

TÓM TẮT

Ngày nay trên thế giới với sự bùng nổ của các ngành công nghệ thông tin, điện tử đã làm cho đời sống của con người ngày càng hoàn thiện. Các thiết bị thông minh đã ngày càng được ứng dụng vào cuộc sống sinh hoạt hằng ngày của mỗi con người. Đặc biệt, smartphone đã trở thành một phần quen thuộc trong cuộc sống thường nhật của mỗi cá nhân và nhu cầu ứng dụng các ứng dụng của smartphone vào đời sống ngày càng thiết thực. Đề tài ứng dụng thực tế **“Thiết kế chế tạo quạt thông minh điều khiển qua điện thoại Android”** để nâng cao chất lượng cuộc sống con người và đáp ứng các nhu cầu ngày càng mạnh mẽ trong thời đại công nghệ số.

Quạt thông minh có thể giúp chúng ta làm các công việc điều khiển các chức năng của quạt như: Hệ thống sẽ có chức năng đo nhiệt độ và đặt lịch làm việc. Khi nhiệt độ của hệ thống cao hơn mức cho phép hoặc đến giờ hoạt động theo thời gian biểu, mạch điều khiển sẽ điều khiển quạt hoạt động. Tốc độ quạt của hệ thống sẽ quay nhanh hay quay chậm tùy thuộc theo môi trường xung quanh. Sau khi hệ thống đạt được đến nhiệt độ lý tưởng quạt sẽ tự ngắt. Khi mất điện, các thông số điều khiển vẫn được lưu lại để khi có điện hệ thống vẫn hoạt động bình thường.

Do vậy em lựa chọn chủ đề **“Thiết kế chế tạo quạt thông minh điều khiển qua điện thoại Android”** làm đề tài đồ án tốt nghiệp nhằm mục đích tìm hiểu, thiết kế, chế tạo ra được một sản phẩm thông minh.

Nội dung đề tài được chia làm ba chương như sau:

Chương 1: Tổng quan.

Chương 2: Cơ sở lý thuyết.

Chương 3: Xây dựng phần cứng phần mềm cho sản phẩm.

Xác nhận của giáo viên hướng dẫn

LỜI CAM ĐOAN

Tên tôi là: Trần Đạt Tuyền

Sinh viên lớp 65DCDT21– Chuyên ngành điện tử, viễn thông – Trường Đại học công nghệ giao thông vận tải, khóa học 2014 - 2018.

Tôi xin cam đoan số liệu và kết quả nghiên cứu trong đề án “*Thiết kế chế tạo quạt thông minh điều khiển qua điện thoại Android*” dưới sự hướng dẫn của Th.S. *Hoàn Thế Phương* là trung thực. Mọi sự giúp đỡ cho việc thực hiện đề án này đã được cảm ơn và các thông tin trích dẫn trong đề án đã được chỉ rõ nguồn gốc rõ ràng và được phép công bố.

Hà Nội, ngày tháng năm 2018

Sinh viên

Trần Đạt Tuyền

MỤC LỤC

LỜI MỞ ĐẦU	1
CHƯƠNG 1 .TỔNG QUAN	2
1.1. Lý do chọn đề tài.....	2
1.2. Mục tiêu của đề tài	2
1.3. Giới hạn và phạm vi của đề tài.....	3
1.4. Phương pháp nghiên cứu	3
CHƯƠNG 2 .CƠ SỞ LÝ THUYẾT.....	4
2.1. Sơ đồ tổng thể của hệ thống.....	4
2.2. Sơ đồ tổng quát của hệ thống	6
2.3. Lựa chọn linh kiện.....	7
2.3.1. <i>Khối vi điều khiển.</i>	7
2.3.2. <i>Khối nguồn.</i>	15
2.3.3. <i>Khối hiển thị và nút nhấn</i>	15
2.3.4. <i>Khối điều khiển công suất quạt.</i>	16
2.3.5. <i>Khối cảm biến</i>	18
2.3.6. <i>Khối giao tiếp.</i>	23
2.4. Sơ đồ nguyên lý của mạch.....	24
CHƯƠNG 3 .XÂY DỰNG PHẦN CỨNG PHẦN MỀM CHO SẢN PHẨM.	25
3.1. Xây dựng phần cứng.	25
3.1.1. <i>Sơ đồ mạch nguyên lý.</i>	25
3.1.2. <i>Hình ảnh mạch PCB.</i>	26
3.2. Xây dựng phần mềm.	26
3.2.1. <i>Xây dựng phần mềm nhúng.</i>	26
3.2.2. <i>Xây dựng phần mềm trên thiết bị android.</i>	29
3.3. <i>Kết quả mô phỏng.</i>	32

3.4. Đánh giá kết luận	33
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	36
PHỤ LỤC	37
TÀI LIỆU THAM KHẢO.....	38

DANH MỤC HÌNH ẢNH

Hình 2.1 Sơ đồ tổng thể hệ thống	4
Hình 2.2 Sơ đồ đặc tả.	5
Hình 2.3 Sơ đồ tổng quát của hệ thống.	6
Hình 2.4 Vi điều khiển STM32F103C8T6.....	7
Hình 2.5 Module STM32F103C8T6.	8
Hình 2.6 Thanh ghi trong dòng ARM.	11
Hình 2.7 Sơ đồ chân vi điều khiển STM32F103C8T6.....	13
Hình 2.8 Mạch nạp Stlink sử dụng để nạp cho vi điều khiển STM32F103C8T6.	14
Hình 2.9 Bộ nguồn AC220V to DC 5V.	15
Hình 2.10 LCD16x2	16
Hình 2.11 Opto quang MOC3020.	17
Hình 2.12 Triac MAC97A8.....	17
Hình 2.13 Cảm biến DHT11.	18
Hình 2.14 Module DHT11.	19
Hình 2.15 Quá trình thay đổi trên chân được ghi lại.	20
Hình 2.16 Quá trình bắt đầu đọc tín hiệu.	21
Hình 2.17 Thời gian chờ cảm biến ổn định.	21
Hình 2.18 Thời gian chờ để nhận dữ liệu tiếp theo.	21
Hình 2.19 Kết quả lần đo 1.....	22
Hình 2.20 Kết quả lần đo 2.....	22
Hình 2.21 Module Bluetooth HC06.	23
Hình 2.22 Sơ đồ nguyên lý của thiết bị.	24
Hình 3.1 Sơ đồ mạch nguyên lý	25
Hình 3.2 Mạch PCB một lớp.....	26

Hình 3.3 Giao diện cubeMX cấu hình chân sử dụng.	27
Hình 3.4 Giao diện cubeMX cấu hình tần số sử dụng.	27
Hình 3.5 Giao diện phần mềm keilC.	28
Hình 3.6 Một số đoạn code của sản phẩm.	30
Hình 3.7 Sản phẩm sau khi hoàn chỉnh.	30
Hình 3.8 Thiết bị được điều khiển bằng điện thoại android.	32
Hình 3.9 Thiết bị có được điều khiển qua điện thoại android.	33
Hình 3.10 Thông số hiển thị trên điện thoại.	34
Hình 3.11 Thông số thiết bị với chế độ bằng tay.	35
Hình 3.12 Thông số thiết bị với chế độ tự động đèn thông báo chế độ tự động.	35

DANH MỤC BẢNG BIỂU

Bảng 0.1. Thông số Vi điều khiển STM32F103C8T6	11
--	----

THUẬT NGỮ VIẾT TẮT

Thuật ngữ	Tiếng Anh	Tiếng Việt
A		
ADC	Analog to Digital Converter	Bộ chuyển đổi tương tự sang số
I		
IC	Integrated Circuit	Vi mạch
L		
LSB	Least significant bit	Bit có trọng số thấp nhất
M		
MSB	Most significant bit	Bit có trọng số cao nhất

LỜI MỞ ĐẦU

Ngày nay trên thế giới với sự bùng nổ của các ngành công nghệ thông tin, điện tử đã làm cho đời sống của con người ngày càng hoàn thiện. Các thiết bị thông minh đã ngày càng được ứng dụng vào cuộc sống sinh hoạt hằng ngày của mỗi con người. Đặc biệt, smartphone đã trở thành một phần quen thuộc trong cuộc sống thường nhật của mỗi cá nhân và nhu cầu ứng dụng các ứng dụng của smartphone vào đời sống ngày càng thiết thực.

Quạt thông minh có thể giúp chúng ta làm các công việc điều khiển các chức năng của quạt như: Hệ thống sẽ có chức năng đo nhiệt độ và đặt lịch làm việc. Khi nhiệt độ của hệ thống cao hơn mức cho phép hoặc đến giờ hoạt động theo thời gian biểu, mạch điều khiển sẽ điều khiển quạt hoạt động. Tốc độ quạt của hệ thống sẽ quay nhanh hay quay chậm tùy thuộc theo môi trường xung quanh. Sau khi hệ thống đạt được đến nhiệt độ lý tưởng quạt sẽ tự ngắt. Khi mất điện, các thông số điều khiển vẫn được lưu lại để khi có điện hệ thống vẫn hoạt động bình thường.

Dưới sự hướng dẫn của **Th.S. Hoàn Thế Phương** em đã mạnh dạn tìm hiểu về chủ đề “*Thiết kế chế tạo quạt thông minh điều khiển qua điện thoại Android*”. Để đạt được mục tiêu trên, đồ án gồm:

Chương 1: Tổng quan.

Chương 2: Cơ sở lý thuyết.

Chương 3: Xây dựng phần cứng phần mềm cho sản phẩm.

Do thời gian nghiên cứu ngắn, tài liệu tham khảo không nhiều và trình độ kiến thức còn hạn chế nên khó tránh khỏi sai sót, rất mong nhận được ý kiến đóng góp của thầy cô cùng toàn thể các bạn.

Em xin gửi lời cảm ơn sâu sắc tới **Th.S. Hoàn Thế Phương** đã nhiệt tình hướng dẫn, góp ý giúp em hoàn thành đồ án này. Em cũng xin chân thành cảm ơn toàn thể các thầy cô Trường đại học công nghệ giao thông vận tải đã giảng dạy và truyền đạt kiến thức quý báu trong thời gian em theo học tại trường.

Em xin chân thành cảm ơn!

CHƯƠNG 1 .TỔNG QUAN

1.1. Lý do chọn đề tài.

Ngày nay trên thế giới với sự bùng nổ của các ngành công nghệ thông tin, điện tử đã làm cho đời sống của con người ngày càng hoàn thiện. Các thiết bị thông minh đã ngày càng được ứng dụng vào cuộc sống sinh hoạt hằng ngày của mỗi con người. Đặc biệt, smartphone đã trở thành một phần quen thuộc trong cuộc sống thường nhật của mỗi cá nhân và nhu cầu ứng dụng các ứng dụng của smartphone vào đời sống ngày càng thiết thực. Đề tài ứng dụng **“Thiết kế chế tạo quạt thông minh điều khiển qua điện thoại “** mạnh mẽ trong thời đại công nghệ số.

Đặc biệt trong những thập niên gần đây cùng với sự phát triển của hệ thống thông minh, ngành tự động hóa đã phát triển và tạo ra bước ngoặt quan trọng trong lĩnh vực chế tạo quạt thông minh phục vụ nhu cầu ngày càng cao của con người trong đời sống. Tại Việt Nam đã bắt đầu có nhiều công ty chuyên lắp đặt quạt thông minh hoặc các thiết bị thông minh khác nhằm phục vụ đời sống thiết yếu của con người hàng ngày. Quạt thông minh có thể giúp chúng ta làm các công việc điều khiển các chức năng của quạt như: Hệ thống sẽ có chức năng đo nhiệt độ và đặt lịch làm việc. Khi nhiệt độ của hệ thống cao hơn mức cho phép hoặc đến giờ hoạt động theo thời gian biểu, mạch điều khiển sẽ điều khiển quạt hoạt động. Tốc độ quạt của hệ thống sẽ quay nhanh hay quay chậm tùy thuộc theo môi trường xung quanh. Sau khi hệ thống đạt được đến nhiệt độ lý tưởng quạt sẽ tự ngắt. Khi mất điện, các thông số điều khiển vẫn được lưu lại để khi có điện hệ thống vẫn hoạt động bình thường.

Xuất phát từ những thực tiễn nói trên, em quyết định thực hiện đề tài cho đồ án tốt nghiệp của mình: **“Thiết kế chế tạo quạt thông minh điều khiển qua điện thoại Android”**.

1.2. Mục tiêu của đề tài

- Tìm hiểu về lập trình vi điều khiển và ứng dụng.
- Thiết kế chế tạo quạt thông minh điều khiển qua điện thoại android.
- Vận hành hệ thống với chức năng:
 - + Có 2 chế độ điều khiển bằng tay và điều khiển tự động:
 - Điều khiển bằng tay là dùng các nút bấm trên điện thoại để tăng giảm tốc độ quạt.

- Điều khiển tự động theo môi trường xung quanh, tốc độ quạt tự động thay đổi theo nhiệt độ của môi trường.
- + Nút chuyển từ chế độ tự động sang chế độ bằng tay.
- + Cài đặt quạt hoạt động được ở các dải nhiệt độ khác nhau.

1.3. Giới hạn và phạm vi của đề tài

- Thiết kế chế tạo quạt thông minh điều khiển qua điện thoại android.
- Hệ thống cần đảm bảo một số yêu cầu chức năng điều khiển quạt cơ bản, tốc độ của quạt phụ thuộc vào môi trường ...

1.4. Phương pháp nghiên cứu

Về lý thuyết:

- Tổng hợp những kiến thức về lập trình vi điều khiển.
- Tổng hợp những kiến thức về lập trình phần mềm cho android.
- Tổng hợp những kiến thức về Thiết kế mạch điện điều khiển.
- Tổng hợp kiến thức về quạt thông minh điều khiển qua điện thoại android.

Về thực nghiệm:

- Thiết kế chế tạo quạt thông minh điều khiển qua điện thoại android.

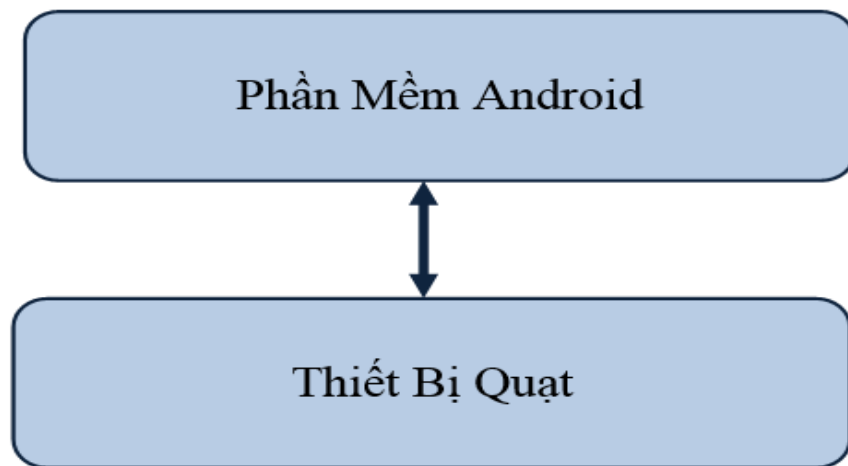
CHƯƠNG 2 .CƠ SỞ LÝ THUYẾT

2.1. Sơ đồ tổng thể của hệ thống

Hệ thống gồm hai phần

- Thiết bị quạt thông minh là một chiếc quạt có thể hiển thị thông báo nhiệt độ độ ẩm, có thể điều khiển các chế độ theo nhiệt độ , và có thể điều khiển bằng thiết bị khác thông qua bluetooth.
- Phần mềm điều khiển quạt trên điện thoại android, có thể điều khiển thiết bị quạt thông qua sóng bluetooth, có thể hiển thị các trạng thái của quạt , nhiệt độ độ ẩm mà cảm biến trên quạt đo được.

Sơ đồ tổng thể của hệ thống được thiết kế theo hình bên dưới:



Hình 2.1 Sơ đồ tổng thể hệ thống

Quạt thông minh có thể điều khiển với hai chế độ *Sơ đồ tổng thể hệ thống*. độ :

- Chế độ tự động: quạt có thể hoạt động theo nhiệt độ, có 3 thiết lập là low, med, và high, theo giá nhiệt độ low(27) med(28) high(30).
- Chế độ bằng tay, có các nút điều khiển các chế độ: low , med , High.

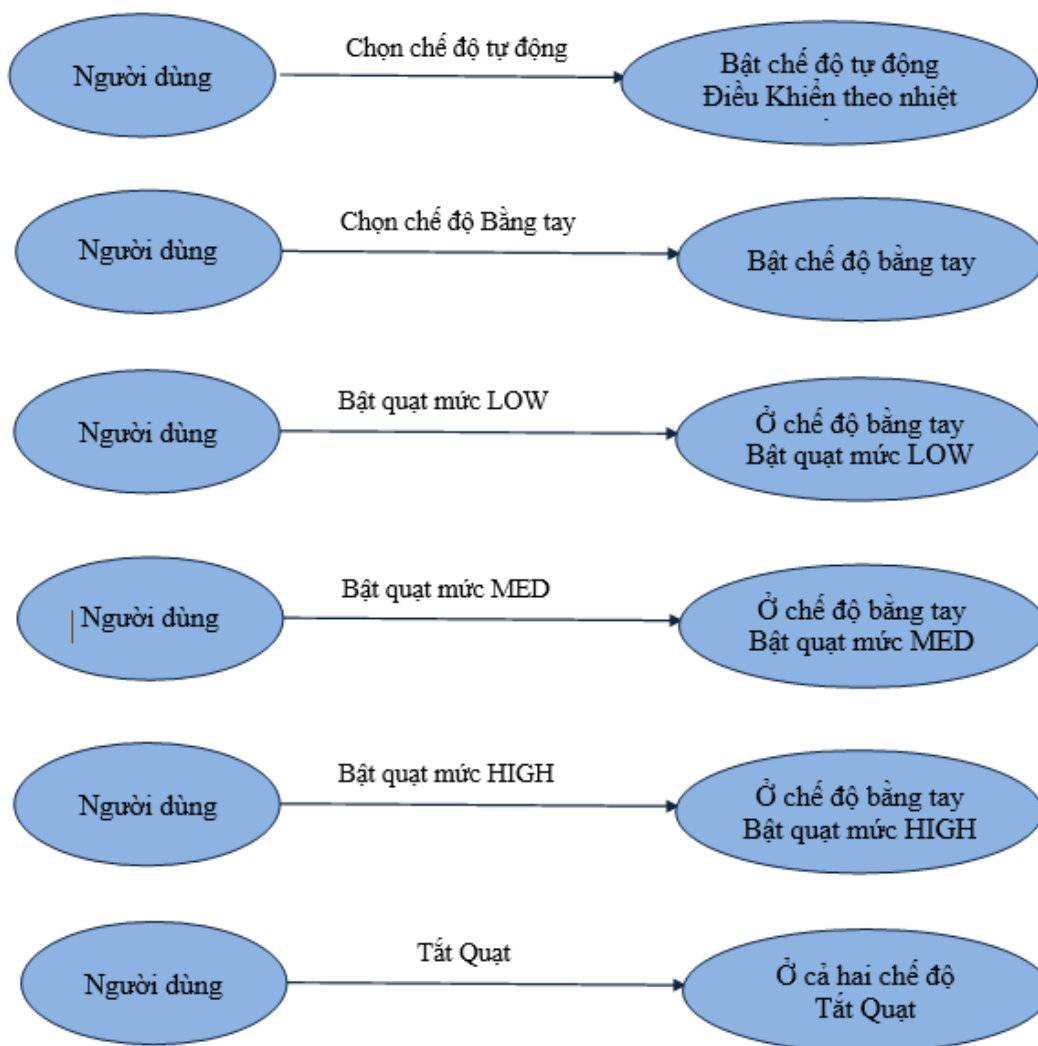
Quạt thông minh có nút nhấn để chuyển hai chế độ:

- Từ tự động qua bằng tay: khi ở chế độ tự động đèn thông báo chế độ sáng đỏ, khi nhấn nút khoảng thời gian nhỏ tầm 150ms thì quạt sẽ chuyển sang chế độ bằng tay , khi đó có thể dùng các nút low, med , high để điều khiển quạt.

- Từ chế độ bằng tay qua chế độ tự động: để đảm bảo an toàn thì chế độ tự động chỉ được bật lên khi quạt đang hoạt động, có nghĩa là quạt phải đang được bật ở một trong ba chế độ là low, med , high. Khi nhấn nút khoảng thời gian nhỏ 150ms thì quạt chuyển chế độ, chuyển sang chế độ tự động quạt điều khiển theo nhiệt độ đo được nhờ cảm biến trên quạt

Ở chế độ bằng tay có thể điều khiển quạt thông qua 3 nút nhấn low , med , high, và có thể tắt quạt bằng các giữ nút chuyển chế độ trong 1s.

Dưới đây là sơ đồ đặc tả.



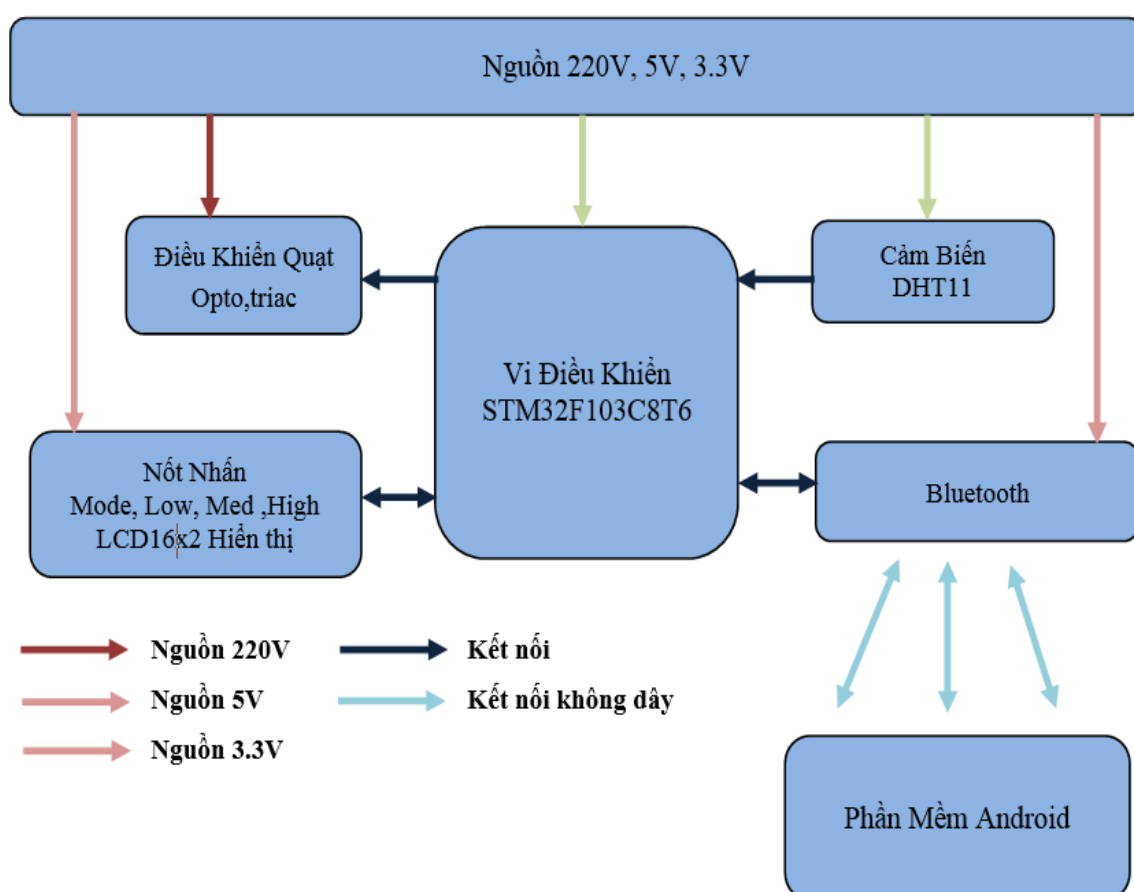
Hình 2.2 Sơ đồ đặc tả..

2.2. Sơ đồ tổng quát của hệ thống

Hệ thống được thiết kế theo các khối gồm :

- Khối nguồn 220V , 5V , 3.3V.
- Khối vi điều khiển Sử dụng vi điều khiển hãng STMicroelectronic.
- Khối nút nhấn, hiển thị gồm 4 nút mode , low, med , high, Lcd 16x2
- Khối điều khiển công suất quạt.
- Khối cảm biến nhiệt độ độ ẩm.
- Khối giao tiếp bluetooth.
- Phần mềm điều khiển qua android.

Dưới đây là sơ đồ tổng quát của hệ thống.



Hình 2.3 Sơ đồ tổng quát của hệ thống.

2.3. Lựa chọn linh kiện.

2.3.1. Khối vi điều khiển.

Vi điều khiển được lựa chọn của hãng STMicroelectronic là STM32F103C8T6 với cấu hình cao, độ ổn định đáp ứng vừa đủ, và giá thành phù hợp.

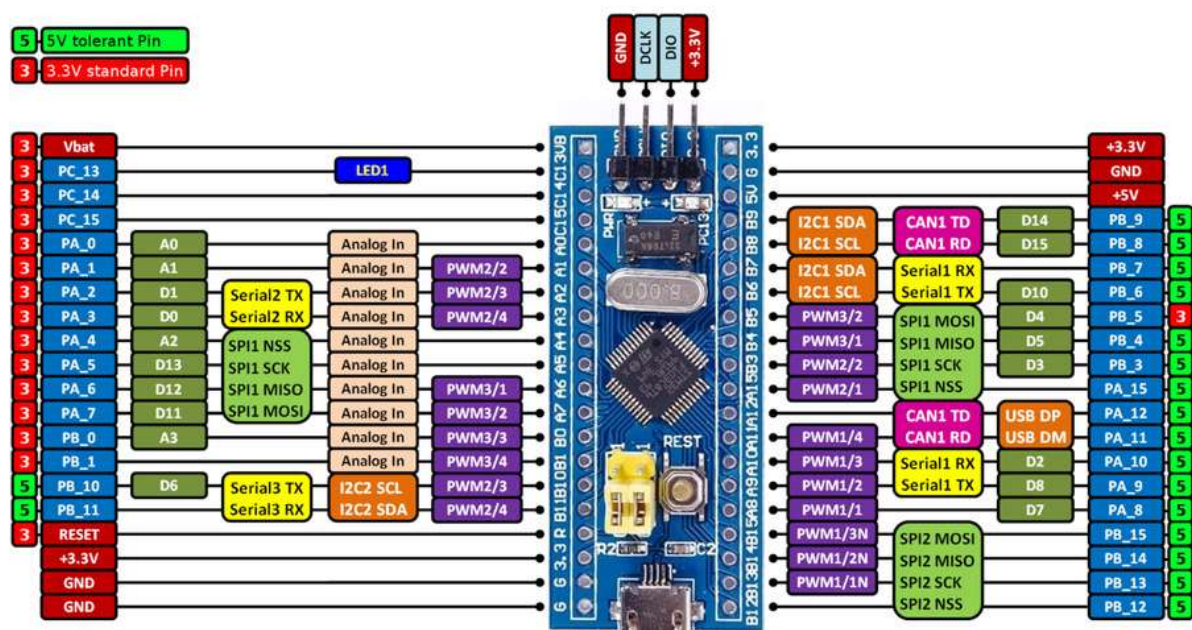
- Nguồn sử dụng cho vi điều khiển là nguồn 3.3V
- Có số lượng chân kết nối lớn, vi điều khiển có 48 chân tính cả chân nguồn.
- Xung nhịp tối đa cho nhân xử lý là 72MHz được cung cấp bằng cách nhân chia xung từ nguồn thạch anh ngoài 8MHz. Xung nhịp cho các bộ timer module ngoại vi lên tới 48MHz, một số bộ hỗ trợ lên tới 72MHz
- Có các thư viện và phần mềm hỗ trợ lập trình đơn giản như Keil C và bộ thư viện do Keil cung cấp.

Hình ảnh vi điều khiển sử dụng:



Hình 2.4 Vi điều khiển STM32F103C8T6.

Do vi điều khiển sử dụng có chân nhỏ nên lựa chọn module để có thể thiết kế mạch dễ dàng nhất. Module rất thông dụng, giá thành phù hợp. Hình ảnh module sử dụng:



Hình 2.5 Module STM32F103C8T6.

2.3.1.1. Giới thiệu về dòng Vi điều khiển ARM.

Việc thiết kế ARM được bắt đầu từ năm 1983 trong một dự án phát triển của công ty máy tính Acorn. Nhóm thiết kế, dẫn đầu bởi Roger Wilson và Steve Furber, bắt đầu phát triển một bộ vi xử lý có nhiều điểm tương đồng với kỹ thuật MOS 6502 tiên tiến. Acorn đã từng sản xuất nhiều máy tính dựa trên 6502, vì vậy việc tạo ra một Vi xử lý như vậy là một bước tiến đáng kể của công ty này. Nhóm thiết kế hoàn thành việc phát triển mẫu gọi là ARM1 vào năm 1985, và vào năm sau, nhóm hoàn thành sản phẩm ARM 2. ARM 2 có đường truyền dữ liệu 32bit, không gian địa chỉ 26bit tức cho phép quản lý đến 64 Mbyte địa chỉ và 16 thanh ghi 32 bit. Một trong những thanh ghi này đóng vai trò là bộ đếm chương trình với 6bit có giá trị cao nhất và 2bit có giá trị thấp nhất lưu giữ các cờ trạng thái của bộ vi xử lý. Thế hệ sau, ARM 3 được tạo ra với 4KB bộ nhớ đệm và có chức năng được cải thiện tốt hơn nữa. Vào những năm cuối thập niên 80, hãng máy tính Apple Computer bắt đầu hợp tác với Acorn để phát triển các thế hệ lõi ARM mới. Công việc này trở nên quan trọng đến nỗi Acorn nâng nhóm thiết kế trở thành một công ty mới gọi là Advanced RISC Machines. Từ lý do đó hình thành chữ viết tắt ARM của Advanced RISC Machines thay vì Acorn RISC Machine. Về sau, Advanced RISC Machines trở thành công ty ARM Limited. Kết quả sự hợp tác

này là ARM6. Mẫu đầu tiên được công bố vào năm 1991 và Apple đã sử dụng bộ vi xử lý ARM 610 dựa trên ARM6 làm cơ sở cho PDA hiệu Apple Newton. Vào năm 1994, Acorn dùng ARM 610 làm CPU trong các máy vi tính RiscPC của họ. 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 sao cho 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. Thế hệ khá thành công của hãng là lõi xử lý ARM7TDMI, với hàng trăm triệu lõi được sử dụng trong các máy điện thoại di động, hệ thống video game cầm tay ARM đã thành một thương hiệu đứng đầu thế giới về các ứng dụng sản phẩm nhúng đòi hỏi tính năng cao, sử dụng năng lượng ít và giá thành thấp. Chính nhờ sự nổi trội về thị phần đã thúc đẩy ARM liên tục được phát triển và cho ra nhiều phiên bản mới. Những thành công quan trọng trong việc phát triển ARM: Giới thiệu ý tưởng về định dạng các tập lệnh được nén lại (Thumb) cho phép tiết kiệm năng lượng và giảm giá thành ở những hệ thống nhỏ. Các ứng dụng cho hệ thống nhúng dựa trên lõi xử lý ARM ngày càng trở nên rộng rãi. Hầu hết các nguyên lý của hệ thống trên Vi xử lý và cách thiết kế bộ xử lý hiện đại được sử dụng trong ARM, ARM còn đưa ra một số khái niệm mới như giải nén động các dòng lệnh. Việc sử dụng ba trạng thái nhận lệnh giải mã thực thi trong mỗi chu kỳ máy mang tính quy phạm để thiết kế các hệ thống xử lý thực. Do đó, lõi xử lý ARM được sử dụng rộng rãi trong các hệ thống phức tạp.

Các đặc tính của ARM bao gồm:

- Cấu trúc nạp / lưu trữ.
- Hỗ trợ tập lệnh trực giao.
- Thanh ghi lớn.
- Hầu hết các lệnh được thực hiện trong 1 chu kỳ CPU.
- Chiều dài mã máy cố định, do đó dễ dàng thực hiện đường ống hóa (pipeline).

2.3.1.2. Mô hình thiết kế ARM

- Program Counter (PC): Bộ đếm chương trình: giữ địa chỉ của lệnh hiện tại.
- Thanh ghi tích lũy (ACC): giữ giá trị dữ liệu khi đang làm việc.

- Đơn vị xử lý số học (ALU): thực thi các lệnh nhị phân như cộng, trừ, gia tăng...
- Thanh ghi lệnh (IR): giữ tập lệnh hiện tại đang thực thi.

Lỗi xử lý MU0 được phát triển đầu tiên và là lỗi xử lý đơn giản, có tập lệnh dài 16-bit, với 12-bit địa chỉ và 4bit mã hóa. Cấu trúc tập lệnh lỗi MU0 có dạng: Mô hình thiết kế đường truyền dữ liệu đơn giản của lỗi xử lý MU0 được mô tả trong Hình 2.11. Việc thiết kế ở cấp chuyển đổi mức thanh ghi (RTL): Bộ đếm chương trình (PC) chỉ đến tập lệnh cần thực thi, nạp vào thanh ghi lệnh (IR), giá trị chứa trong IR chỉ đến vùng địa chỉ ô nhớ, nhận giá trị, kết hợp với giá trị đang chứa trong thanh ghi tích lũy (ACC) qua đơn vị xử lý số học (ALU) để tạo giá trị mới, chứa vào ACC. Mỗi một lệnh như vậy, tùy vào số lần truy cập ô nhớ mà tốn số chu kỳ xung nhịp tương đương. Sau mỗi lệnh thực thi, PC sẽ được tăng thêm.

Để phục vụ mục đích của người dùng: r0 ÷ r14 là 15 thanh ghi đa dụng, r15 là thanh ghi Program Counter (PC), thanh ghi trạng thái chương trình hiện tại (CPSR-Current Program Status Register). Các thanh ghi khác được giữ lại cho hệ thống (như thanh ghi chứa các ngắt).

- Thanh ghi CPSR được người dùng sử dụng chứa các bit điều kiện.
- N: Negative- cờ này được bật khi bit cao nhất của kết quả xử lý ALU bằng 1.
- Z: Zero- cờ này được bật khi kết quả cuối cùng trong ALU bằng 0.
- V: Overflow-cờ báo tràn sang bit dấu.
- Thanh ghi SPSR (Save Program Status Register) dùng để lưu giữ trạng thái của thanh ghi CPSR khi thay đổi chế độ
- C: Carry - cờ này được bật khi kết quả cuối cùng trong ALU lớn hơn giá trị 32-bit và tràn.



Hình 2.6 Thanh ghi trong dòng ARM.

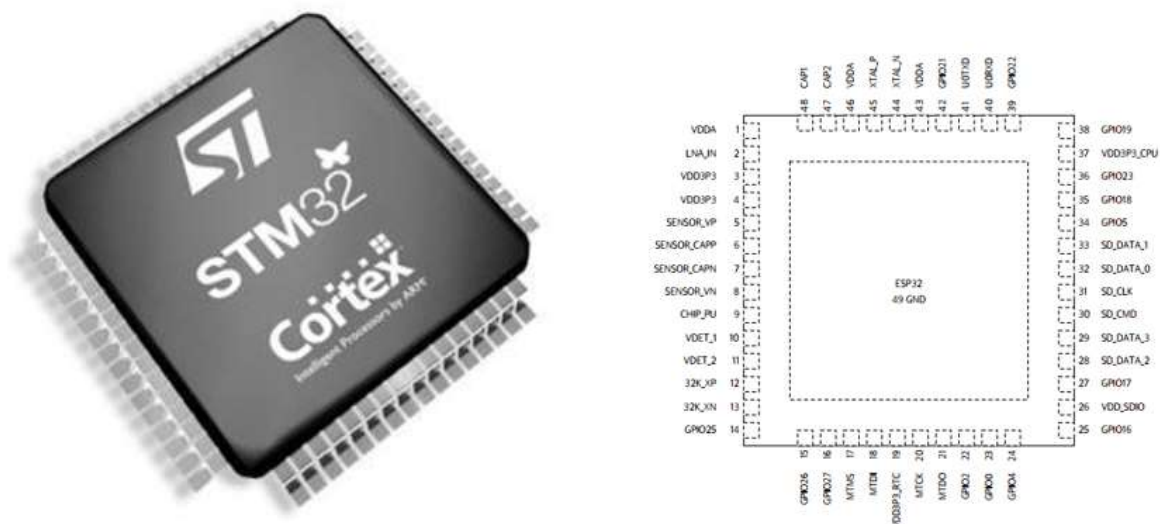
2.3.1.3. Vi điều khiển STM32F103C8T

Thông số kỹ thuật của Vi điều khiển STM32F103C8T được mô tả như trong Bảng 2.1 dưới đây:

Bảng 2.1. Thông số Vi điều khiển STM32F103C8T6

Loại	Miêu tả
Tốc độ Vi xử lý	Tốc độ có thể lên tới 72MHZ, 1.25 DMIPS/MHz (Dhrystone 2.1)
Bộ nhớ	<ul style="list-style-type: none"> 64 or 128 Kbytes of Flash memory 20 Kbytes of SRAM
Xung đồng hồ, Điện áp nguồn	<ul style="list-style-type: none"> 2.0 to 3.6 V application supply and I/Os POR, PDR, and programmable voltage detector (PVD) 4-to-16 MHz crystal oscillator Internal 8 MHz factory-trimmed RC

	<ul style="list-style-type: none"> • Internal 40 kHz RC • PLL for CPU clock • 32 kHz oscillator for RTC with calibration
Chế độ tiết kiệm năng lượng	<ul style="list-style-type: none"> • Step – chế độ ngủ • Stop – dừng không hoạt động • Standby – Ngủ tạm thời
Ngoại vi	<ul style="list-style-type: none"> • 16 kênh ADC 12bit • 7 kênh DMA • SPI • I2c • UART (baudrate lên tới 115200) • Hỗ trợ ngắt ngoài tất cả các chân • CAN, LIN • USB 2.0
Debug	Serial wire debug (SWD) & JTAG
Đồng hồ	<ul style="list-style-type: none"> • 3 timer 16-bit dùng để đếm hoặc tính xung của encoder bấm xung với PWM • chương trình watchdog timer dùng reset khi treo • 1 đồng hồ hệ thống chạy trên core 24bit



Hình 2.7 Sơ đồ chân vi điều khiển STM32F103C8T6.

Giống như các dòng vi điều khiển khác ARM cũng có thể lập trình bằng assembly hoặc bằng C. Để dễ dàng lập trình thì trong khóa luận này, ngôn ngữ C được sử dụng để viết chương trình. Trên thị trường cũng có nhiều mã nguồn mở dùng để viết các chương trình cho arm. Trong số đó có keilc là 1 trong những compiler được nhiều người yêu thích vì tính đa dạng và hỗ trợ nhiều dòng vi điều khiển khác nhau. Giao diện cũng dễ nhìn và dễ sử dụng. Phần mềm cũng hỗ trợ các công cụ debug với dòng arm như ST-link. Chúng ta có thể dễ dàng tìm mua ở các cửa hàng linh kiện điện tử.

Phần mềm KeilC arm V5 hỗ trợ đầy đủ các dòng Vi xử lý của arm. Cung cấp các thư viện và hàm cơ bản cho thanh ghi. Quá trình cho người mới bắt đầu keilc cũng có một vài ví dụ cho người đọc hiểu chức năng của các tác dụng của các hàm. Ngoài ra chương trình còn hỗ trợ nhiều công cụ debug ngoài ST-link. Khi lập trình nếu sai về định dạng code ngay lập tức phần mềm sẽ báo lại mà chưa cần build. Điều này sẽ giúp cảnh báo cho người sử dụng những lỗi không cần thiết. Giao diện dễ dung là một điểm cộng cho công cụ lập trình này. Ngoài phần mềm KeilC ra chúng ta có thể sử dụng IAR, coscos code ide... Làm trình biên dịch code. Chúng ta cũng có thể sử dụng assembly là ngôn ngữ lập trình hoặc nạp trực tiếp file.hex hoặc.bin vào bộ nhớ của vi xử lý thông qua các giao tiếp UART tạo các bootloader cho vi xử lý.

2.3.1.4. Mạch nạp sử dụng.

Mạch nạp sử dụng là mạch nạp Stlink rất nhỏ gọn và dễ dàng có được. Với kết nối đơn giản gồm 4 chân 3.3V, GND, DCLK, DIO.

Hỗ trợ phần mềm như:

- ST-Link Utility 2.0 hoặc phiên bản cao hơn.
- STVD 2 hoặc phiên bản cao hơn.
- IAR EWARM V6.20
- Keil RVMDK v4.21

Hỗ trợ phần mềm tự động và chắc chắn rằng được hãng ST hỗ trợ. Khi hãng nâng cấp phần mềm cao nhất là V2

Hình ảnh mạch nạp:



Hình 2.8 Mạch nạp Stlink sử dụng để nạp cho vi điều khiển STM32F103C8T6.

2.3.2. Khối nguồn.

Khối nguồn cần cung cấp 3 loại nguồn AC220V , DC5V, DC3.3V. Vì vậy sử dụng một bộ chuyển đổi nguồn AC220 xuống DC5V. Hình ảnh bộ chuyển đổi nguồn.



Hình 2.9 Bộ nguồn AC220V to DC 5V.

Nguồn DC5V sử dụng cung cấp cho module STM32F103C8T6 và khối kết nối bluetooth.

Trong module STM32F103C8T6 có hỗ trợ một bộ chuyển đổi nguồn DC5V xuống DC3.3V do đó có thể sử dụng luôn để cung cấp cho chip vi điều khiển , và cảm biến nhiệt độ độ ẩm.

2.3.3. Khối hiển thị và nút nhấn

Khối nút nhấn gồm 4 nút nhấn, sử dụng loại nút nhấn nhà 4 chân.

Phần hiển thị được sử dụng LCD16x2 với kết nối gồm 7 chân , 3 chân điều khiển và 4 chân dữ liệu. Lcd16x2 có nhiều thư viện mà STM có thể sử dụng được để giao tiếp. do đó việc giao tiếp hiển thị dễ dàng. Hình ảnh LCD 16x2.

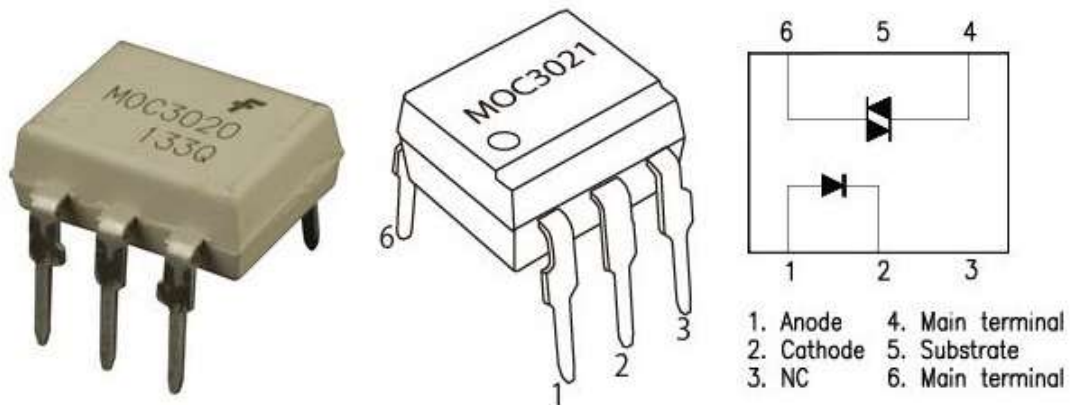


Hình 2.10 LCD16x2

2.3.4. Khối điều khiển công suất quạt.

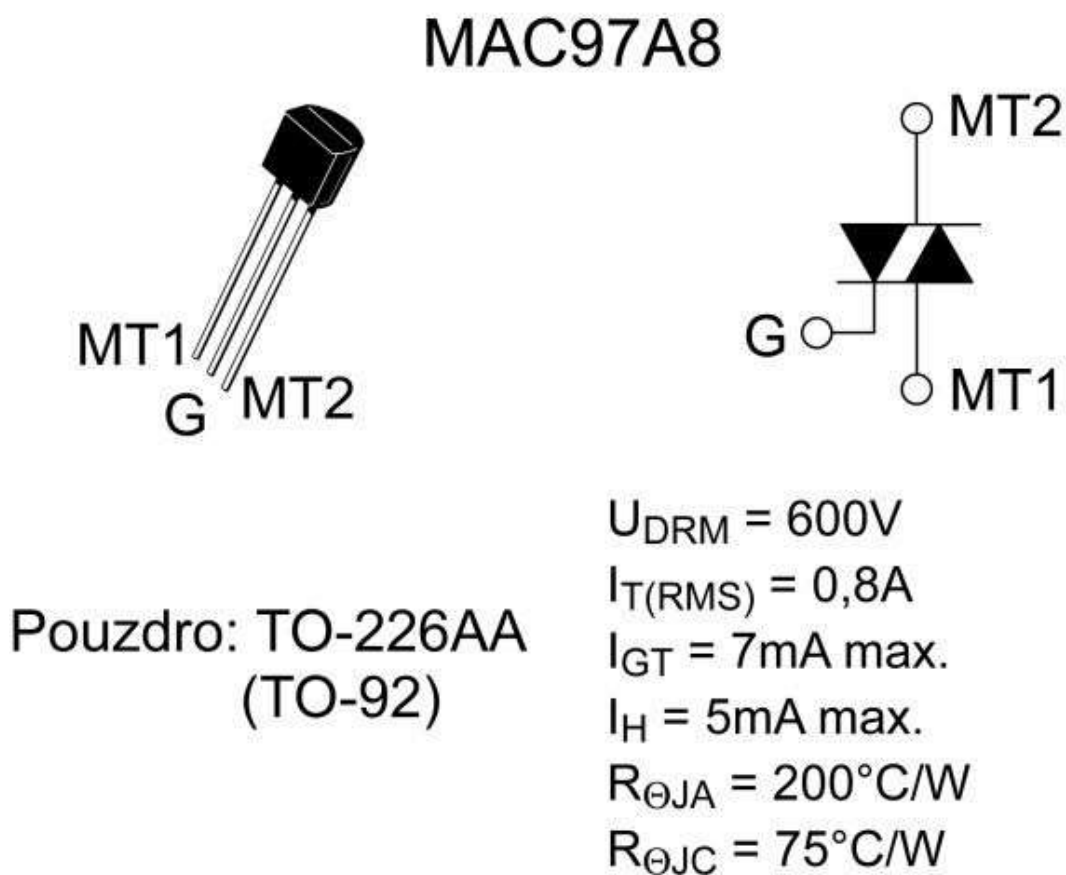
Khối điều khiển công suất quạt sử dụng opto quang và triac để điều khiển dòng điện xoay chiều 220V cấp cho phần công suất quạt. quạt điện sử dụng là loại quạt có 3 chế độ (3 cuộn dây) do đó cần 3 bộ điều khiển công suất. Việc sử dụng opto quang và triac nhằm mục đích giảm thiểu nhiễu gây ra từ bộ phận công suất lên vi điều khiển.

Opto quang sử dụng là loại dùng để điều khiển triac với tên : MOC3020 . Dưới đây là hình ảnh MOC3020.



Hình 2.11 Opto quang MOC3020.

Triac sử dụng là loại triac điều khiển được giải điện áp ở mức 600VAC và dòng điều khiển ổn định ở mức 0.8A . Lựa chọn MAC97A8. Hình ảnh triac MAC97A8:



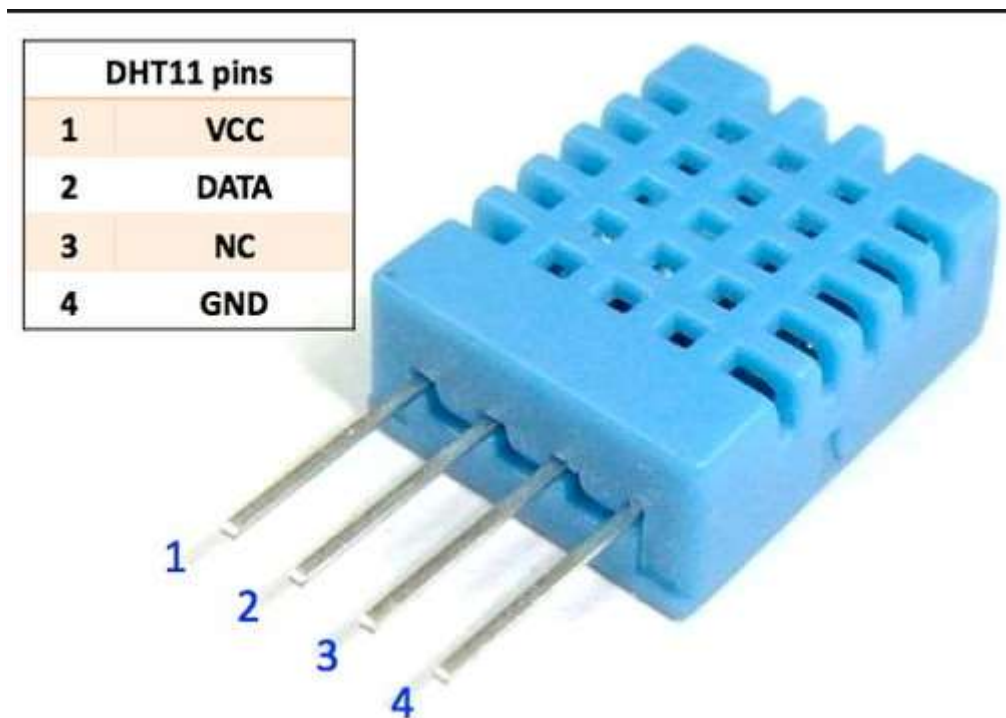
Hình 2.12 Triac MAC97A8.

2.3.5. Khối cảm biến

Khối cảm biến có thể đo được nhiệt độ và độ ẩm. DHT11 có tích hợp hai thông số đo trên một cảm biến duy nhất, việc giao tiếp với DHT11 cũng dễ dàng. Trên internet có nhiều thư viện hỗ trợ giao tiếp do đó việc kết nối cũng dễ dàng.

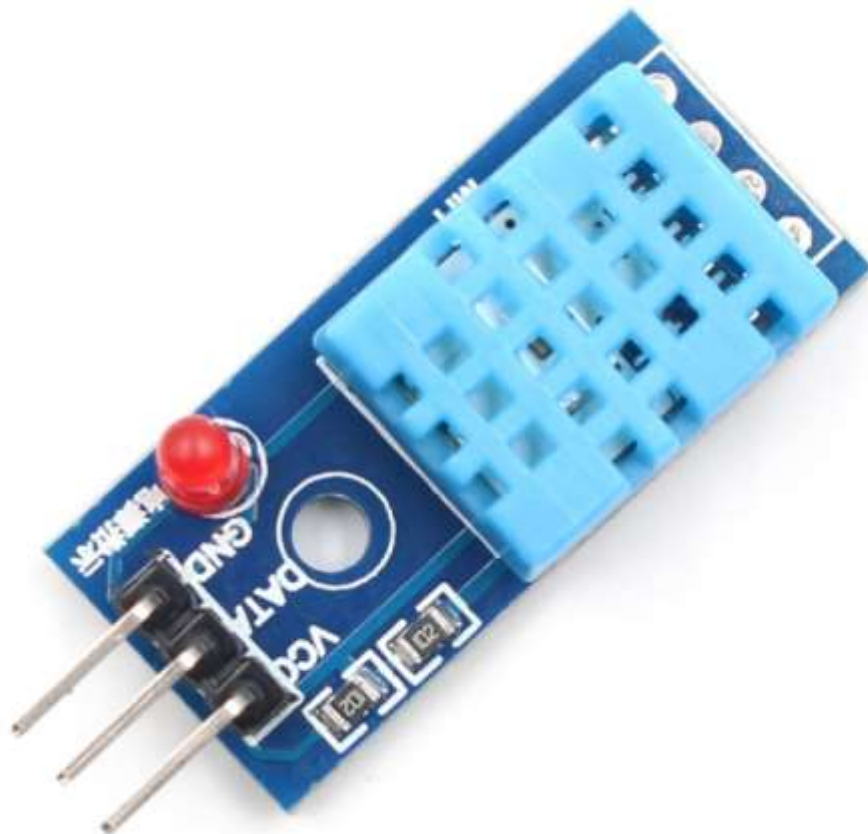
DHT11 sử dụng giao tiếp Single-Wire Two-Way là giao tiếp trên một đường tín hiệu, dữ liệu truyền và nhận được dựa vào độ dài xung xuống hay lên để tính ra giá trị nhiệt độ độ ẩm. Dữ liệu nhận về mỗi lần đọc dữ liệu gồm 40bit trong đó 16 bit nhiệt độ , 16 bit độ ẩm , 8bit kiểm tra tổng (hai dữ liệu nhiệt độ và độ ẩm được chia thành 4 khối mỗi khối 8 bit 4 khối này được cộng thập phân với nhau và so sánh với 8 bit cuối cùng)

Hình ảnh cảm biến DHT11.



Hình 2.13 Cảm biến DHT11.

Sử dụng module DHT11 với kết nối dễ dàng hơn, trong module có điện trở kéo chân data lên nguồn VCC để ghim điện áp. Và led báo nguồn. Hình ảnh module DHT11 sử dụng:



Hình 2.14 Module DHT11.

2.3.5.1. Tổng Quan.

DHT11 cảm biến nhiệt độ và độ ẩm kỹ thuật số là một bộ cảm biến composite có chứa hiệu chuẩn đầu ra tín hiệu số của nhiệt độ và độ ẩm. Dữ liệu được đọc trên 1 đường bus duy nhất. Thiết bị thu sẽ gửi 1 số dữ liệu xuống để có thể lấy lại được dữ liệu về nhiệt độ độ ẩm. Cảm biến được sử dụng khác rộng rãi. Ứng dụng đa dạng trong cách sản phẩm. Sản phẩm cũng có những yêu cầu bảo đảm về chất lượng sai số khi đo.

2.3.5.2. Thông số kỹ thuật

Thông số độ ẩm.

- Data: 16Bit
- Độ sai số: $\pm 1\%$ RH
- Điều kiện môi trường: At $25^{\circ}\text{C} \pm 5\%$ RH
- Thời gian lấy mẫu: $1/e$ (63%) of 25°C 6s
- Độ trễ: $< \pm 0.3\%$ RH

Thông số nhiệt độ.

- Data: 16Bit
- Sai số: $\pm 0.2^{\circ}\text{C}$
- Điều kiện môi trường: At $25^{\circ}\text{C} \pm 2^{\circ}\text{C}$
- Thời gian phản hồi: $1 / e$ (63%) 10S

Chức năng các chân của cảm biến.

Chân 1: Nguồn 3 ~5.5 V DC

Chân 2: Dữ liệu

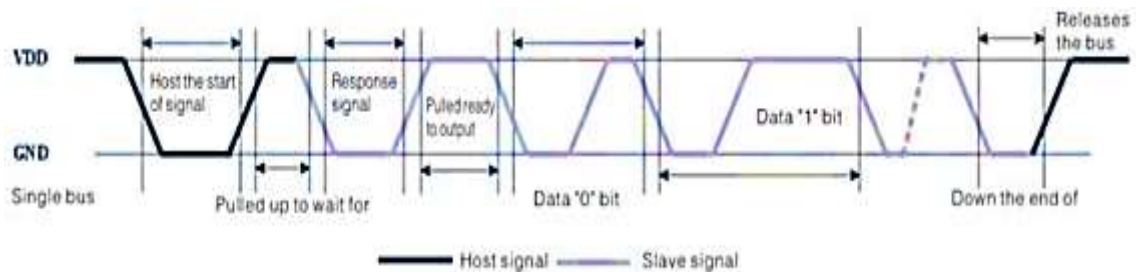
Chân 3: Không sử dụng

Chân 4: Chân đất GND.

2.3.5.3. Dữ liệu đầu ra

DHT11 gửi dữ liệu nối tiếp trên 1 bus duy nhất. Mỗi khi vi điều khiển gửi 1 tín hiệu yêu cầu bắt đầu đọc dữ liệu thì cảm biến DHT11 sẽ phản hồi dữ liệu cũng trên đường bus đó. Khi đọc dữ liệu phải để có treo trở để xác định rõ tín hiệu. Nếu khi truyền tín hiệu lỗi cảm biến cũng sẽ không thể trả về data.

Kết nối với vi điều khiển.

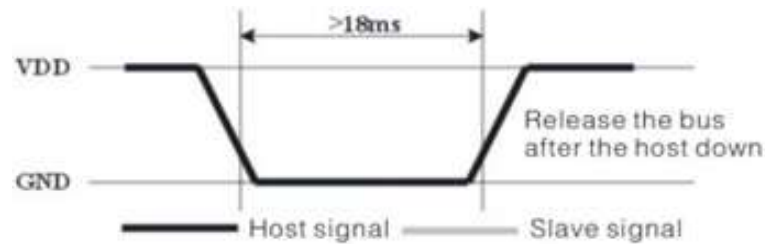


Hình 2.15 Quá trình thay đổi trên chân được ghi lại.

Bước 1: Sau khi cắm nguồn DHT11 (DHT11 sau khi cắm nguồn đợi 1s qua trạng thái không ổn định trong khoảng thời gian này không thể gửi bất kỳ dữ liệu nào), nhiệt độ môi trường thử nghiệm và dữ liệu độ ẩm, và ghi dữ liệu, trong khi DHT11 các dòng dữ liệu dữ liệu kéo bởi điện trở kéo lên đã được duy trì mức cao; các DHT11 DATA pin là ở trạng thái đầu vào, thời điểm phát hiện các tín hiệu bên ngoài.

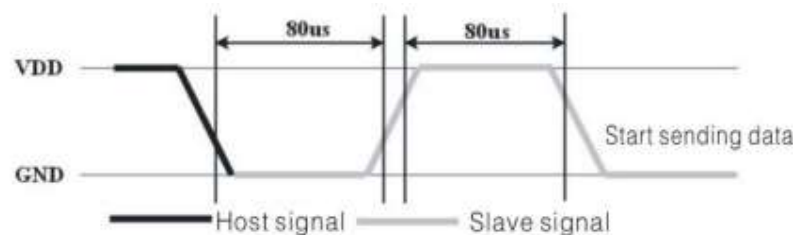
Bước 2: Vi xử lý thiết lập là tín hiệu ra ở mức thấp, và thời gian giữ thấp không thể được ít hơn 18ms, sau đó vi xử lý được đặt thành trạng thái đầu vào, do điện trở

kéo lên, một bộ vi xử lý I/ O DHT11 các dòng dữ liệu DATA cũng sẽ cao, chờ đợi DHT11 để trả lời tín hiệu, gửi các tín hiệu như được hiển thị.



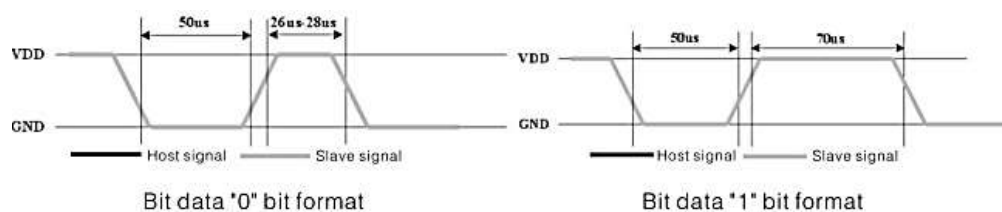
Hình 2.16 Quá trình bắt đầu đọc tín hiệu.

Bước 3: Chân DATA được phát hiện với một tín hiệu bên ngoài của DHT11 thấp, chờ đợi tín hiệu bên ngoài thấp kết thúc. Dừng DHT11 DATA pin ở trạng thái đầu ra với mức thấp trong khoảng thời gian 80 microseconds như là tín hiệu phản hồi, Tiếp theo là đầu ra của 80 micro giây của thiết bị ngoại vi thông báo mức điện áp đã sẵn sàng để nhận dữ liệu, Vi xử lý tại thời điểm này ở trạng thái đầu vào được phát hiện các I / O thấp (tín hiệu phản ứng DHT11), chờ đợi 80 microseconds nhận và gửi tín hiệu highdata như được hiển thị.



Hình 2.17 Thời gian chờ cảm biến ổn định.

Bước 4: Đầu ra của DHT11 pin DATA 40, bộ vi xử lý nhận được 40bit dữ liệu của dữ liệu "0" định dạng mức thấp là 50 microsecond và 26-28 microsecond theo sự thay đổi cấp độ cấp I / O, bit dữ liệu "1" định dạng: mức cao ở mức thấp cộng, 50 microseconds đến 70 microseconds. Dữ liệu bit "0", "1" tín hiệu



Hình 2.18 Thời gian chờ để nhận dữ liệu tiếp theo.

Kết quả lần đo 1.

<u>0011 0101</u>	<u>0000 0000</u>	<u>0001 1000</u>	<u>0000 0000</u>	<u>0100 1101</u>
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

Hình 2.19 Kết quả lần đo 1.

- 8 bit cao độ ẩm: 0011 0101
- 8 bit thấp độ ẩm: 0000 0000
- 8 bit cao nhiệt độ: 0001 1000
- 8 bit thấp nhiệt độ: 0100 1101
- 8 bit kiểm tra: 0100 1101

Đầu tiên chúng ta kiểm tra dữ liệu nhận được đã đúng chưa ta cộng 32bit đầu tiên lại với nhau

$$0011\ 0101 + 0000\ 0000 + 0001\ 1000 + 0100\ 1101 = 0100\ 11\ 01$$

Dữ liệu có tổng bằng với 8bit kiểm tra. Vì vậy dữ liệu nhận được là đúng chúng ta tiếp tục lấy của nhiệt độ và độ ẩm

- Độ ẩm = $0011\ 0101 + 0000\ 0000 = 0011\ 0101 = 35\ H = 53\% \text{ RH}$
- Nhiệt độ = $0001\ 1000 + 0000\ 0000 = 0001\ 1000 = 18\ H = 24^\circ \text{C}$

Kết quả lần đo 2.

<u>0011 0101</u>	<u>0000 0000</u>	<u>0001 1000</u>	<u>0000 0000</u>	<u>0100 1001</u>
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

Hình 2.20 Kết quả lần đo 2.

- 8bit cao độ ẩm: 0011 0101
- 8bit thấp độ ẩm: 0000 0000
- 8bit cao nhiệt độ: 0001 1000
- 8bit thấp nhiệt độ: 000 0000
- 8bit kiểm tra: 0100 1001

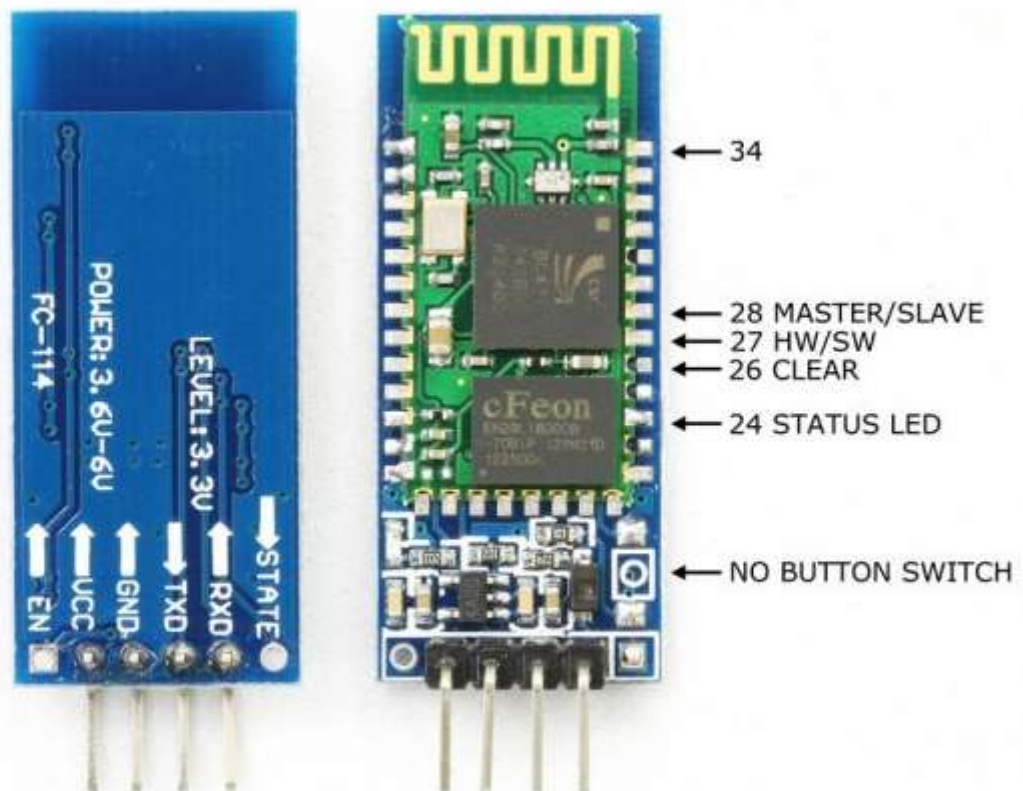
Tính toán.

$$0011\ 0101 + 0000\ 0000 + 0001\ 1000 + 0000\ 0000 = 01001101$$

Dữ liệu khác với 8bit kiểm tra điều này là sai vì vậy data nhận được là không đúng chúng ta không được lấy được.

2.3.6. *Khởi giao tiếp.*

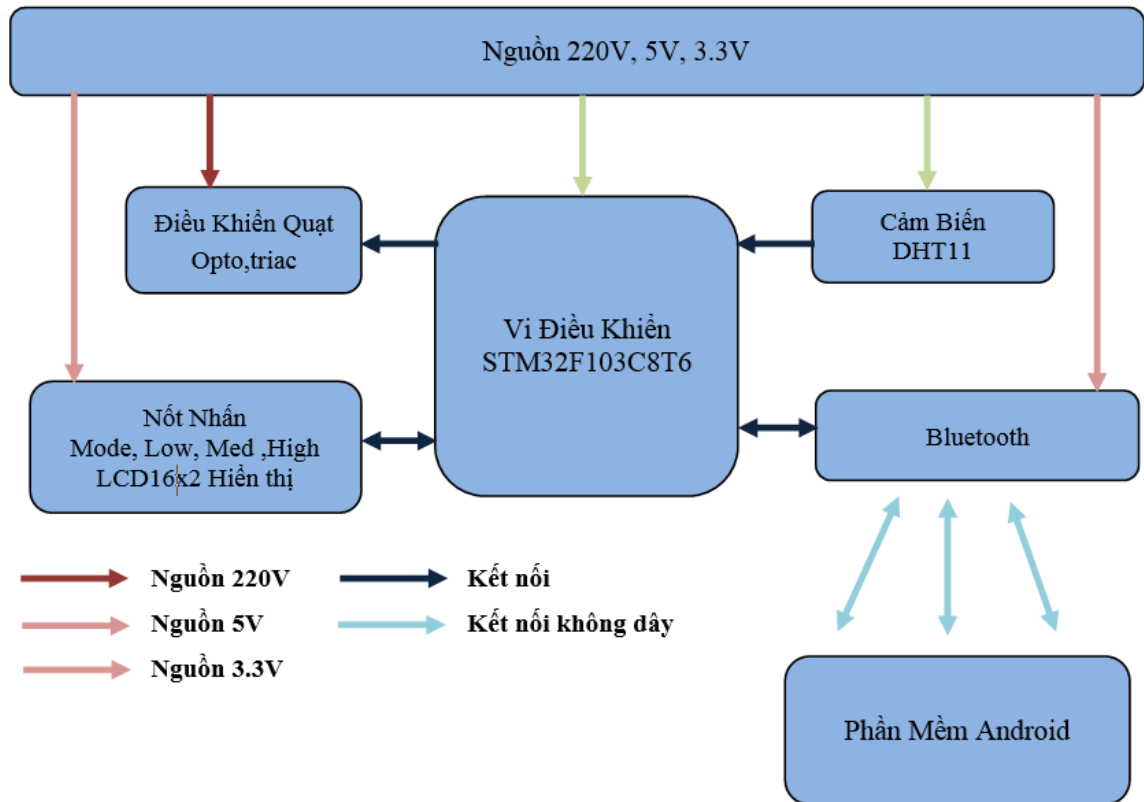
Lựa chọn kết nối bluetooth để kết nối với thiết bị android dễ dàng nhất, Module lựa chọn là bluetooth HC06, với kết nối dễ dàng. HC06 có thể giao tiếp với vi điều khiển thông qua cổng UART. Khi kết nối Vi điều khiển có thể sử dụng ở các baurate từ 200 đến 115200, sử dụng nhiều nhất ở 9600. HC06 sử dụng nguồn 5V và điện áp tham chiếu chân ở dải 3.3V phù hợp với vi điều khiển đang sử dụng. Hình ảnh Module Bluetooth:



Hình 2.21 Module Bluetooth HC06.

2.4. Sơ đồ nguyên lý của mạch.

Sơ đồ nguyên lý của mạch điều khiển thể hiện trong hình dưới.



Hình 2.22 Sơ đồ nguyên lý của thiết bị.

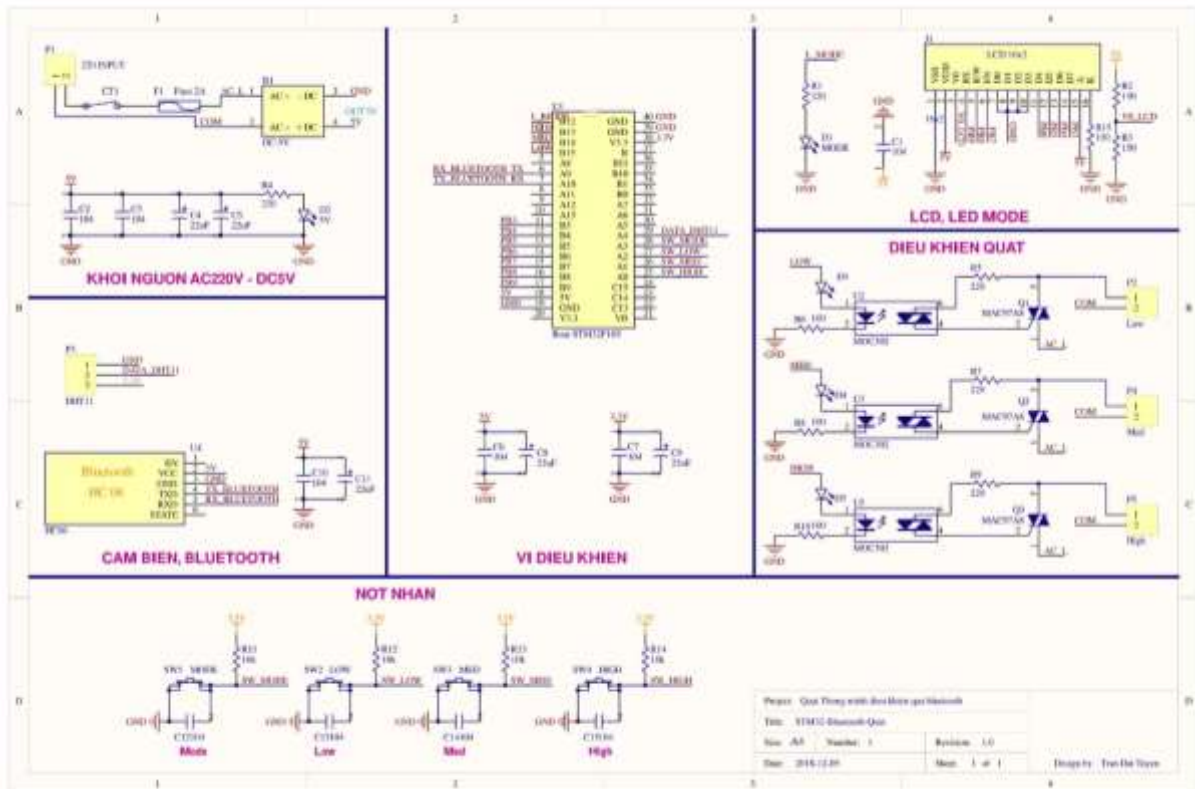
CHƯƠNG 3 .XÂY DỰNG PHẦN CỨNG PHẦN MỀM CHO SẢN PHẨM.

3.1. Xây dựng phần cứng.

Mạch điện được thiết kế trên phần mềm Altium , là phần mềm chuyên dụng để thiết kế mạch điện

3.1.1. Sơ đồ mạch nguyên lý.

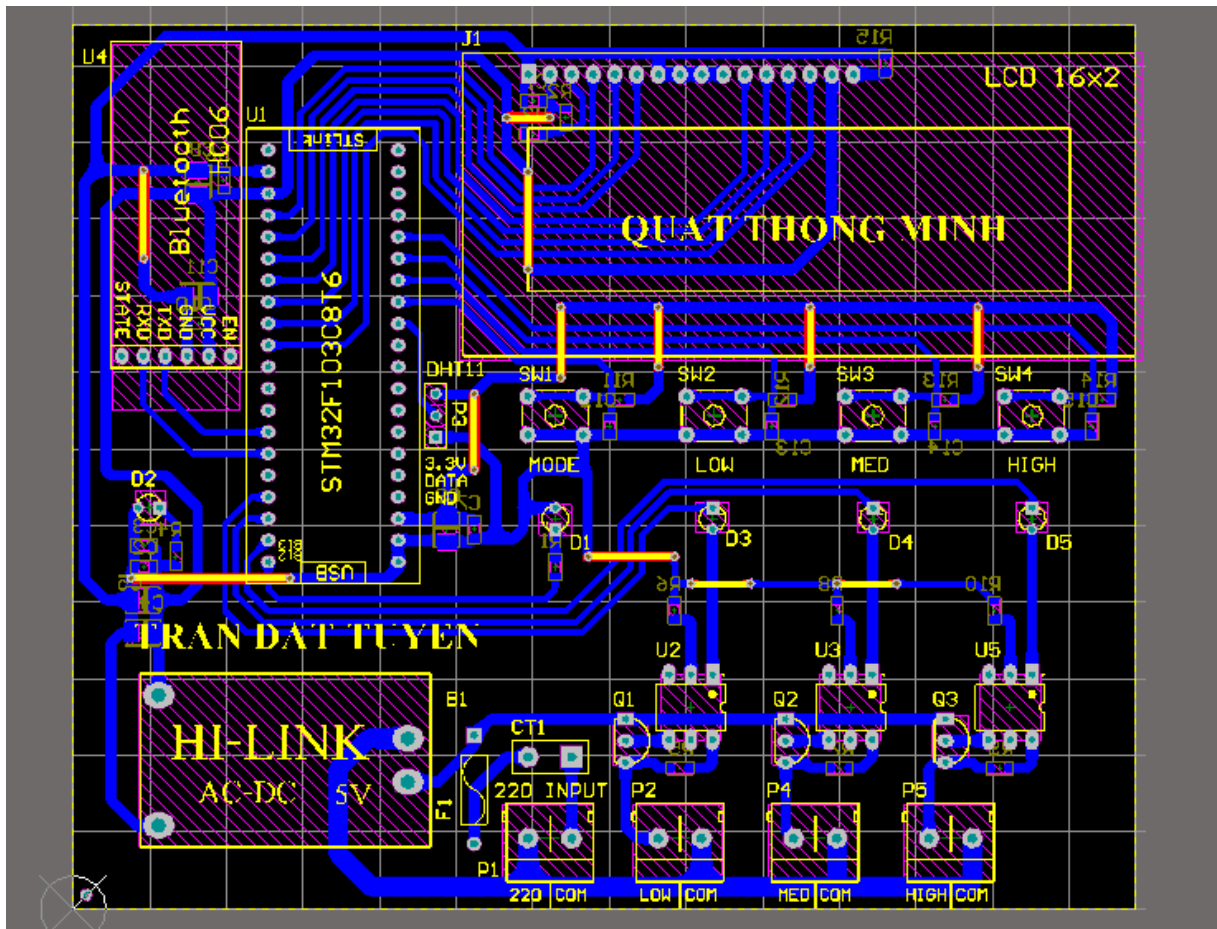
Sơ đồ nguyên lý được thiết kế trên phần mềm chuyên dụng.



Hình 3.1 Sơ đồ mạch nguyên lý

3.1.2. Hình ảnh mạch PCB.

Sơ đồ mạch PCB được thiết kế trên phần mềm chuyên dụng.



Hình 3.2 Mạch PCB một lớp.

Mạch PCB được thiết kế một lớp, với cấu trúc linh kiện xa nhau, phần công suất xa phần điều khiển để cách li và chống nhiễu.

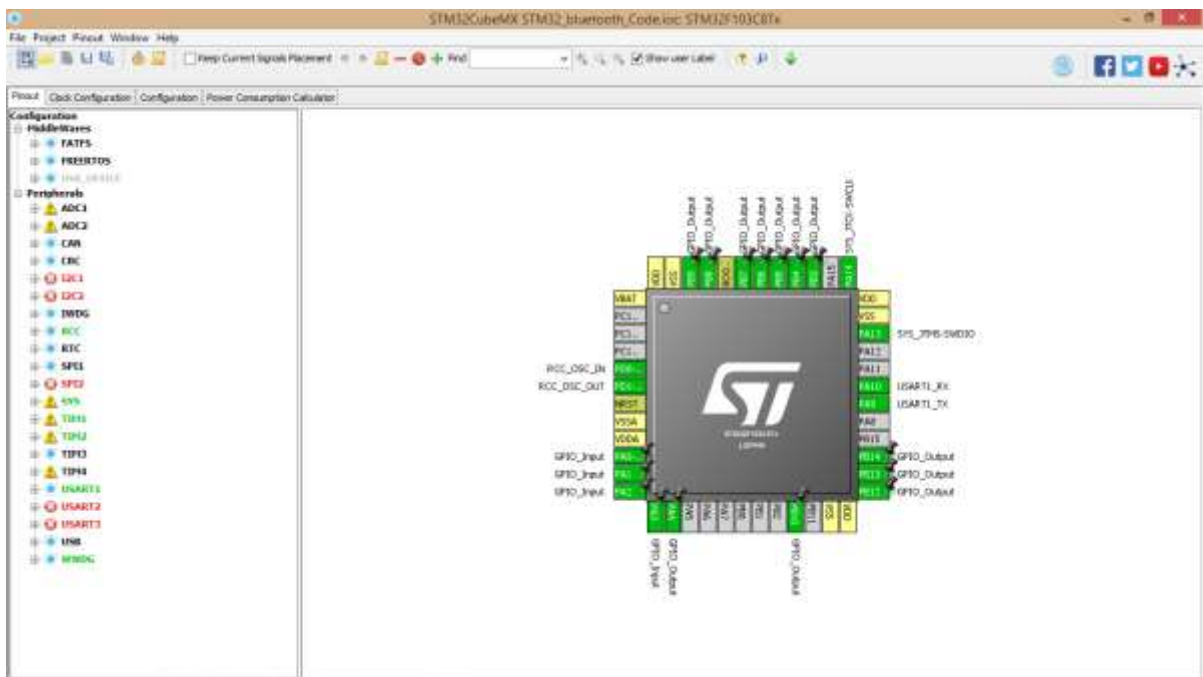
3.2. Xây dựng phần mềm.

3.2.1. Xây dựng phần mềm nhúng.

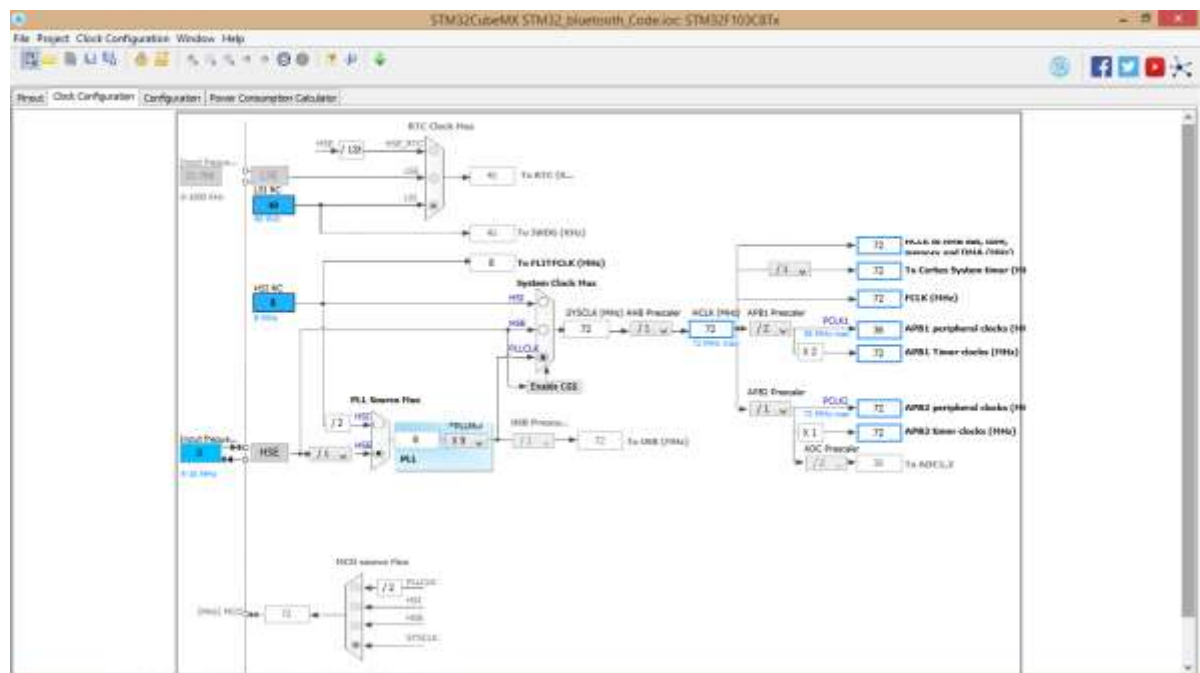
3.2.1.1. Phần mềm CubeMX.

CubeMX là phần mềm hỗ trợ lập trình cho dòng chip STM32 sử dụng bộ thư viện HAL. Nó hỗ trợ cấu hình xung clock, chân output input, các giao tiếp ngoại vi như UART SPI I2C bộ TIMER. CubeMX tạo nên các project để sử dụng cho nhiều phần mềm lập trình nhúng khác để người sử dụng có thể dễ dàng lập trình con chip của mình hơn.

Một số hình ảnh cấu hình trên CubeMX.



Hình 3.3 Giao diện cubeMX cấu hình chân sử dụng.

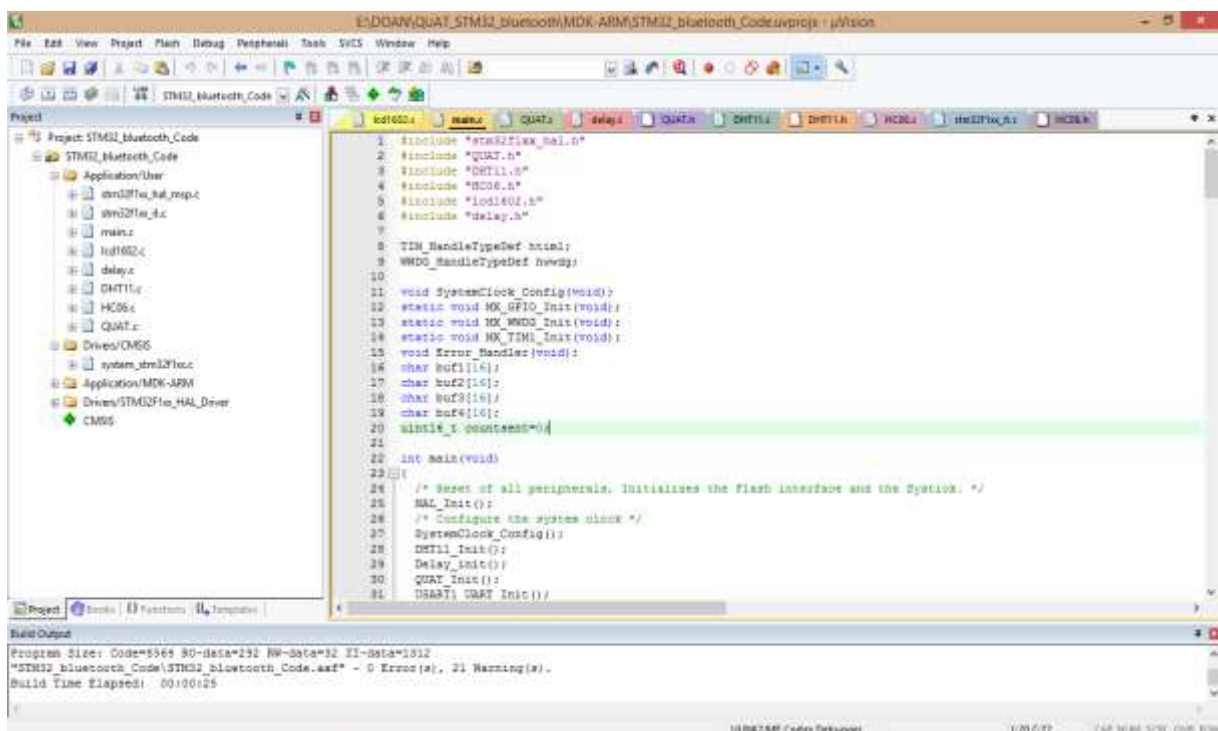


3.2.1.2. Phần mềm lập trình keilC

Giống như các dòng vi điều khiển khác ARM cũng có thể lập trình bằng assembly hoặc bằng C. Để dễ dàng lập trình thì trong khóa luận này tôi sử dụng C là ngôn ngữ để viết chương trình. Trên thị trường cũng có nhiều mã nguồn mở dùng để viết các chương trình cho arm. Trong số đó có KeilC là 1 trong những compiler được nhiều người yêu thích vì tính đa dạng và hỗ trợ nhiều dòng vi điều khiển khác nhau. Giao diện cũng dễ nhìn và dễ sử dụng. Phần mềm cũng hỗ trợ các công cụ debug với dòng ARM như ST-link. Chúng ta có thể dễ dàng tìm mua ở các cửa hàng linh kiện điện tử.

Project trên keilc dễ dàng dduwwocj tạo ra qua phần mềm CubeMx ở trên do đó không tốn nhiều thời gian để tự cấu hình và đơn giản hóa trong lập trình nhúng hơn. Giao diện phần mềm KeilC để sử dụng và tiếp cận.

Giao diện KeilC.



Hình 3.5 Giao diện phần mềm keilC.

3.2.1.3. Xây dựng phần mềm nhúng.

Phần mềm được chia nhỏ thành các file để dễ dàng quản lý. File Main.c là file chính để gọi các hàm xử lý trong các file chức năng khác.

Hàm main là hàm chính, cấu trúc hàm main sẽ thực hiện các công việc.

- Khởi tạo tần số clock cho vi điều khiển.
- Khởi tạo kết nối tới cảm biến nhiệt độ độ ẩm.
- Khởi tạo chế độ điều khiển ban đầu của quạt.
- Khởi tạo kết nối tới module Bluetooth.
- Khởi tạo hiển thị lên LCD.
- Thực hiện vòng lặp kiểm tra nút nhấn, tín hiệu điều khiển từ module bluetooth, đọc giá trị nhiệt độ độ ẩm và hiển thị lên LCD.

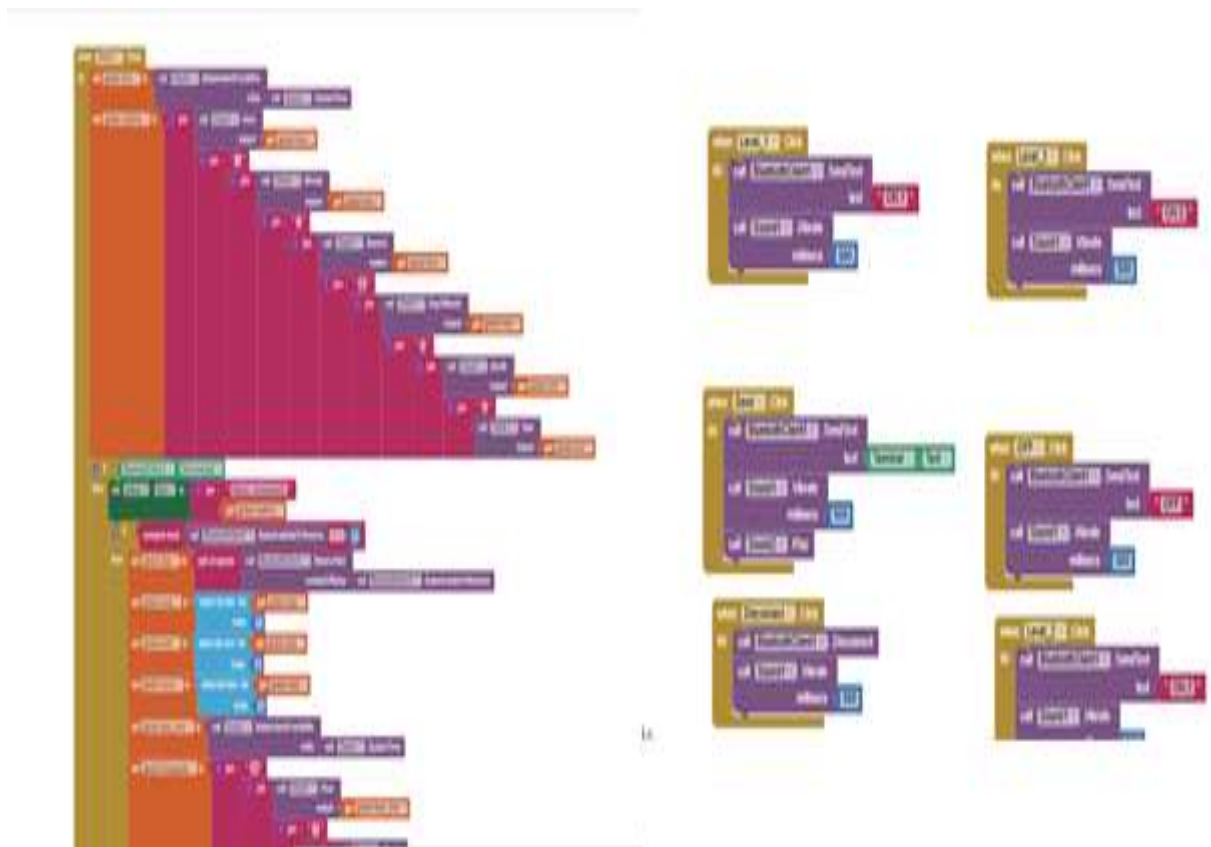
3.2.2. Xây dựng phần mềm trên thiết bị android.

Thiết kế phần mềm trên hệ điều hành điều khiển quạt trên hệ điều hành android. nền tảng MIT App Inventor

Là một môi trường lập trình trực quan, trực quan cho phép mọi người - xây dựng các ứng dụng đầy đủ chức năng cho điện thoại thông minh và máy tính bảng. Những người mới đến MIT App Inventor có thể có một ứng dụng đầu tiên đơn giản và chạy trong thời gian ngắn. Môi trường phù hợp với những ứng dụng đơn giản giúp tiết kiệm thời gian và chi phí. Sử dụng các khối giúp cho việc thiết kế dễ dàng và tiện lợi.

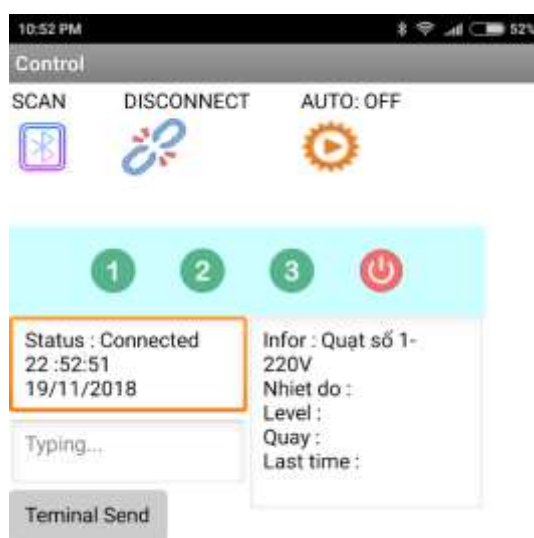
Chức năng của ứng dụng web

- Hiện thị danh sách thiết bị có thể connect.
- Hiện thị trạng thái thiết bị tốc độ quạt nhiệt độ môi trường.
- Chức năng chạy tự động hay chức năng chạy bằng tay
- Thông báo khi có lỗi xảy ra.



Hình 3.6 Một số đoạn code của sản phẩm.

3.2.2.1. Kết quả



Hình 3.7 Sản phẩm sau khi hoàn chỉnh.

Trước khi mở ứng dụng người dùng cần bật bluetooth trên điện thoại của mình. Nếu không thiết bị sẽ hiện thông báo. Sau khi đã vào đến màn hình chính. Người dùng kích/ bấm vào biểu tượng bluetooth và lựa chọn thiết bị kết nối. Nếu có lỗi thì sản phẩm sẽ thông báo lại về cho người sử dụng. Sản phẩm với chức năng điều khiển quạt người dùng có 2 chế độ chính là chế độ rảnh tay và chế độ điều khiển bằng tay

- Chế độ điều khiển rảnh tay quạt sẽ tự động thay đổi số dựa vào nhiệt độ thu được. nhiệt độ và số của quạt sẽ luôn được cập nhật và gửi về app.
- Chế độ điều khiển bằng tay người dùng sẽ có thể sử dụng bảng điều khiển với các phím số trên app. Nút nguồn dùng để tắt quạt. mọi thông tin đều được gửi về ứng dụng trên điện thoại về số của quạt nhiệt độ quay xung quanh thời gian cập nhật cuối cùng.lưu ý khi ở chế độ điều khiển tự động người sử dụng sẽ không thể sử dụng được các nút điều khiển như chế độ điều khiển bằng tay

Ưu điểm: đáp ứng đủ những yêu cầu đề ra cho 1 ứng dụng điều khiển bằng điện thoại android, giao diện dễ nhìn

Nhược điểm: Sản phẩm mới chỉ đáp ứng được các nhu cầu đề ra cho 1 ứng dụng cơ bản

Vì hạn chế của thời gian và người viết không không phải sinh viên đã có kinh nghiệm viết lập trình web. Nên web chỉ đáp ứng được một vài yêu cầu cơ bản của hệ thống đơn giản. Các chức năng cũng được tối giản hóa sao cho đơn giản và dễ dàng thao tác. Quá trình viết lập trình website bao gồm.

Đầu tiên tôi sử dụng phần mềm soạn thảo code web là Notepad++ với giao diện hỗ trợ lập trình ngôn ngữ php. Qua quá trình kiểm tra chạy thử nghiệm trên máy tính cá nhân thông qua sever tự tạo trên máy tính bằng phần mềm xampp. Sau khi chạy thử thành công.

3.3. Kết quả mô phỏng.

3.3.1. Thiết bị demo thực tế.



Hình 3.8 Thiết bị được điều khiển bằng điện thoại android.

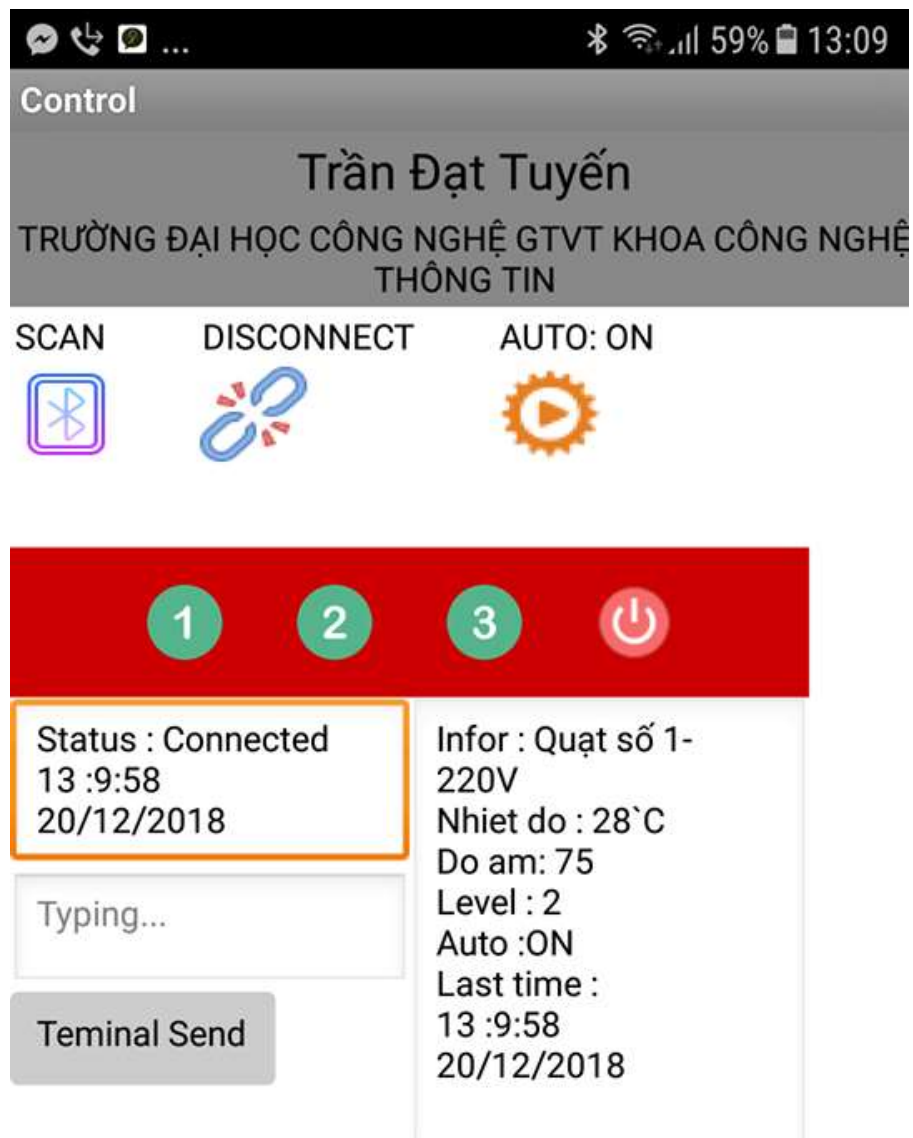
3.3.2. Thiết bị được điều khiển bằng điện thoại android.



Hình 3.9 Thiết bị có được điều khiển qua điện thoại android.

3.4. Đánh giá kết luận

Qua quá trình test trên thực tế, thiết bị quạt thông minh có thể được điều khiển bằng điện thoại android dễ dàng và đáng tin cậy, dữ liệu nhiệt độ độ ẩm cũng như các chế độ được cập nhật nhanh chóng và chính xác. Thiết bị có thể điều khiển với



Hình 3.10 Thông số hiển thị trên điện thoại.

Với chế độ bằng tay , thiết bị hoạt động chính xác và đúng cơ chế đã đề ra , có đèn và màn hình thông báo các thông số chính xác và đầy đủ.



Hình 3.11 Thông số thiết bị với chế độ bằng tay.

Với chế độ tự động , thiết bị hoạt động chính xác, điều khiển quạt đúng theo dải nhiệt độ đã đề ra. Nhưng dải nhiệt độ hoạt động là dải nhiệt cố định, chưa có chế độ cài đặt dải nhiệt độ.



Hình 3.12 Thông số thiết bị với chế độ tự động đèn thông báo chế độ tự động.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết Luận

Qua quá trình nghiên cứu thực hiện đề tài khóa luận “**Thiết kế chế tạo quạt thông minh điều khiển qua điện thoại Android**” dưới sự hướng dẫn của thầy em đã hiểu thêm được rất nhiều công nghệ, và biết thêm cách áp dụng thiết kế một sản phẩm phục vụ cho nhu cầu cuộc sống.

Khóa luận đã thiết kế được một thiết bị ổn định, hoạt động với độ tin cậy cao. Các linh kiện được chọn hợp lý. Các thông số điều khiển, thông báo tin cậy và chính xác. Sản phẩm có thể phát triển thêm và đưa vào sử dụng trên thực tế phục vụ cuộc sống.

Định hướng phát triển.

Thiết bị hiện tại chưa thể cấu hình dải nhiệt độ điều khiển khi ở chế độ tự động. khi phát triển thêm chức năng này thiết bị quạt sẽ hoàn thiện hơn về chức năng

Hiện tại , do thời gian có hạn nên giao diện phần mềm điều khiển trên android vẫn còn sơ sài. Update thêm giao diện sẽ làm cho sản phẩm có phần mềm điều khiển đẹp hơn. Người sử dụng có hứng thú hơn.

PHỤ LỤC

TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt:

- [1] Ngô Diên Tập, *Vi điều khiển với lập trình C*, NXB Khoa học và kỹ thuật, Hà nội.
- [2] Huỳnh Đức Thắng, *Cẩm nang thực hành vi mạch tuyến tính, TTL/LS, CMOS*, NXB Khoa học và kỹ thuật, Hà nội, 1994.
- [3]. Giáo trình kỹ thuật vi xử lý và vi điều khiển, NXB khoa học và kỹ thuật.

Tài liệu tiếng Anh

- [4] http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-mainstream-mcus/stm32f1-series/stm32f103.html
- [5] en.CD00171190.pdf, *STM32F1xx Reference manual*

Tài liệu từ các website:

- [6] Thông số kỹ thuật do hãng stmicroelectronics cung cấp. Có thể truy cập vào địa chỉ <http://www.st.com/resource/en/datasheet/cd00161566.pdf>
- [7] Bài viết của Trịnh Quang trên diễn đàn cơ điện tử Việt Nam. Có thể truy cập vào địa chỉ <http://codientu.org/threads/9755> .
- [8] Aosong (Guangzhou) Electronics Co. Truy cập vào tháng 4 năm 2018. Tại địa chỉ <https://akizukidenshi.com/download/ds/aosong/DHT11.pdf>.