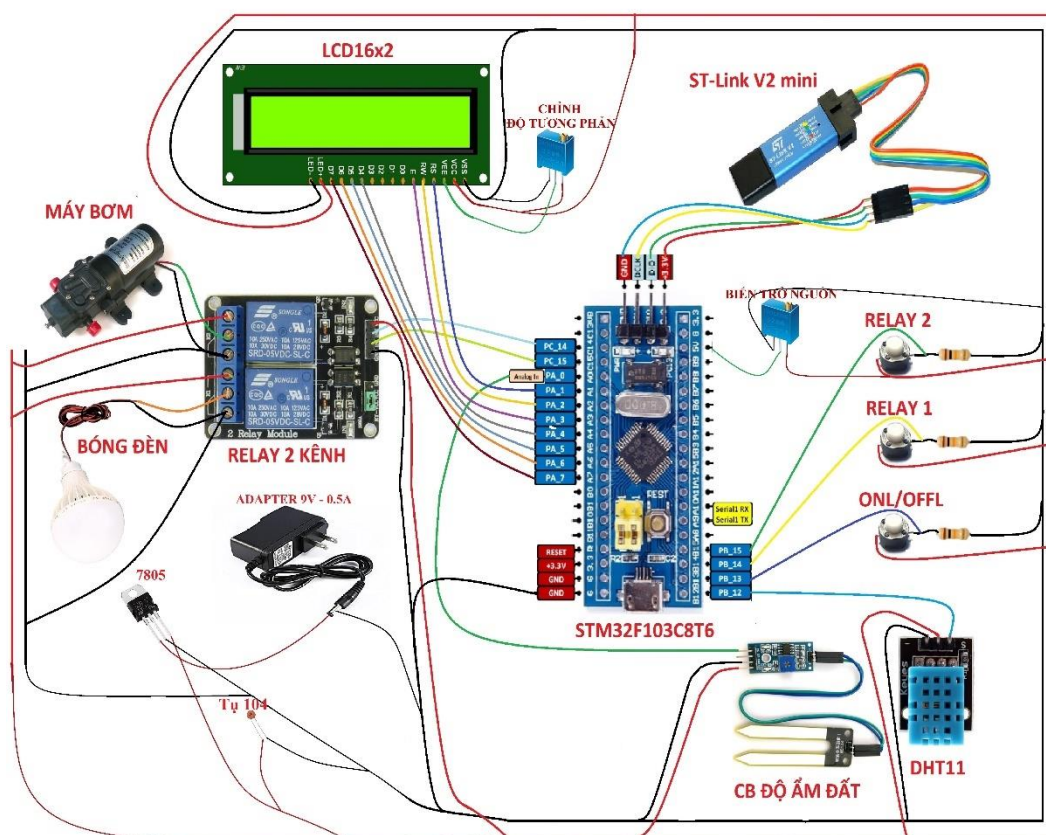
Sơ đồ kết nối chân của module relay 2 kênh.

RELAY	KIT STM32F103C8T6
VCC	NGUỒN 5V
GND	GND
IN0	PIN_C14
IN1	PIN_C15
COM 1/COM2	NỐI 220V AC
NO1/NO2	NỐI ĐÈN/BƠM
NC1/NC2	-

Bảng 3. 7 Kết nối chân của module relay 2 kênh.

Vì khi relay kích thì đèn/bơm sẽ được bật nên nhóm nghiên cứu sẽ nối chân NO1/NO2 vào đèn/bơm, còn chân NC1/NC2 thì để trống.

3.4 Sơ đồ kết nối mạch.



Hình 3. 12 Sơ đồ kết nối chân trong mô hình

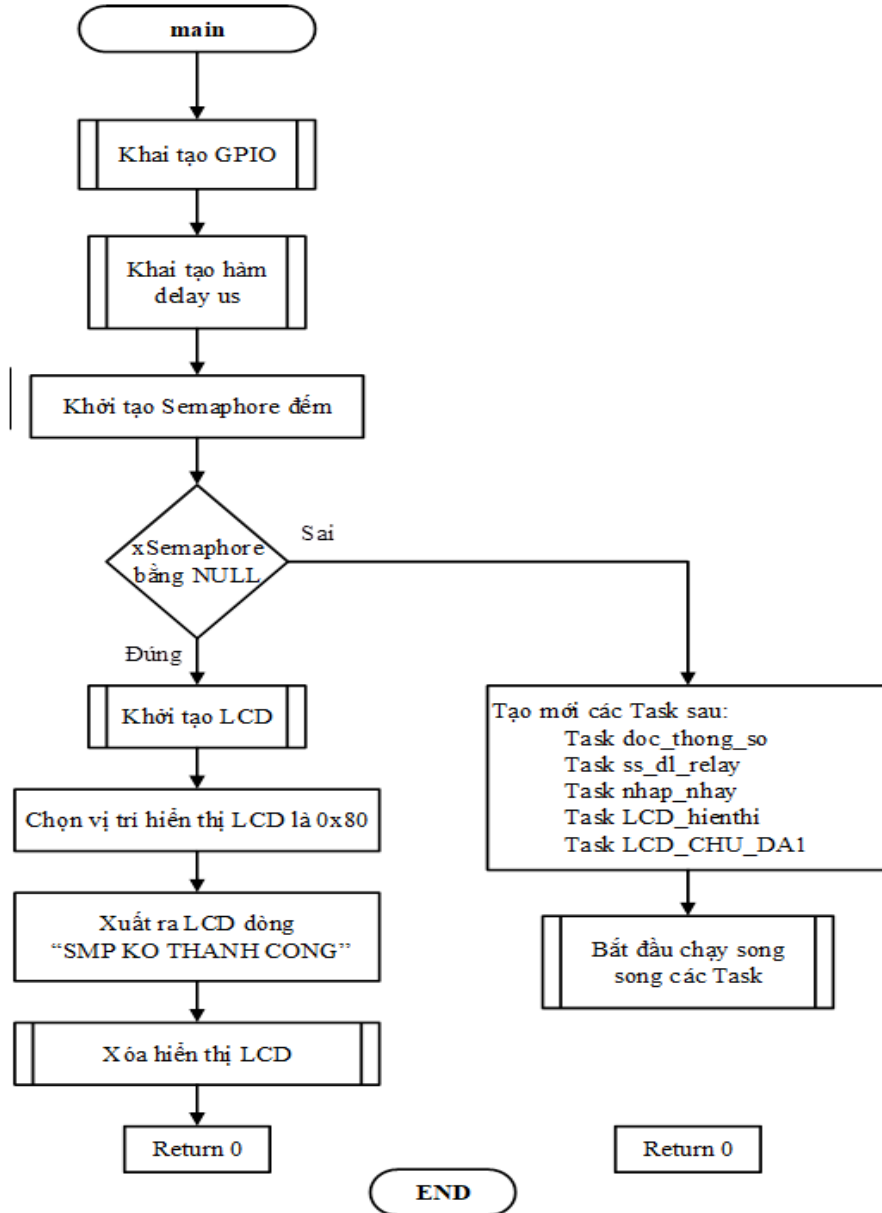
Sơ đồ tổng thể liên kết các linh kiện trong hệ thống. Dây màu đen tương ứng với GND. Dây màu đỏ tương ứng với 5V và nguồn adapter 9V. Các màu còn lại là đường tín hiệu và điều khiển.

CHƯƠNG 4

LƯU ĐỒ GIẢI THUẬT TRONG HỆ THỐNG

4.1 Hàm chính.

Giả sử bạn đầu các biến đã được khai báo toàn cục, các định nghĩa biến,...



Hình 4. 1 Lưu đồ hàm chính của chương trình.

Bước đầu vào chương trình gọi hàm khởi tạo GPIO, khởi tạo các chân cần thiết sử dụng trong hệ thống để giao tiếp với các thiết bị ngoại vi. Khởi tạo hàm delay us từ thư

viện ngoài, vì RTOS không hỗ trợ delay us. Khởi tạo semaphore khiếm đếm để thực hiện đa tác vụ các task. Kết thúc phần khởi tạo.

Kiểm tra điều kiện khởi tạo semaphore nếu không thành công thì thông báo cho người dùng qua LCD hiển thị rồi kết thúc chương trình.

Nếu khởi tạo thành công thì tiến hành tạo 5 task chạy song song: task hiển thị ký tự tạo CGRAM, đọc thông số cảm biến, task điều khiển nút nhấn – relay, task nhấp nháy, task LCD hiển thị. Sau đó tiến hành lập lịch để chạy đa task vụ.

4.2 Hàm con.

Bao gồm có 5 task (chương trình con) trong chương trình.

Chương trình hiển thị ký tự lớn do nhóm nghiên cứu tự tạo.

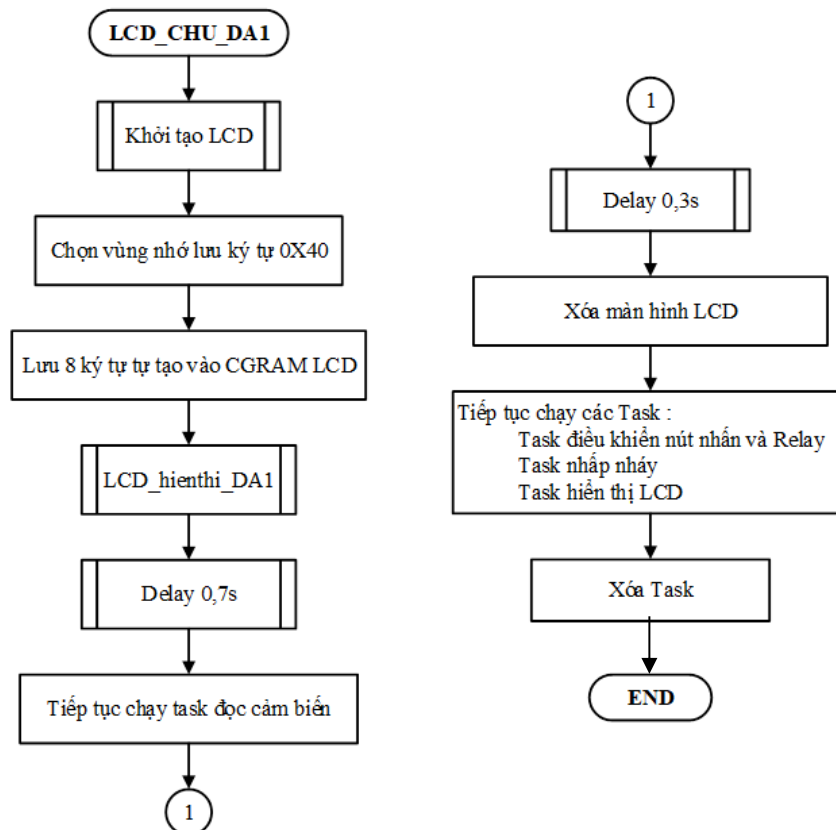
Chương trình nhấp nháy led ở chân C13 để kiểm tra chương trình.

Chương trình đọc thông số cảm biến – cả 2 cảm biến.

Chương trình điều khiển relay từ nút nhấn.

Chương trình hiển thị các thông số lên LCD 16x02.

4.2.1 Lưu đồ tạo và hiển thị ký tự ở CGRAM LCD.



Hình 4. 2 Lưu đồ tạo ký tự và hiển thị từ CGRAM.

Chạy chương trình khởi tạo LCD từ thư viện. Chọn vùng nhớ thứ 0: 0x40 để lưu 8 ký tự tự tạo vào CGRAM. Sau đó, gửi 64 bit, tương ứng 1 ký tự là 1 byte vào CGRAM bắt đầu từ địa chỉ 0x40 - 0x47.

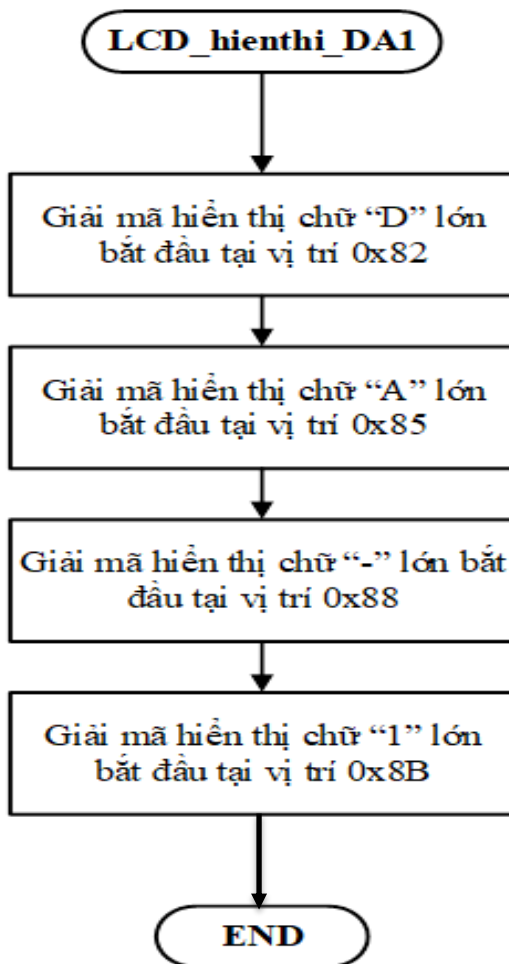
Gọi chương trình hiển thị chữ “DA – 1” để hiển thị các ký tự vừa gửi vào CGRAM. Sau đó delay 0.7s rồi gọi chương trình đọc thông số để cho phép task đọc thông số tiếp tục chạy. Tiếp tục delay 0.3s.

Nhóm nghiên cứu làm như thế là bởi vì, muốn hiển thị chữ “DA – 1” trong thời gian 1s. Nhưng vì task đọc thông số có thời gian khởi động, nên cho task này chạy trước.

Sau đó xóa màn hình LCD bằng lệnh. Cho phép các task đã đứng tiếp tục chạy: task điều khiển nút nhấn – relay, task nhấp nháy, task LCD hiển thị.

Xóa task ra khỏi chương trình lập lịch.

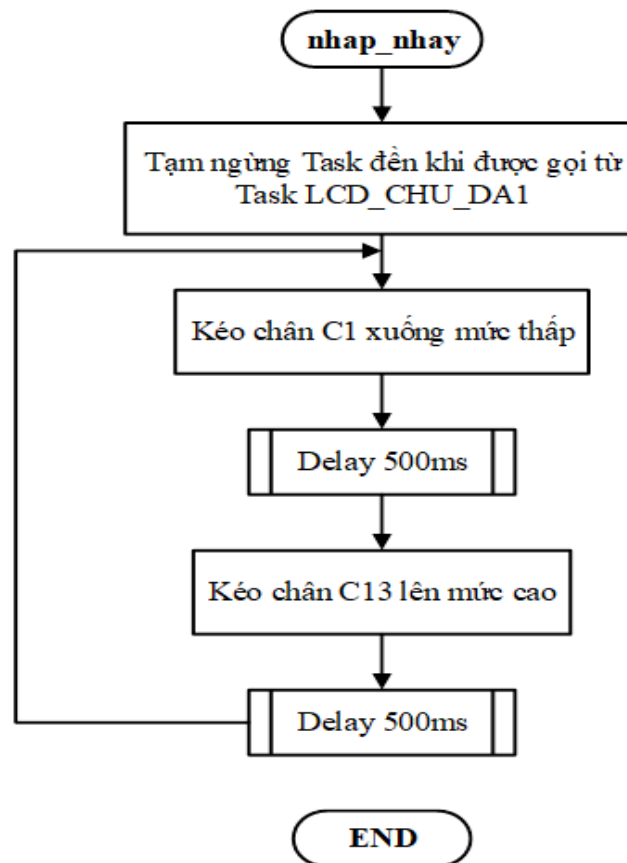
Lưu đồ hiển thị chữ “DA – 1”:



Hình 4. 3 Lưu đồ hiển thị chữ “DA – 1”.

Khi được gọi, chương trình bắt đầu chạy. Giải mã các ký tự tự tạo do nhóm nghiên cứu lưu sẵn trong CGRAM theo thứ tự “D” – “A” – “-” – “1”. Vì thế ta thấy, LCD hiển thị có 1 độ trễ nhất định. Sau khi phân giải xong, chương trình kết thúc.

4.2.2 Lưu đồ nháy nháy led C13.



Hình 4. 4 Lưu đồ nháy nháy led C13.

Sau khi chương trình được lập lịch cấp quyền vào bộ nhớ, chương trình bị đứng lại chờ task LCD_CHU_DA1 chạy đến khi gặp lệnh gọi chương trình thì mới được chạy tiếp tục.

Sau khi được gọi, kéo chân C13 xuống mức thấp, led tắt.

Delay 0.5s.

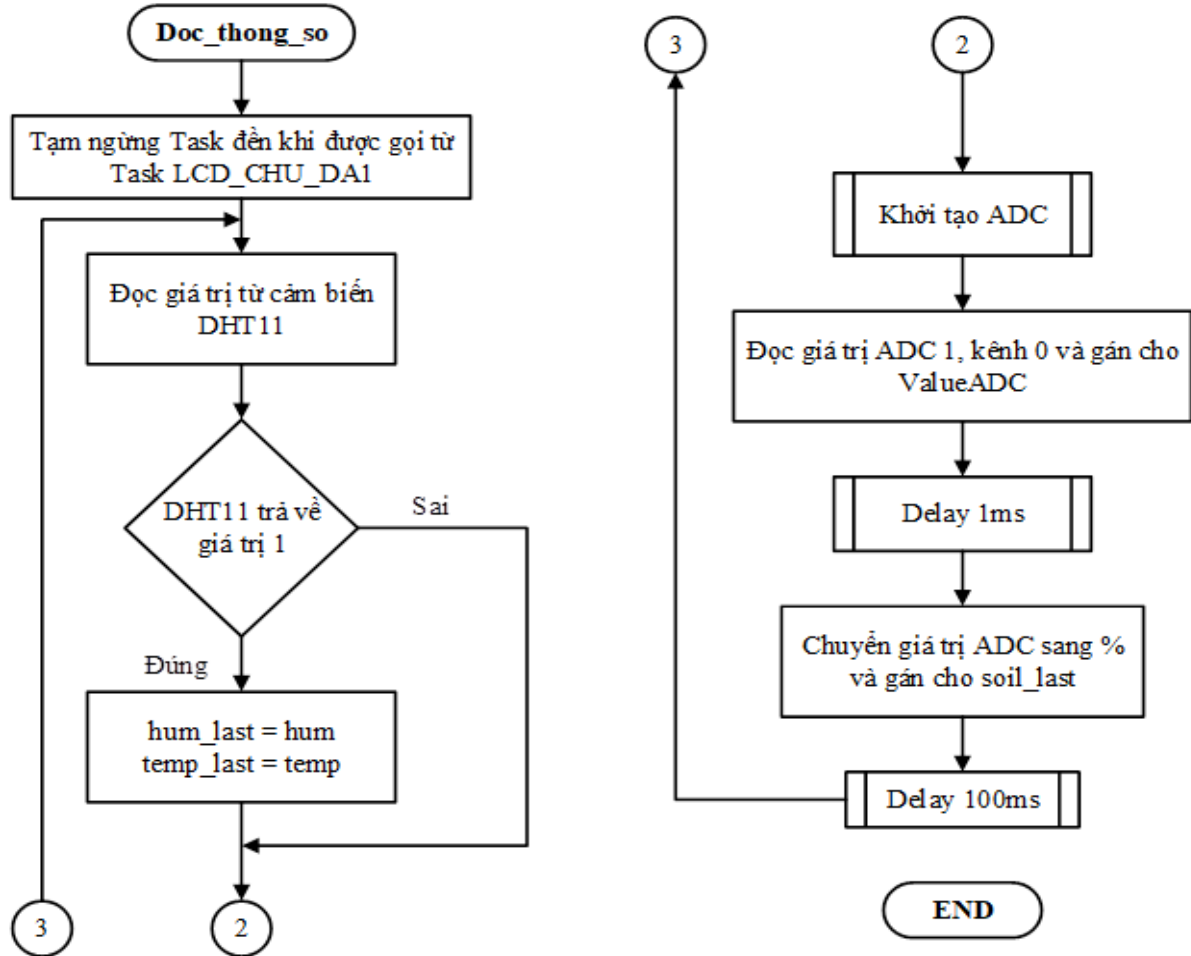
Rồi kéo chân C13 lên mức cao, led sáng.

Delay 0.5s.

Rồi lại kéo chân C13 xuống thấp, led tắt.

Chương trình cứ chạy theo vòng lặp while vô tận, đến khi tắt hệ thống.

4.2.3 Lưu đồ đọc thông số cảm biến.



Hình 4. 5 Đọc thông số thứ 2 cảm biến.

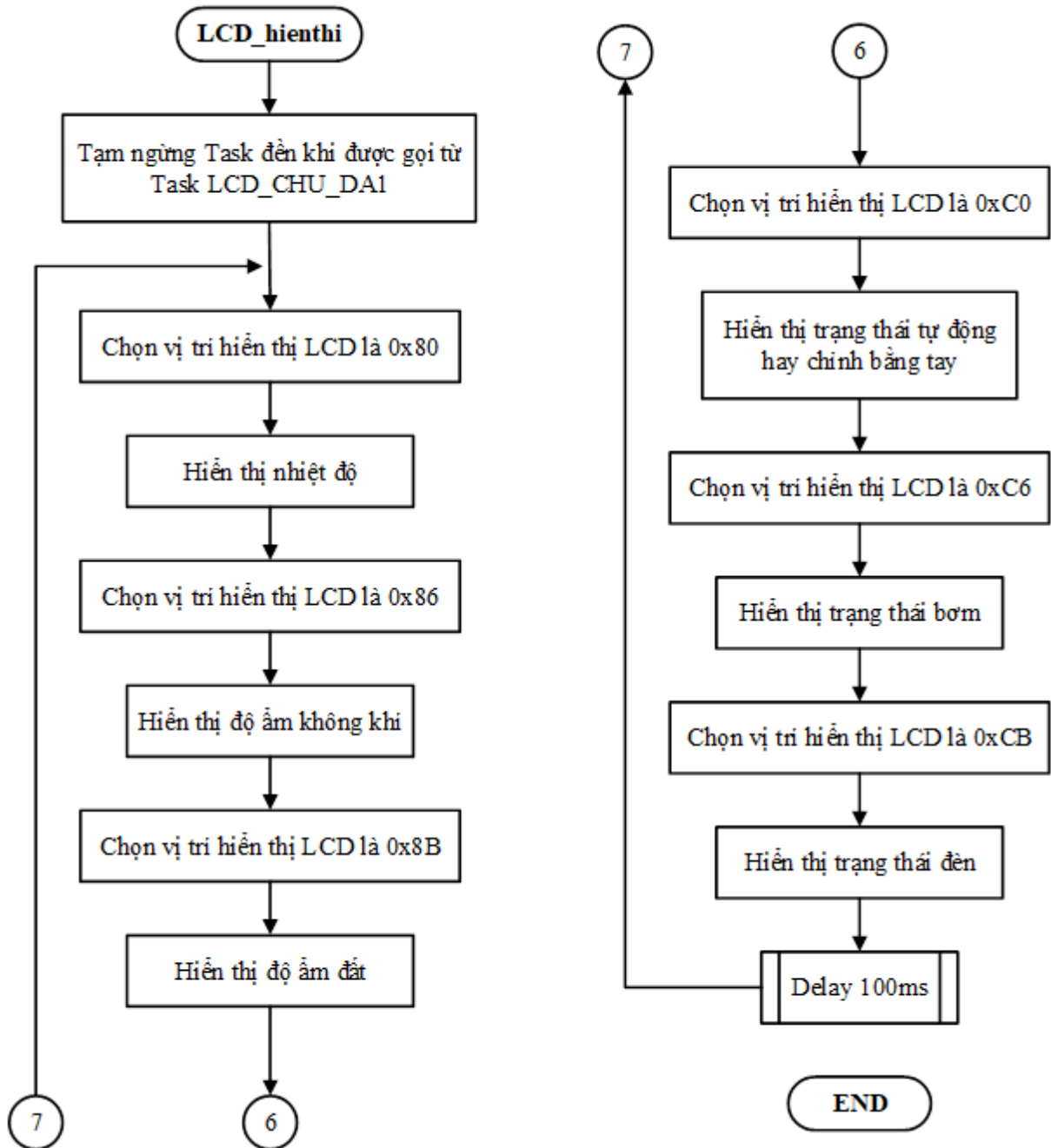
Sau khi chương trình được lập lịch cấp quyền vào bộ nhớ, chương trình bị dừng lại chờ task LCD_CHU_DA1 chạy đến khi gặp lệnh gọi chương trình thì mới được chạy tiếp tục.

Tiến hành chạy chương trình từ thư viện dht11 để giao tiếp với dht11 và nhận dữ liệu. Kiểm tra nếu giao tiếp thành công với dht11 thì gán dữ liệu để đưa lên LCD hiển thị. Nếu không giao tiếp được thì bỏ qua bước lấy dữ liệu.

Khởi tạo ADC để tiến hành đọc cảm biến độ ẩm đất. Tiến hành đọc giá trị ADC1 ở kênh 0. Xử lý giá trị sang giá trị phần trăm và đưa lên LCD để hiển thị.

Delay 100ms để cho các task khác có quyền vào miền găng. Sau đó lại giao tiếp với dht11 để lấy dữ liệu. Chương trình cứ chạy vô tận trong vòng lặp while.

4.2.4 Lưu đồ hiển thị thông số lên LCD.



Hình 4. 6 Hiển thị thông số lên LCD.

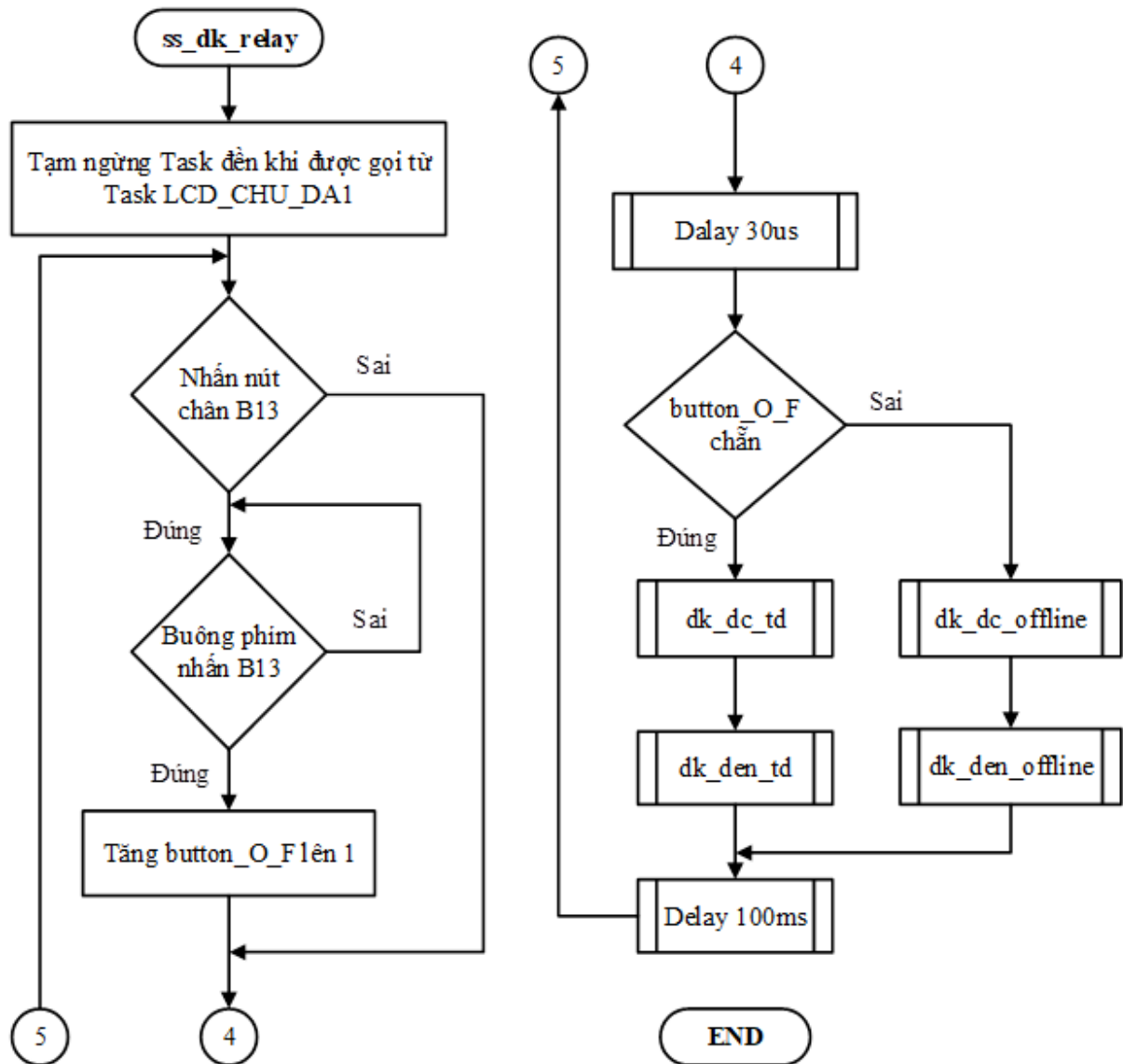
Sau khi chương trình được lập lịch cấp quyền vào bộ nhớ, chương trình bị đứng lại chờ task LCD_CHU_DA1 chạy đến khi gặp lệnh gọi chương trình thì mới được chạy tiếp tục.

Chọn vị trí và hiển thị và hiển thị thông số.

0x80: nhiệt độ. 0x86: độ ẩm. 0x8B: độ ẩm đất. 0xC0: hiển thị trạng thái online, offline. 0xC6: hiển thị trạng thái bơm. 0xCB: hiển thị trạng thái đèn.

Delay 100ms để cấp quyền cho task khác. Sau đó quay về vị trí 0x80 và hiển thị nhiệt độ. Chạy vô tận với vòng lặp while luôn đúng.

4.2.5 Lưu đồ điều khiển relay.



Hình 4. 7 Lưu đồ điều khiển relay.

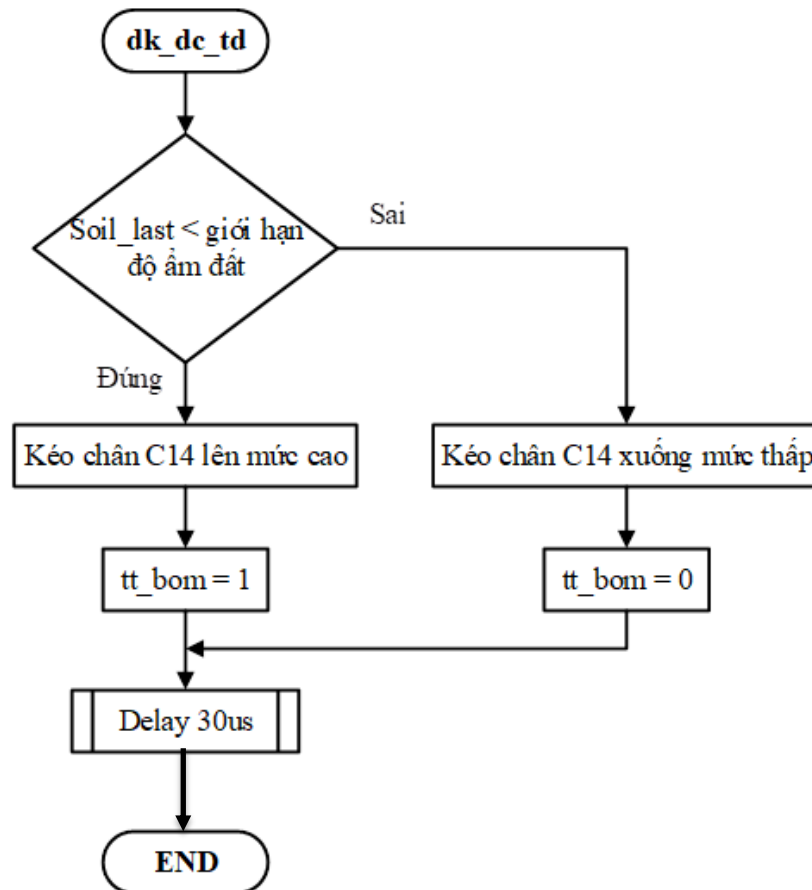
Sau khi chương trình được lập lịch cấp quyền vào bộ nhớ, chương trình bị đứng lại chờ task LCD_CHU_DA1 chạy đến khi gặp lệnh gọi chương trình thì mới được chạy tiếp tục.

Kiểm tra nút nhấn B13 có nhấn hay không. Nếu có thì chờ buông phím rồi tăng biến đếm online_offline lên 1. Nếu không nhấn thì không làm gì cả. Sau đó delay 30 us.

Kiểm tra biến đếm online_offline nếu chẵn thì gọi hàm điều khiển đèn và bơm qua relay 2 kênh tự động. Nếu biến đếm online_offline lẻ thì điều khiển đèn và bơm qua relay 2 kênh bằng tay qua 2 nút nhấn C14, C15.

Delay 100ms, sau đó lặp lại vô tận với vòng lặp while true luôn đúng.

Lưu đồ điều khiển động cơ tự động:



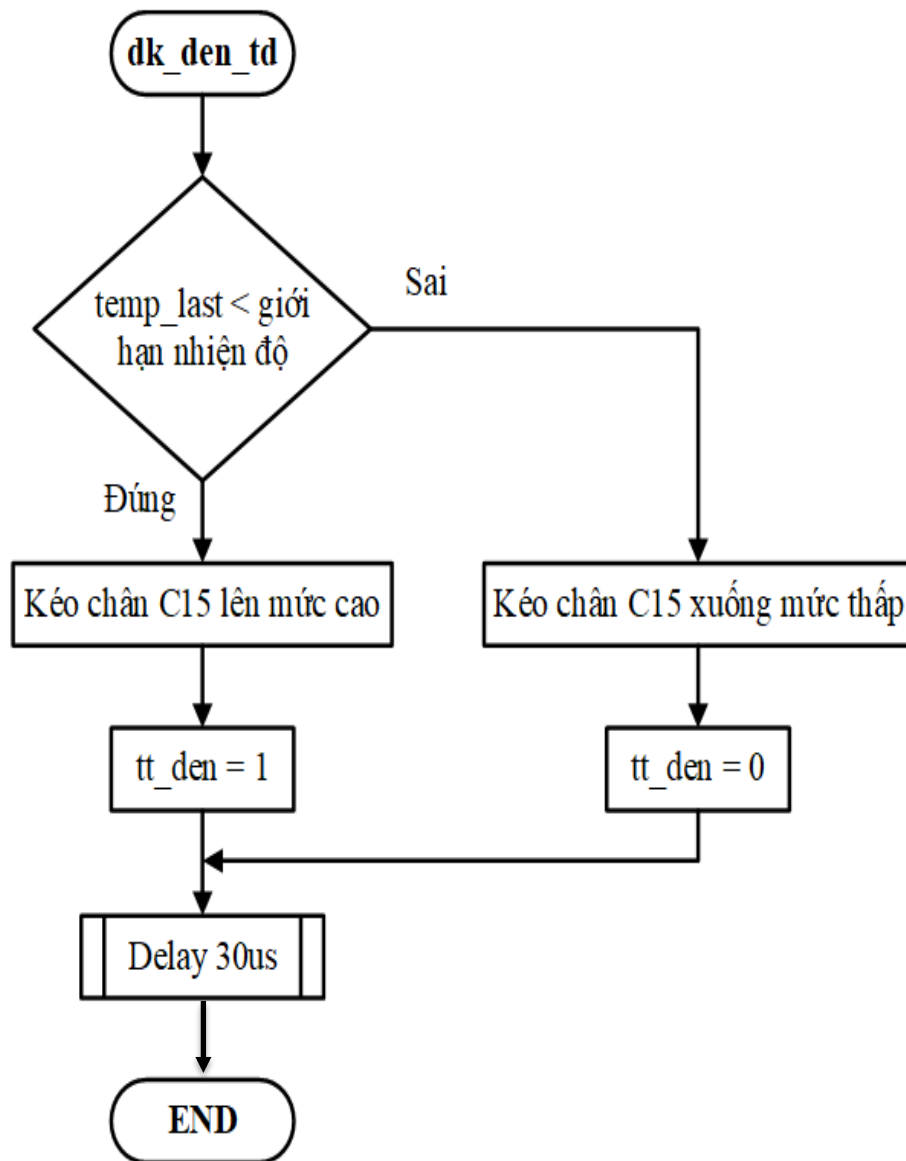
Hình 4. 8 Lưu đồ điều khiển động cơ.

Khi hàm được gọi từ chương trình cha. Chương trình sẽ thực thi.

Nếu giá trị đọc từ cảm biến độ ẩm đất nhỏ hơn giới hạn dưới của độ ẩm đất được định nghĩa ban đầu thì kéo chân C14 lên mức cao – kích relay hoạt động và đổi trạng thái tt_bom = 1 – hiển thị LCD. Nếu sai thì kéo chân C14 xuống thấp và tt_bom = 0.

Delay 30us sau đó kết thúc chương trình.

Lưu đồ điều khiển đèn tự động:



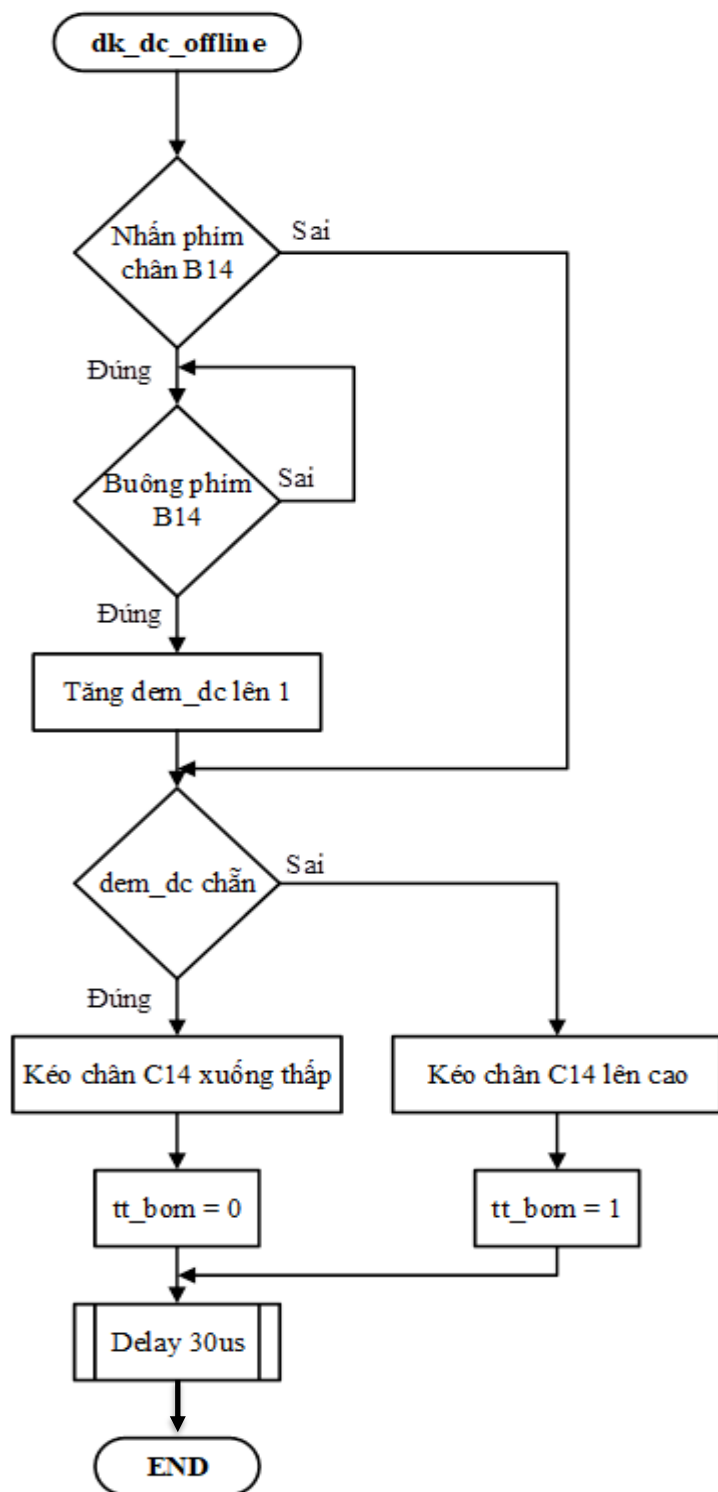
Hình 4. 9 Lưu đồ điều khiển đèn tự động.

Khi hàm được gọi từ chương trình cha. Chương trình sẽ thực thi.

Nếu giá trị nhiệt độ đọc từ dht11 nhỏ hơn giới hạn nhiệt độ định nghĩa trước được định nghĩa ban đầu thì kéo chân C15 lên mức cao – kích relay hoạt động và đổi trạng thái tt_den = 1 – hiển thị LCD. Nếu sai thì kéo chân C14 xuống thấp và tt_den = 0.

Delay 30us sau đó kết thúc chương trình.

Lưu đồ điều khiển động cơ bằng tay:



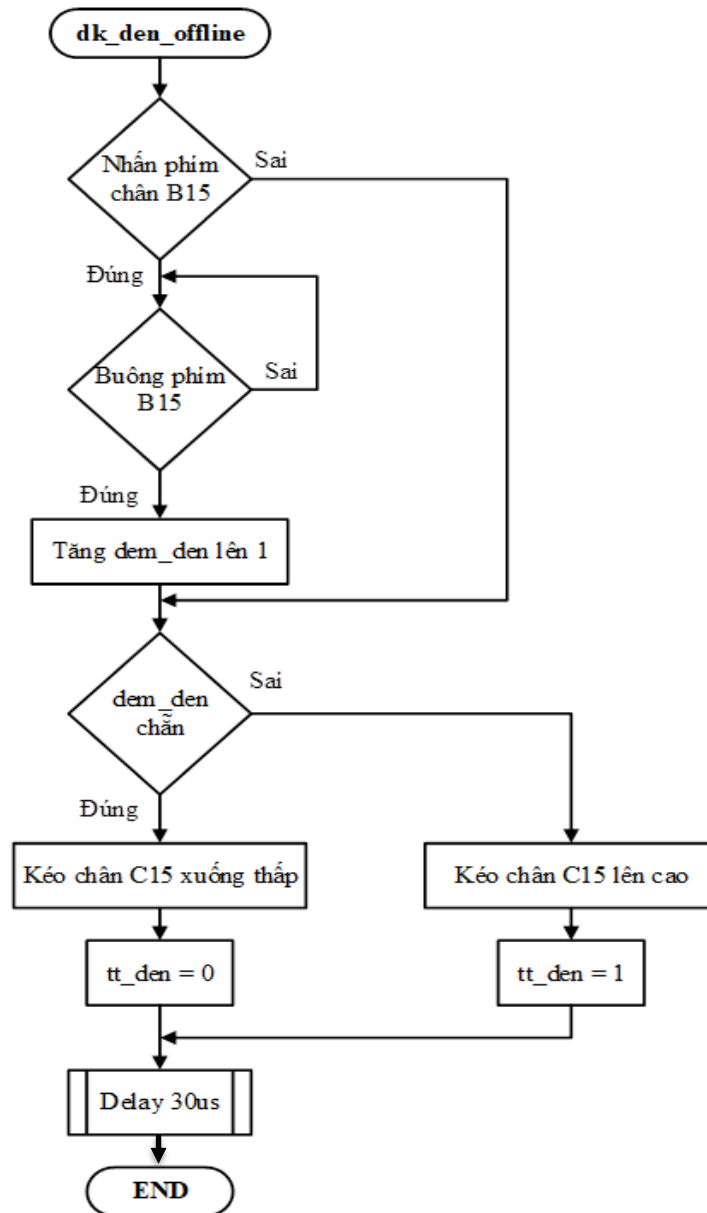
Hình 4. 10 Lưu đồ điều khiển động cơ bằng tay.

Khi hàm được gọi từ chương trình cha. Chương trình sẽ thực thi.

Kiểm tra có nhấn phím chân C14 hay không, nếu có thì chờ nhả phím. Sau khi nhả phím, tăng biến dem_dc lên 1. Nếu không nhấn phím thì không tăng dem_dc. Nếu dem_dc là số chẵn thì kéo chân C14 xuống thấp, đồng thời gán tt_bom = 0. Nếu dem_dc là số lẻ thì kéo chân C14 lên cao, đồng thời gán tt_bom = 1

Sau đó delay 30us rồi kết thúc chương trình.

Lưu đồ điều khiển đèn bằng tay:



Hình 4. 11 Lưu đồ điều khiển đèn bằng tay

Khi hàm được gọi từ chương trình cha. Chương trình sẽ thực thi.

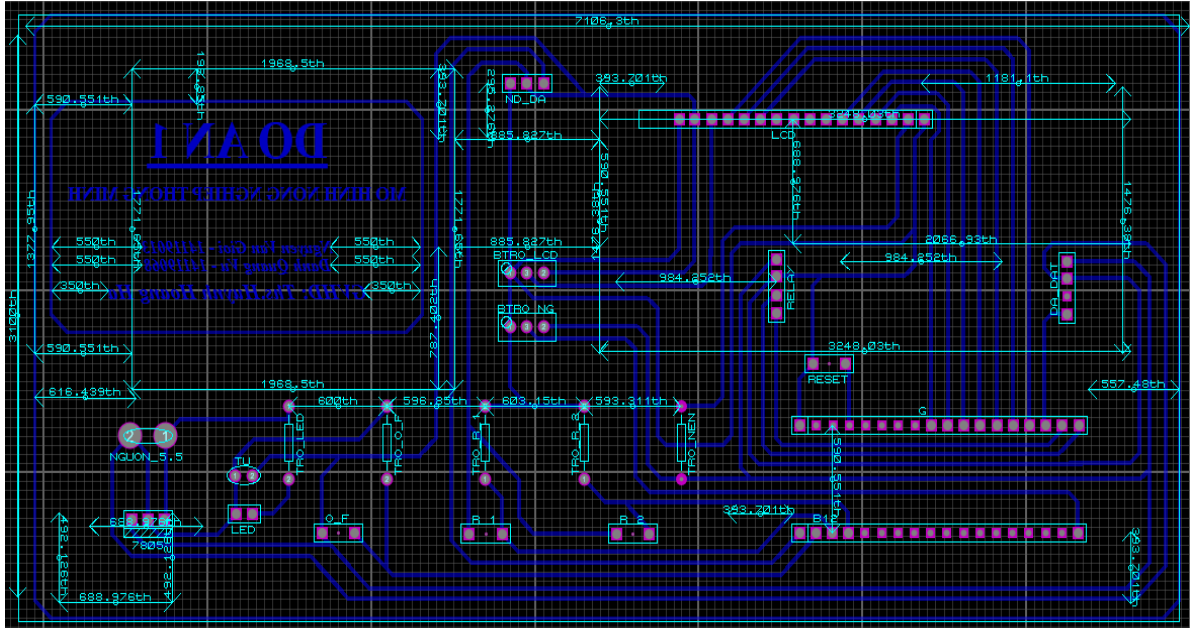
Kiểm tra có nhấn phím chân C15 hay không, nếu có thì chờ nhả phím. Sau khi nhả phím, tăng biến dem_den lên 1. Nếu không nhấn phím thì không tăng dem_den. Nếu dem_dc là số chẵn thì kéo chân C14 xuống thấp, đồng thời gán tt_den = 0. Nếu dem_dc là số lẻ thì kéo chân C14 lên cao, đồng thời gán tt_den = 1

Sau đó delay 30us rồi kết thúc chương trình.

CHƯƠNG 5

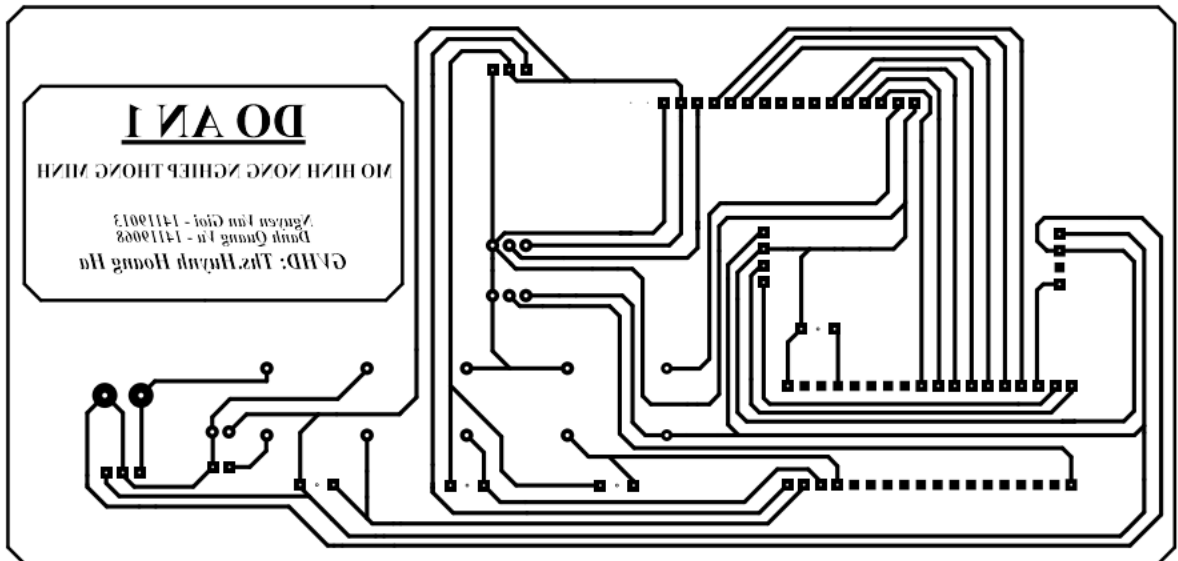
KẾT QUẢ VÀ ĐÁNH GIÁ

5.1 Sơ đồ thiết kế hệ thống.



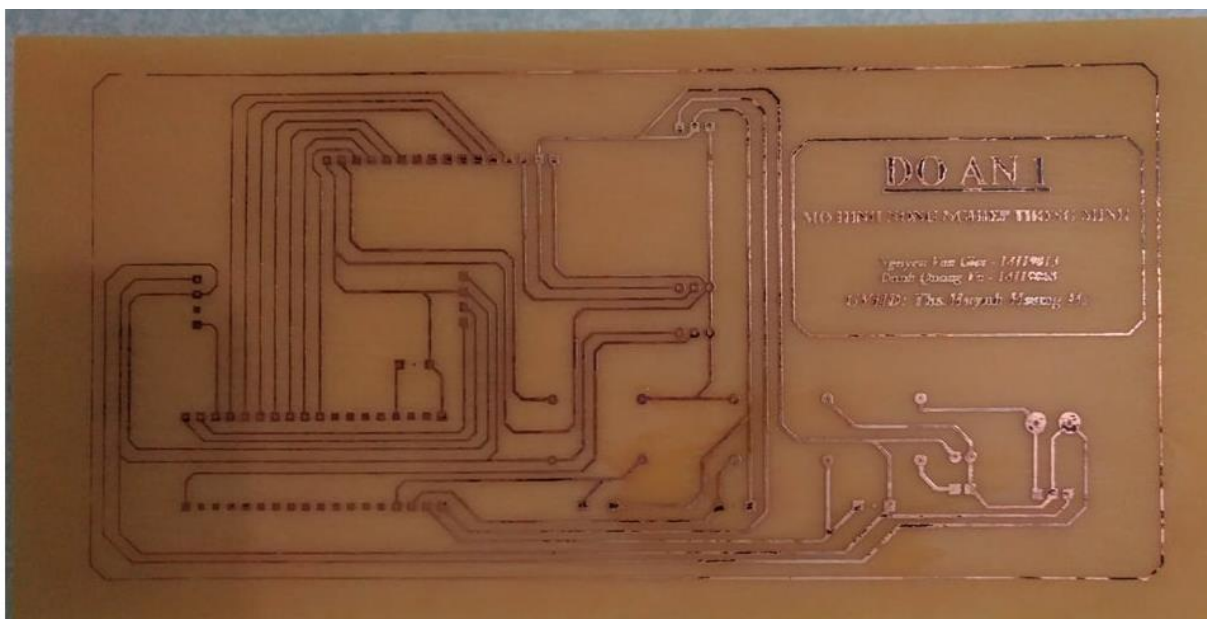
Hình 5. 1 Sơ đồ layout thiết kế trên Proteus

Sơ đồ phân bố linh kiện được thiết kế trên proteus đáp ứng các yêu cầu của hệ thống.



Hình 5. 2 Sơ đồ mạch in

Sơ đồ mạch in sau khi layout từ proteus.



Hình 5. 3 Sản phẩm mạch in

Mạch in hoàn thiện được in trên board đồng. Nhóm nghiên cứu đã đo đạt và đảm bảo không bị đứt hay chong dây trong mạch.

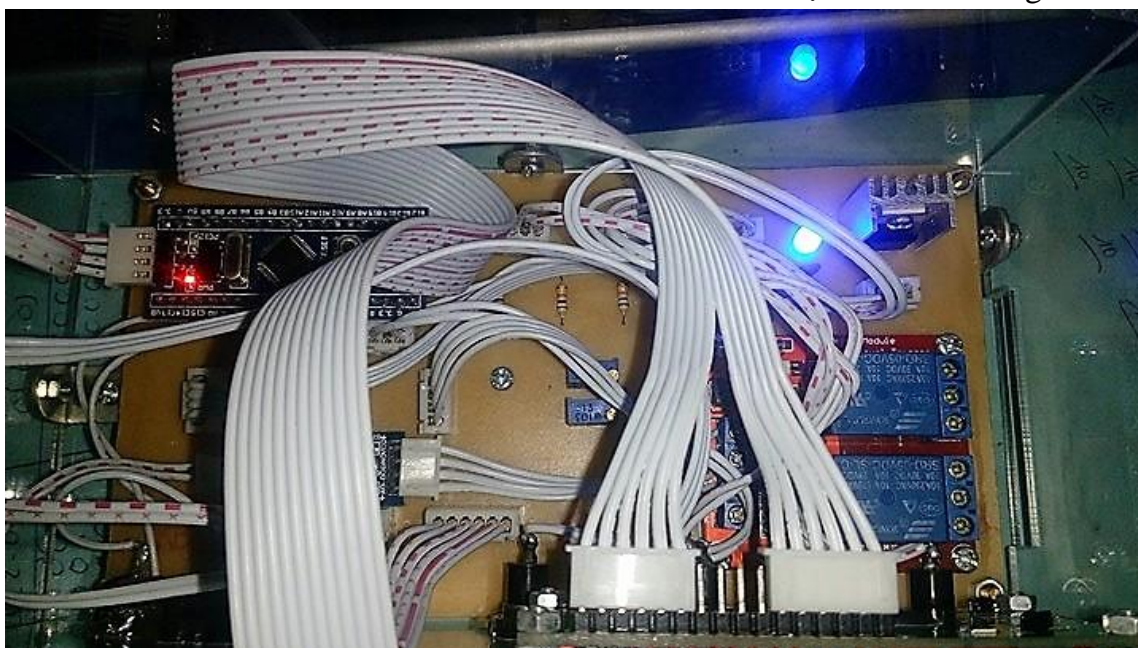
5.2 Sản phẩm hoàn thiện.

Sản phẩm sau khi thiết kế hoàn thiện có thể hiển thị được ký tự tự tạo. Đọc được các thông số giá trị độ ẩm, nhiệt độ từ dht11; đọc được cảm biến độ ẩm đất. Tất cả các thông số đều được hiển thị lên LCD. Căn cứ vào giá trị đọc được điều khiển được relay module 2 kênh hoặc có thể điều khiển bằng tay thông qua 3 nút nhấn cài đặt.



Hình 5. 4 Mặt trước của sản phẩm

Mặt trước bao gồm cổng 5.5 mm và công tắc nguồn. Có 3 nút nhấn điều khiển màu xanh và 1 nút reset màu đỏ. Màn hình LCD 16x02 hiển thị tất cả các thông số.



Hình 5. 5 Mặt nhìn từ trên xuống

Bao gồm các linh kiện được thiết kế gọn trong hộp bảo vệ để chống các tác động từ môi trường bên ngoài, tăng tuổi thọ cho các thiết bị.

Chỉ có đầu cắm cảm biến độ ẩm đất và dht11 là tiếp xúc với môi trường bên ngoài.

5.3 Đánh giá đề tài.

5.3.1 Ưu điểm.

Hệ thống hoạt động thời gian thực, hoạt động song song giữa các task vụ nên hoạt động này không ảnh hưởng đến hoạt động khác, hoạt động một cách độc lập nhau.

Thiết kế nhỏ gọn, dễ di chuyển, thuận lợi để mọi nơi trong trang trại.

Thiết kế gọn trong hộp nhựa, cách ly với môi trường ngoài nên tuổi thọ linh kiện được kéo dài.

Chi phí thiết kế thấp, giá linh kiện rẻ, dễ tìm.

Hệ thống được điều khiển bởi kit ARM, được hỗ trợ rộng rãi bởi cộng đồng lập trình.

Hệ thống giúp cho chủ trang trại giảm nhân công, công sức trong việc tưới nước, gia nhiệt, giúp cho môi trường thuận lợi cho cây trồng phát triển tốt.

Hệ thống có khả năng nâng cấp và có tính ứng dụng rộng rãi.

5.3.2 Nhược điểm.

Trong quá trình hoạt động, gặp một số hạn chế như sau:

DHT11 nhiều lúc không giao tiếp được vì yêu cầu quá chặt chẽ về mặt thời gian.
Đầu cảm cảm biến độ ẩm đất tiếp xúc môi trường nước lâu ngày gây nên hiện tượng ăn mòn sắt dẫn đến đo đạt không chính xác.

DHT11 tiếp xúc trực với môi trường ngoài, tuổi thọ không được kéo dài.

Phím nhấn cơ tiếp xúc không tốt, đôi lúc không thực hiện được.

CHƯƠNG 6

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận.

6.1.1 Những vấn đề nghiên cứu.

Nghiên cứu ứng dụng hệ thống nhúng trong thực tiễn.

Nghiên cứu về hệ thống thời gian thực – RTOS.

Nghiên cứu lập trình ARM trên Keil C.

Nghiên cứu tạo ký tự lưu vào vùng nhớ CGRAM của LCD 16x02.

Nghiên cứu cách đọc, tính toán và gửi dữ liệu lên KIT ARM.

6.1.2 Những vấn đề hoàn thành.

Hoàn thành việc chạy song song các tác vụ trên hệ thống RTOS trên ARM.

Hoàn thành việc giao tiếp và lấy được dữ liệu từ DHT11 và cảm biến độ ẩm đất.

Hoàn thành việc hiển thị ký tự tự tạo lên màn hình LCD.

Hoàn thành việc điều khiển relay để điều khiển bơm và đèn.

6.1.3 Những hạn chế của đề tài.

Bên cạnh các vấn đề hoàn thành phía trên thì nhóm nghiên cứu cảm thấy đề tài có một số hạn chế như sau:

Đề tài chưa thật sự mới mẻ và ấn tượng.

Chưa nghiên cứu đọc nhiều cảm biến cùng lúc với quy mô trang trại cây trồng lớn.

Chưa gửi được thông số và điều khiển qua wifi hoặc từ xa.

6.2 Hướng phát triển của đề tài.

Từ những hạn chế ở mục trên thì người kế thừa đề tài cần phát triển thêm nhiều tính năng để sản phẩm thật sự có ích và có thể áp dụng vào trang trại với quy mô lớn.

Tăng số lượng cảm biến và kích thước màn hình hiển thị LCD lớn hơn.

Có thể theo dõi và điều khiển relay qua mạng internet hoặc qua ứng dụng trên điện thoại. Có thể giám sát trang trại từ xa được.

TÀI LIỆU THAM KHẢO

Tên sách tham khảo:

- [1] Nguyễn Đình Phú (2015), “Giáo trình vi điều khiển PIC”, Đại học Sư Phạm Kỹ Thuật TP.HCM, Khoa Điện – Điện Tử - Bộ môn Điện tử Công nghiệp.
- [2] Nguyễn Đình Phú (2014), “Thực hành vi điều khiển – ARM STM32”, Đại học Sư Phạm Kỹ Thuật TP.HCM, Khoa Điện – Điện Tử - Bộ môn Điện tử Công nghiệp.

Trang web:

<http://www.freertos.org/>

<http://www2.keil.com/mdk5>

<http://www.alldatasheet.com/>

<http://www.datasheetspdf.com/datasheet/DHT11.html>

<http://www.st.com/en/microcontrollers/stm32f103c8.html>

<http://arduino.vn/bai-viet/302-module-relay-cach-su-dung-ro-le-va-nhung-ung-dung-hay-cua-no>

<https://sites.google.com/site/maicuongtho/vi-dieu-khien/kien-thuc-bo-tro/dhientucoban-tacdungcuadientrokeolenkeoxuong>

<http://thanhnt.com/blog/co-ban-dien-tro-keo-la-gi/>

<http://www.alldatasheet.com/datasheet-pdf/pdf/82833/FAIRCHILD/LM7805.html>

PHỤ LỤC

```
#include "main.h"
#include "var.h"

#define GH_ND_TREN 40
#define GH_ND_DUOI 25
#define GH_DA_TREN 60
#define GH_DA_DUOI 40

#define CHAN_B13 GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_13)
#define CHAN_B14 GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_14)
#define CHAN_B15 GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_15)

const unsigned char LCD_CHU_X_LAN1[4][6] = {
    0,4,1,0,2,3,
    6,5,1,0,32,0,
    32,32,32,32,7,32,
    4,1,32,2,0,2
};

const unsigned char LCD_MA_8DOAN_CHU_CAI_LAN1[] = {
    0X1F,0X1F,0X1F,0X1F,0X1F,0X1F,0X1F,0X1F, //THU 0
    0x18,0x1c,0x1e,0x1f,0x1f,0x1f,0x1f,0x1f, //THU 1
    0x00,0x00,0x00,0x00,0x00,0x1f,0x1f,0x1f, //THU 2
    0x1f,0x1f,0x1f,0x1f,0x1f,0x1e,0x1c,0x18, //THU 3
    0x1f,0x1f,0x1f,0x00,0x00,0x00,0x00,0x00, //THU 4
    0x1f,0x1f,0x1f,0x00,0x00,0x00,0x1f,0x1f, //THU 5
    0x03,0x07,0x0f,0x1f,0x1f,0x1f,0x1f,0x1f, //THU 6
    0X1F,0X1F,0X1F,0X00,0X00,0X00,0X00,0X00}; //THU 7

int_fast32_t j=0,jj;
unsigned char temp, hum, trave_dht11;
int i, ValueADC, button_O_F = 0;
int dem_dc=0, dem_den =0, tt_bom=0, tt_den = 0, tam1;
int soil_last, temp_last, hum_last;
int Dodai_temp = 0, Dodai_humi = 0, Dodai_soil = 0;
```

```
char    tt_bom_hienthi[1]    ,tt_den_hienthi[1],    temp_hienthi[1],    hum_hienthi[1],
soil_hienthi[1];
```

```
xSemaphoreHandle xSemaphore=NULL;
TaskHandle_t xHandle1;
TaskHandle_t xHandle2;
TaskHandle_t xHandle3;
TaskHandle_t xHandle4;
void doc_thong_so(void *ptr);
void LCD_hienthi(void *ptr);
void ss_dk_relay(void *ptr);
void nhap_nhay(void *ptr);
void LCD_CHU_DA1(void *ptr);
void LCD_hienthi_DA1(void);
void dk_dc_td(void);
void dk_den_td(void);
void dk_dc_offline(void);
void dk_den_offline(void);
int main(void)
{
    GPIO_Configuration();
    initUsTimer();
    xSemaphore = xSemaphoreCreateCounting(1,0);
    if(xSemaphore == NULL)
    {
        LCD_Init();
        LCD_Gotoxy(0,0);
        LCD_Puts((uint8_t *)"SMP KO THANH CONG");
        LCD_Clear();
        return 0;
    }
    else
    {
        xTaskCreate(doc_thong_so,
        NULL,configMINIMAL_STACK_SIZE,NULL,2,&xHandle1);
```

```

        xTaskCreate(ss_dk_relay,
        NULL,configMINIMAL_STACK_SIZE,NULL,1,&xHandle2);
        xTaskCreate(nhap_nhay,
        NULL,configMINIMAL_STACK_SIZE,NULL,0,&xHandle3);
        xTaskCreate(LCD_hienthi,
        NULL,configMINIMAL_STACK_SIZE,NULL,0,&xHandle4);
        xTaskCreate(LCD_CHU_DA1,
        NULL,configMINIMAL_STACK_SIZE,NULL,0,NULL);
        vTaskStartScheduler();
        return 0;
    }
}

void LCD_HIENITHI_SO_Z_TOADO_XY(int LCD_SO, int X1, int Y1)
{
    LCD_Gotoxy(X1,Y1);
    for(i=0;i<6;i++)
    {
        if (i == 3) LCD_Gotoxy(X1,Y1+1);
        LCD_PutChar(LCD_CHU_X_LAN1[LCD_SO][i]);
    }
}

void LCD_CHU_DA1(void *ptr)
{
    LCD_Init();
    LCD_Command(0X40);
    for (j = 0;j<64;j++)
    LCD_PutChar(LCD_MA_8DOAN_CHU_CAI_LAN1[j]);
    LCD_hienthi_DA1();
    vTaskDelay(700);
    vTaskResume(xHandle1); //doc cam bien
    vTaskDelay(300);
    LCD_Clear();
    vTaskResume(xHandle2); // dieu khien nut nhan va relay
    vTaskResume(xHandle3); //nhap nhay
}

```

```

        vTaskResume(xHandle4); //LCD hien thi
        vTaskDelete(NULL);
    }

void doc_thong_so(void *ptr)
{
    vTaskSuspend(xHandle1);
    while(1)
    {
        trave_dht11 = DHT_GetTemHumi(&temp,&hum) ;
        if(trave_dht11 == 1)
        {
            hum_last=hum;
            temp_last=temp;
        }
        ADC1_Configuration();
        ValueADC=read_adc(ADC1,0);
        //Ra gia tri tu 2200 (do am bang 100%) - 4095(do am bang 0)
        vTaskDelay(1);
        tam1 = 4095 - ValueADC;
        soil_last = (tam1*100)/1895;
        if (soil_last > 100) soil_last = 100;
        vTaskDelay(100);
    }
}

void LCD_hienthi(void *ptr)
{
    vTaskSuspend(xHandle4);
    while(1)
    {
        //NHIET DO CAM BIEN
        LCD_Gotoxy(0,0);
        sprintf(temp_hienthi,"%d",temp_last);
        LCD_Puts((uint8_t *)"T");
    }
}

```

```

LCD_Puts((uint8_t *)temp_hienthi);
Dodai_temp = strlen(temp_hienthi);
LCD_RS=1;
LCD_SendCommand(0xDF);
LCD_RS=0;
LCD_Puts((uint8_t *)"C");
for (int l=2+Dodai_temp;l<5;l++)
{
    LCD_Gotoxy(l+1,0);
    LCD_Puts((uint8_t *)" ");
}
//DO AM CAM BIEN
LCD_Gotoxy(6,0);
sprintf(hum_hienthi,"%d",hum_last);
LCD_Puts((uint8_t *)"H");
LCD_Puts((uint8_t *)hum_hienthi);
Dodai_humi = strlen(hum_hienthi);
LCD_Puts((uint8_t *)"%");
for (int l=8+Dodai_humi;l<11;l++)
{
    LCD_Gotoxy(l,0);
    LCD_Puts((uint8_t *)" ");
}

//DO AM DAT CAM BIEN
LCD_Gotoxy(11,0);
sprintf(soil_hienthi,"%d",soil_last);
LCD_Puts((uint8_t *)"S");
LCD_Puts((uint8_t *)soil_hienthi);
Dodai_soil = strlen(soil_hienthi);
LCD_Puts((uint8_t *)"%");
for (int l=13+Dodai_soil;l<16;l++)
{
    LCD_Gotoxy(l,0);
    LCD_Puts((uint8_t *)" ");
}

```

```

    }
//HIEN THI TRANG THAI
    LCD_Gotoxy(0,1);
    LCD_Puts((uint8_t *)"TT:");
    if (button_O_F%2 ==0)
        LCD_Puts((uint8_t *)"ON");
    else
        LCD_Puts((uint8_t *)"OF");

//HIEN THI CHE DO
    LCD_Gotoxy(6,1);
    sprintf(tt_bom_hienthi,"%d",tt_bom);
    LCD_Puts((uint8_t *)"B:");
    LCD_Puts((uint8_t *)tt_bom_hienthi);

    LCD_Gotoxy(11,1);
    sprintf(tt_den_hienthi,"%d",tt_den);
    LCD_Puts((uint8_t *)"D:");
    LCD_Puts((uint8_t *)tt_den_hienthi);
    vTaskDelay(100);
}
}

void nhap_nhay(void *ptr)
{
    vTaskSuspend(xHandle3);
    while(1)
    {
        GPIO_WriteBit(GPIOC,GPIO_Pin_13,Bit_RESET);
        vTaskDelay(500);
        GPIO_WriteBit(GPIOC,GPIO_Pin_13,Bit_SET);
        vTaskDelay(500);
    }
}

```

```

void ss_dk_relay(void *ptr)
{
    vTaskSuspend(xHandle2);
    while (1)
    {
        if (CHAN_B13 == 1)
        {
            while (CHAN_B13== 1){};
            button_O_F++;
        }
        delayUs(30);
        if (button_O_F%2 ==0)
        {
            dk_dc_td();
            dk_den_td();
        }
        else
        {
            dk_dc_offline();
            dk_den_offline();
        }
        vTaskDelay(100);
    }
}

void dk_dc_td()
{
    //DIEU KHIEN DONG CO QUA RELAY 1 TU DONG.
    if (soil_last < GH_DA_DUOI)
    {
        GPIO_WriteBit(GPIOC,GPIO_Pin_14,Bit_SET);
        tt_bom = 1;
    }
    else
    {

```



```

        GPIO_ResetBits (GPIOC,GPIO_Pin_14);
        tt_bom = 0;
    }
    delayUs(30);
}

void dk_den_td()
{
    //DIEU KHIEN DEM QUA RELAY_TU DONG.
    if (temp_last < GH_ND_DUOI)
    {
        GPIO_SetBits(GPIOC,GPIO_Pin_15);
        tt_den = 1;
    }
    else
    {
        GPIO_ResetBits (GPIOC,GPIO_Pin_15);
        tt_den = 0;
    }
    delayUs(30);
}

void dk_dc_offline()
{
    //DIEU KHIEN DONG CO QUA RELAY 1 BANG TAY
    if (CHAN_B14== 1)
    {
        while (CHAN_B14== 1);
        dem_dc++;
    }
    if (dem_dc%2 ==0)
    {
        GPIO_ResetBits (GPIOC,GPIO_Pin_14);
        tt_bom = 0;
    }
}

```

```

else
{
    GPIO_SetBits(GPIOC,GPIO_Pin_14);
    tt_bom = 1;
}
delayUs(30);
}
void dk_den_offline()
{
    //DIEU KHIEN DEN QUA RELAY 2 BANG TAY
    if (CHAN_B15== 1)
    {
        while (CHAN_B15== 1);
        dem_den++;
    }
    if (dem_den%2==0)
    {
        GPIO_ResetBits (GPIOC,GPIO_Pin_15);
        tt_den = 0;
    }
    else
    {
        GPIO_SetBits(GPIOC,GPIO_Pin_15);
        tt_den = 1;
    }
    delayUs(30);
}
void LCD_hienthi_DA1()
{
    LCD_HIEN THI_SO_Z_TOADO_XY(0,2,0); //D
    LCD_HIEN THI_SO_Z_TOADO_XY(1,5,0); //A
    LCD_HIEN THI_SO_Z_TOADO_XY(2,8,0); //-
    LCD_HIEN THI_SO_Z_TOADO_XY(3,11,0);//1
}

```

