

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG



ĐỒ ÁN TỐT NGHIỆP

NGÀNH ĐIỆN TỬ ĐỘNG CÔNG NGHIỆP

Sinh viên: Đinh Hữu Mạnh

Giảng viên hướng dẫn: TS. Ngô Quang Vĩ

HẢI PHÒNG – 2020

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

ĐIỀU KHIỂN ĐỘNG CƠ 1 CHIỀU SỬ DỤNG VI XỬ LÝ
STM32F4

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP

Sinh viên : Đinh Hữu Mạnh

Giảng viên hướng dẫn: T.S Ngô Quang Vĩ

HẢI PHÒNG – 2020

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Đinh Hữu Mạnh

MSV: 1913102003

Lớp : DCL2301

Nghành : Điện Tự Động Công Nghiệp

Tên đề tài: Điều khiển động cơ 1 chiều sử dụng vi xử lý STM32F4

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp (về lý luận, thực tiễn, các số liệu cần tính toán và các bản vẽ).

.....

.....

.....

.....

.....

2. Các số liệu cần thiết để thiết kế, tính toán.

.....

.....

.....

.....

.....

3. Địa điểm thực tập tốt nghiệp.....

.....

.....

.....

.....

.....

CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : Ngô Quang Vĩ

Học hàm, học vị : Tiến sĩ

Cơ quan công tác : Trường Đại học Quản lý và Công nghệ Hải Phòng

Nội dung hướng dẫn: Điều khiển động cơ 1 chiều xử dụng vi xử lý STM32F4

Đề tài tốt nghiệp được giao ngày 12 tháng 10 năm 2020

Yêu cầu phải hoàn thành xong trước ngày 31 tháng 12 năm 2020

Đã nhận nhiệm vụ Đ.T.T.N

Sinh viên

Đinh Hữu Mạnh

Đã giao nhiệm vụ Đ.T.T.N

Cán bộ hướng dẫn Đ.T.T.N

TS. Ngô Quang Vĩ

Hải Phòng, ngày.....tháng năm 2020.

TRƯỞNG KHOA

Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên: Ngô Quang Vĩ

Đơn vị công tác: Trường Đại học Quản lý và Công nghệ Hải Phòng

Họ và tên sinh viên: Đinh Hữu Mạnh

Chuyên ngành: Điện tự động công nghiệp

Nội dung hướng dẫn : Toàn bộ đề tài

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

.....
.....

2. Đánh giá chất lượng của đồ án/khóa luận(so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận, thực tiễn, tính toán số liệu).

.....
.....

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Được bảo vệ Không được bảo vệ Điểm hướng dẫn

Hải Phòng, ngày.....tháng.....năm 2020

Giảng viên hướng dẫn

TS. Ngô Quang Vĩ

Cộng hòa xã hội chủ nghĩa Việt Nam

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên:.....

Đơn vị công tác:.....

Họ và tên sinh viên:**Chuyên ngành:**.....

Đề tài tốt nghiệp:

.....

1. Phần nhận xét của giảng viên chấm phản biện

.....

.....

.....

2. Những mặt còn hạn chế

.....

.....

.....

3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ Không được bảo vệ Điểm hướng dẫn

Hải Phòng, ngày.....tháng.....năm 2020

Giảng viên chấm phản biện

Lời mở đầu

Hiện nay các quá trình sản xuất và quản lý như các hệ thống đo lường điều khiển tự động trong sản xuất công nghiệp, các hệ thống di động và không dây tiên tiến, các hệ thống thông tin vệ tinh các hệ thống dựa trên Web, chính phủ điện tử, thương mại điện tử, các cơ sở dữ liệu của nhiều ngành kinh tế và của Quốc gia, các hệ thống Y tế hiện đại, các thiết bị dân dụng ,.....đều là sản phẩm kết hợp giữa các lĩnh vực khoa học trên .

Do vậy các dòng Vi xử lý và Vi điều khiển đã hình thành và phát triển từ rất sớm, nhưng đối với Việt Nam thì sự ứng dụng chưa được nhiều. Những nơi sử dụng Vi xử lý và Vi điều khiển chủ yếu ở những khu công nghệ cao, viện nghiên cứu hoặc những trường đại học top đầu, việc tiếp cận với công nghệ mới, những dòng Vi xử lý và Vi điều khiển mới cần được nghiên cứu có những hạn chế nhất định. Do vậy đó em nhận thấy thật sự cần thiết để nghiên cứu và tìm hiểu về vấn đề này.

Quá trình học tập tại trường em có được giao đề tài: “*Điều khiển động cơ 1 chiều sử dụng Vi xử lý STM32F4*”. Đề tài này dưới sự hướng dẫn của thầy Ngô Quang Vĩ, dù đã cố gắng hết sức để có thể hoàn thành tốt nhất đề tài được giao. Nhưng quá trình làm đề tài không tránh khỏi những sai sót, em hy vọng nhận được sự đóng góp ý kiến của các thầy cô, bạn bè cùng lớp. Em xin chân thành cảm ơn!

Hải Phòng ngày ... tháng ... năm 2020

Sinh Viên

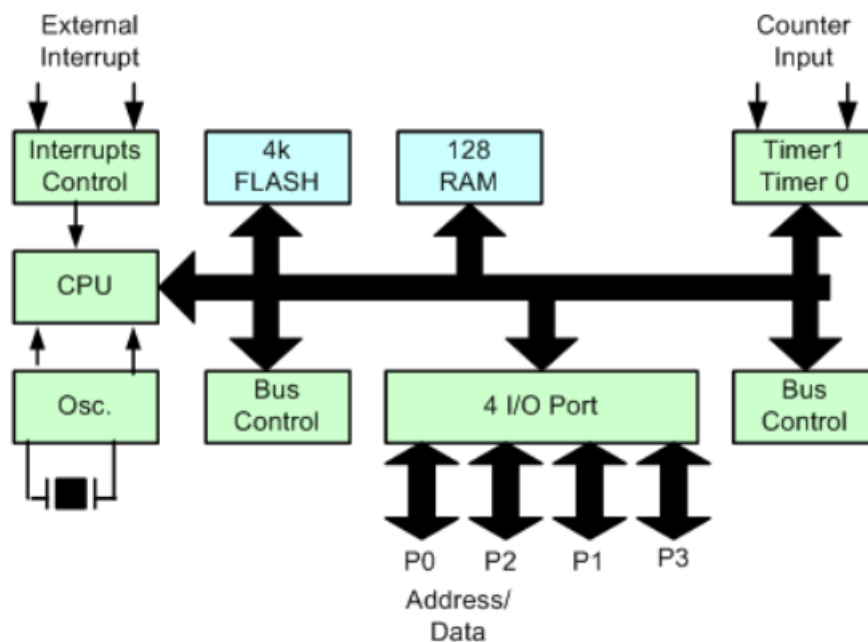
Đinh Hữu Mạnh

Chương 1 Giới thiệu dòng vi điều khiển STM32

1.1. Khái niệm về Vi xử lý và Vi điều khiển

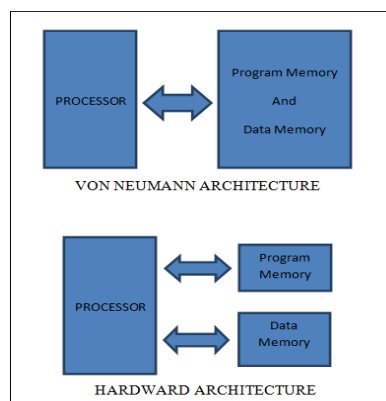
1.1.1. Vi xử lý

Bộ vi xử lý (microprocessor) là một máy tính nhỏ hoặc CPU (đơn vị xử lý trung tâm) được sử dụng để tính toán, thực hiện phép toán logic, kiểm soát hệ thống và lưu trữ dữ liệu vv. Vi xử lý sẽ xử lý các dữ liệu đầu vào / đầu ra (input/output) thiết bị ngoại vi và đưa ra kết quả trở lại để chúng hoạt động. Dòng vi xử lý 4 bit đầu tiên được Intel sản xuất vào tháng 11/1971 với tên gọi là 4004.



Hình 1. 1. Sơ đồ khối hệ thống Vi xử lý

Các loại cấu trúc:



Hình 1. 2. Kiến trúc phần cứng của STM32

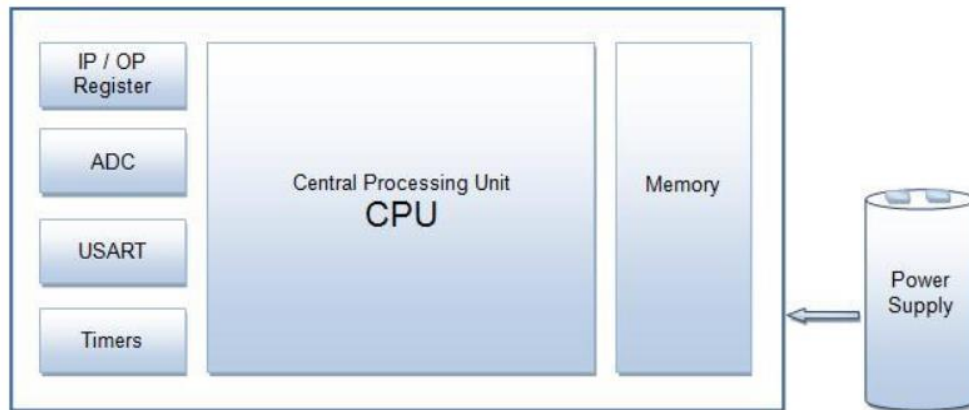
- + Các vi xử lý đầu tiên sử dụng **cấu trúc Von-Neumann**. Trong cấu trúc Von Neumann bộ nhớ dữ liệu và bộ nhớ chương trình được đặt trong một bộ nhớ. Để xử lý một lệnh từ bộ nhớ hoặc yêu cầu từ I / O, nó nhận được lệnh thông qua bus từ bộ nhớ hoặc I / O, và đặt vào thanh ghi, xử lý nó trong các thanh ghi. Bộ xử lý có thể lưu kết quả trong bộ nhớ thông qua các bus. Nhưng kiến trúc này có một số nhược điểm như chậm và quá trình truyền dữ liệu không đồng thời xảy ra cùng một lúc bởi vì chia sẻ cùng một bus chung.
- + Sau này **cấu trúc Harvard** (*Atmega328, Atmega168,... Arduino đang dùng*) được phát triển. Trong cấu trúc Harvard bộ nhớ dữ liệu và bộ nhớ chương trình và các bus được tách biệt với nhau. Ngoài ra còn có hai loại CPU micro programming và hardwired programming. Microprogramming còn chậm khi so sánh với hardwired programming.
- + **Kiến trúc tập lệnh Complex Instruction Set Computer**: complex instruction set computer (CISC) là tập lệnh phức tạp nên sẽ tốn nhiều thời gian để thực hiện; tập lệnh phức tạp có thể bao gồm quá trình xử lý opcode và các toán hạng ...vv tốc độ thực hiện lệnh sẽ chậm. **Cấu trúc X86** là một ví dụ.
- + **Reduced Instruction Set Computer**: Reduced Instruction Set Computer (RISC) là tập lệnh thu gọn và tốc độ thực hiện nhanh. Việc thực hiện rất đơn giản và không yêu cầu cấu trúc phức tạp. RISC được sử dụng rộng rãi trong các ứng dụng hệ thống nhúng. SHARC và PowerPC sử dụng RISC.

Bộ vi xử lý thường được dùng trong các ứng dụng nhỏ. Tùy theo các ứng dụng và thiết bị ngoại vi bạn đang sử dụng mà có thể chọn bộ vi xử lý cần thiết để thực hiện.

1.1.2. Vi điều khiển(microcontroller)

Nó cũng là một máy tính nhỏ, trong đó CPU, bộ nhớ (RAM, ROM), I / O thiết bị ngoại vi, timers, counters, được nhúng vào trong một mạch tích hợp (IC) nơi mà các bộ vi xử lý và tất cả các khối này được kết hợp vào trong một board thông qua hệ thống bus. Vi điều khiển có thể dễ dàng giao tiếp với thiết bị ngoại vi bên ngoài như cổng nối tiếp, ADC, DAC, Bluetooth, Wi-Fi, ...vv quá trình giao tiếp nhanh hơn khi so sánh với các bộ vi xử lý. Hầu hết các vi điều khiển sử

dụng cấu trúc RISC. Ngoài ra còn có một số vi điều khiển sử dụng cấu trúc CISC như 8051, motorolla, vv...

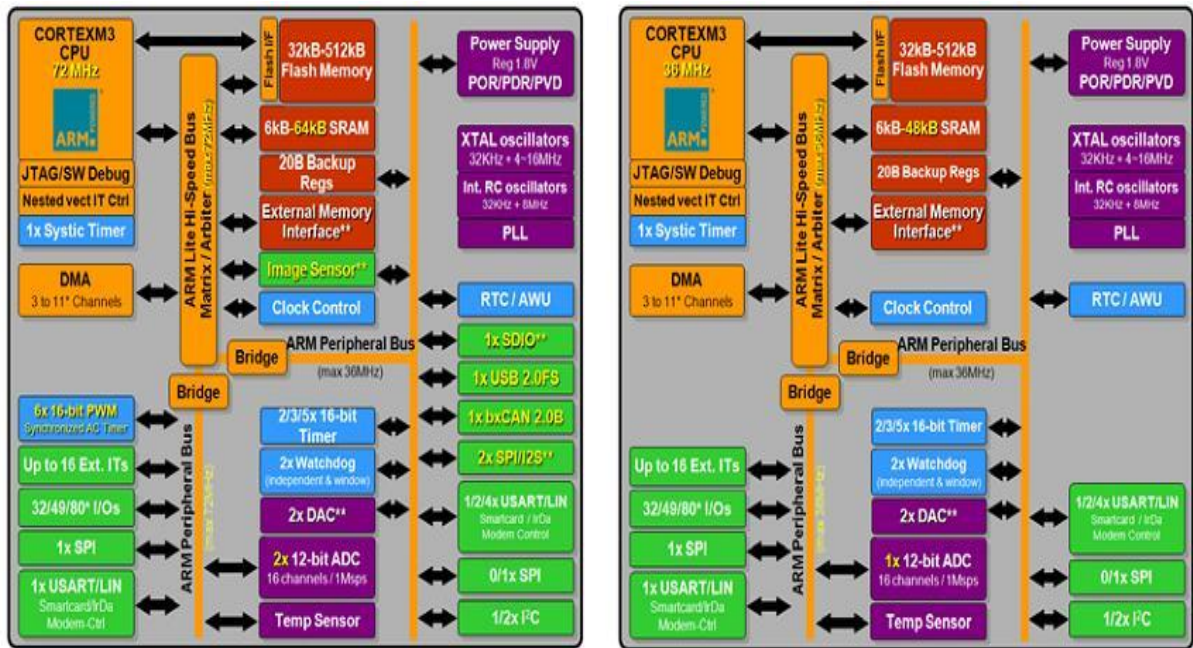


Hình 1. 3. Kiến trúc phần cứng Vi điều khiển

1.2. Đặc điểm nổi bật của STM32

Những đặc điểm nổi trội của dòng ARM Cortex đã thu hút các nhà sản xuất IC, hơn 240 dòng vi điều khiển dựa vào nhân Cortex đã được giới thiệu. Không nằm ngoài xu hướng đó, hãng sản xuất chip ST Microelectronic đã nhanh chóng đưa ra dòng STM32F4. STM32F4 là vi điều khiển dựa trên nền tảng lõi ARM Cortex-M3 thế hệ mới do hãng ARM thiết kế. Lõi ARM Cortex-M3 là sự cải tiến từ lõi ARM7 truyền thống từng mang lại thành công vang dội cho công ty ARM.

STM đã đưa ra thị trường 4 dòng vi điều khiển dựa trên ARM7 và ARM9, nhưng STM32 là một bước tiến quan trọng trên đường cong chi phí và hiệu suất (price/performance), giá chỉ gần 1 Euro với số lượng lớn, STM32 là sự thách thức thật sự với các vi điều khiển 8 và 16-bit truyền thống. STM32 đầu tiên gồm 14 biến thể khác nhau, được phân thành hai dòng: dòng Performance có tần số hoạt động của CPU lên tới 72Mhz và dòng Access có tần số hoạt động lên tới 36Mhz. Các biến thể STM32 trong hai nhóm này tương thích hoàn toàn về cách bố trí chân (pin) và phần mềm, đồng thời kích thước bộ nhớ FLASH ROM có thể lên tới 512K và 64K SRAM.



Hình 1. 4. Kiến trúc của STM32 nhánh Performance và Access

Nhánh Performance hoạt động với xung nhịp lên đến 72Mhz và có đầy đủ các ngoại vi, nhánh Access hoạt động với xung nhịp tối đa 36Mhz và có ít ngoại vi hơn so với nhánh Performance.

1.2.1. Sự tinh vi

Thoạt nhìn thì các ngoại vi của STM32 cũng giống như những vi điều khiển khác, như hai bộ chuyển đổi ADC, timer, I2C, SPI, CAN, USB và RTC. Tuy nhiên mỗi ngoại vi trên đều có rất nhiều đặc điểm thú vị. Ví dụ như bộ ADC 12-bit có tích hợp một cảm biến nhiệt độ để tự động hiệu chỉnh khi nhiệt độ thay đổi và hỗ trợ nhiều chế độ chuyển đổi. Mỗi bộ định thời có 4 khối capture compare (dùng để bắt sự kiện với tính năng input capture và tạo dạng sóng ở ngõ ra với output compare), mỗi khối định thời có thể liên kết với các khối định thời khác để tạo ra một mảng các định thời tinh vi hơn. Một bộ định thời cao cấp chuyên hỗ trợ điều khiển động cơ, với 6 đầu ra PWM với dead time (khoảng thời gian được chèn vào giữa hai đầu tín hiệu xuất PWM bù nhau trong điều khiển mạch cầu H) lập trình được và một đường break input (khi phát hiện điều kiện dừng khẩn cấp) sẽ buộc tín hiệu PWM sang một trạng thái an toàn đã được cài sẵn. Ngoại vi nối tiếp SPI có một khối kiểm tổng (CRC) bằng phần cứng cho 8 và 16 word hỗ trợ tích cực cho giao tiếp thẻ nhớ SD hoặc MMC.

STM32 có hỗ trợ thêm tối đa 12 kênh DMA (Direct Memory Access). Mỗi kênh có thể được dùng để truyền dữ liệu đến các thanh ghi ngoại vi hoặc từ các thanh ghi ngoại vi đi với kích thước từ (word) dữ liệu truyền đi có thể là 8/16 hoặc 32-bit. Mỗi ngoại vi có thể có một bộ điều khiển DMA (DMA controller) đi kèm dùng để gửi hoặc đòi hỏi dữ liệu như yêu cầu. Một bộ phân xử bus nội (bus arbiter) và ma trận bus (bus matrix) tối thiểu hoá sự tranh chấp bus giữa truy cập dữ liệu thông qua CPU (CPU data access) và các kênh DMA. Điều đó cho phép các đơn vị DMA hoạt động linh hoạt, dễ dùng và tự động điều khiển các luồng dữ liệu bên trong vi điều khiển.

STM32 là một vi điều khiển tiêu thụ năng lượng thấp và đạt hiệu suất cao. Nó có thể hoạt động ở điện áp 2V, chạy ở tần số 72MHz và dòng tiêu thụ chỉ có 36mA với tất cả các khối bên trong vi điều khiển đều được hoạt động. Kết hợp với các chế độ tiết kiệm năng lượng của Cortex, STM32 chỉ tiêu thụ 2 μ A khi ở chế độ Standby. Một bộ dao động nội RC 8MHz cho phép chip nhanh chóng thoát khỏi chế độ tiết kiệm năng lượng trong khi bộ dao động ngoài đang khởi động. Khả năng nhanh đi vào và thoát khỏi các chế độ tiết kiệm năng lượng làm giảm nhiều sự tiêu thụ năng lượng tổng thể.

1.2.2. Sự an toàn

Ngày nay các ứng dụng hiện đại thường phải hoạt động trong môi trường khắc khe, đòi hỏi tính an toàn cao, cũng như đòi hỏi sức mạnh xử lý và càng nhiều thiết bị ngoại vi tinh vi. Để đáp ứng các yêu cầu khắc khe đó, STM32 cung cấp một số tính năng phần cứng hỗ trợ các ứng dụng một cách tốt nhất. Chúng bao gồm một bộ phát hiện điện áp thấp, một hệ thống bảo vệ xung Clock và hai bộ Watchdogs. Bộ đầu tiên là một Watchdog cửa sổ (windowed watchdog). Watchdog này phải được làm tươi trong một khung thời gian xác định. Nếu nhần nó quá sớm, hoặc quá muộn, thì Watchdog sẽ kích hoạt. Bộ thứ hai là một Watchdog độc lập (independent watchdog), có bộ dao động bên ngoài tách biệt với xung nhịp hệ thống chính. Hệ thống bảo vệ xung nhịp có thể phát hiện lỗi của bộ dao động chính bên ngoài (thường là thạch anh) và tự động chuyển sang dùng bộ dao động nội RC 8MHz.

1.2.3. Tính bảo mật

Một trong những yêu cầu khắc khe khác của thiết kế hiện đại là nhu cầu bảo mật mã chương trình để ngăn chặn sao chép trái phép phần mềm. Bộ nhớ Flash của STM32 có thể được khóa để chống truy cập đọc Flash thông qua cổng Debug. Khi tính năng bảo vệ đọc được kích hoạt, bộ nhớ Flash cũng được bảo vệ chống ghi để ngăn chặn mã không tin cậy được chèn vào bảng vector ngắt. Hơn nữa bảo vệ ghi có thể được cho phép trong phần còn lại của bộ nhớ Flash. STM32 cũng có một đồng hồ thời gian thực và một khu vực nhỏ dữ liệu trên SRAM được nuôi nhờ nguồn pin. Khu vực này có một đầu vào chống giả mạo (anti-tamper input), có thể kích hoạt một sự kiện ngắt khi có sự thay đổi trạng thái ở đầu vào này. Ngoài ra một sự kiện chống giả mạo sẽ tự động xóa dữ liệu được lưu trữ trên SRAM được nuôi bằng nguồn pin.

1.2.4. Phát triển phần mềm

Nếu bạn đã sử dụng một vi điều khiển dựa trên lõi ARM, thì các công cụ phát triển cho ARM hiện có đã được hỗ trợ tập lệnh Thumb-2 và dòng Cortex. Ngoài ra ST cũng cung cấp một thư viện điều khiển thiết bị ngoại vi, một bộ thư viện phát triển USB như là một thư viện ANSI C và mã nguồn đó là tương thích với các thư viện trước đó được công bố cho vi điều khiển STR7 và STR9. Có rất nhiều RTOS mã nguồn mở và thương mại và middleware (TCP/IP, hệ thống tập tin, v.v.) hỗ trợ cho họ Cortex. Dòng Cortex-M3 cũng đi kèm với một hệ thống gỡ lỗi hoàn toàn mới gọi là CoreSight. Truy cập vào hệ thống CoreSight thông qua cổng truy cập Debug (Debug Access Port), cổng này hỗ trợ kết nối chuẩn JTAG hoặc giao diện 2 dây (serial wire-2 Pin), cũng như cung cấp trình điều khiển chạy gỡ lỗi, hệ thống CoreSight trên STM32 cung cấp hệ thống điểm truy cập (data watchpoint) và một công cụ theo dõi (instrumentation trace). Công cụ này có thể gửi thông tin về ứng dụng được lựa chọn đến công cụ gỡ lỗi. Điều này có thể cung cấp thêm các thông tin gỡ lỗi và cũng có thể được sử dụng trong quá trình thử nghiệm phần mềm.

1.2.5. Dòng Performance và Access của STM32

Họ STM32 có hai nhánh đầu tiên riêng biệt: dòng Performance và dòng Access. Dòng Performance tập hợp đầy đủ các thiết bị ngoại vi và chạy với xung nhịp tối đa 72MHz. Dòng Access có các thiết bị ngoại vi ít hơn và chạy tối đa

36MHz. Quan trọng hơn là cách bố trí chân (pins layout) và các kiểu đóng gói chip (package type) là như nhau giữa dòng Access và dòng Performance. Điều này cho phép các phiên bản khác nhau của STM32 được hoán vị mà không cần phải sửa đổi sắp xếp lại footprint (mô hình chân của chip trong công cụ layout bo mạch) trên PCB (Printed Circuit Board).

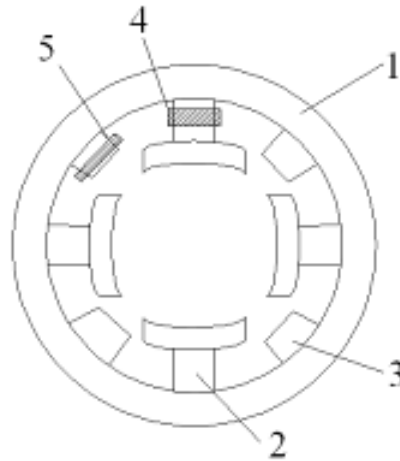
Chương 2 Các phương pháp điều khiển động cơ 1 chiều

2.1. Tổng quan về động cơ điện một chiều

2.1.1. Cấu tạo động cơ điện một chiều

Cấu tạo của máy điện một chiều gồm có stator và rotor nó được thể hiện ở Hình 2. 1.

Stator (còn gọi là phần tĩnh hay phần cảm)

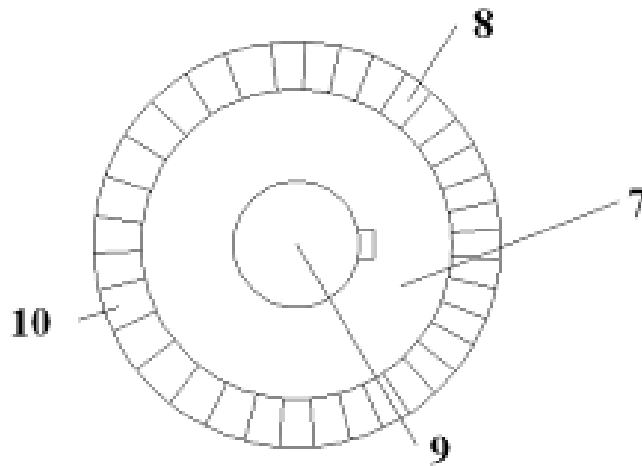


Hình 2. 1. Cấu tạo stator động cơ điện một chiều

1. Vỏ máy: làm bằng thép đúc làm nhiệm vụ dẫn từ, gá lắp các cực từ và bảo vệ máy.
2. Cực từ chính: bao gồm thân cực và mặt cực, thân cực làm bằng thép đúc, trên thân cực từ chính có quấn dây quấn kích từ. Mặt cực làm bằng lá thép kỹ thuật điện để tránh tác động của dòng xoáy fuco lan truyền từ rotor sang.
3. Cực từ phụ: làm bằng thép đúc, trên có quấn dây quấn kích từ phụ.
4. Dây quấn trên cực từ chính: làm bằng dây đồng bọc cách điện. Gọi là cuộn kích từ độc lập hay kích từ song song tùy thuộc vào cách đấu dây.
5. Dây quấn trên cực từ phụ: giống với cuộn nối tiếp nhưng quấn trên cực từ phụ và đấu nối tiếp với cuộn kích từ nối tiếp.

Phần ứng (rotor)

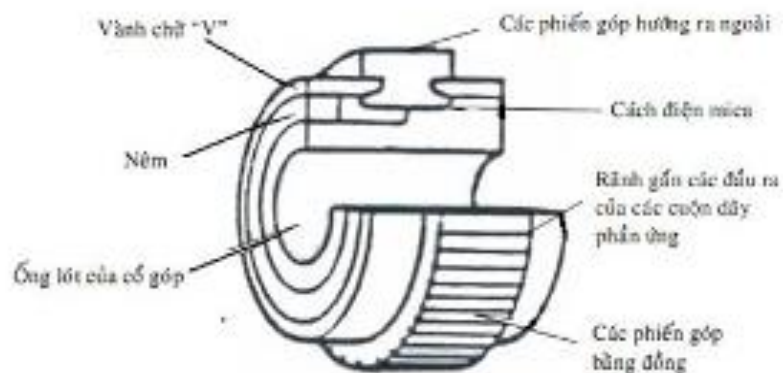
Rôto (phần ứng) của máy điện một chiều gồm: lõi thép, dây quấn phần ứng, cổ góp và trục máy Hình 2. 2



Hình 2. 2 Cấu tạo roto động cơ điện một chiều

6. Lõi thép phần ứng: Hình trụ làm bằng các lá thép kỹ thuật điện dày 0,5mm, phủ sơn cách điện ghép lại. Các lá thép được dập các lỗ thông gió và rãnh để đặt dây quấn phần ứng
7. Khe đặt dây quấn phần ứng: chu vi mặt ngoài được xẻ rãnh đều đặn để đặt dây quấn.
8. Trục rotor: làm bằng hợp kim thép có độ cứng cao, được đỡ ở 2 đầu bằng 2 vòng bi.
9. Dây quấn phần ứng: Gồm nhiều phần tử mắc nối tiếp nhau, đặt trong các rãnh của phần ứng tạo thành một hoặc nhiều vòng kín, hai đầu nối với hai phiến góp của vành góp.

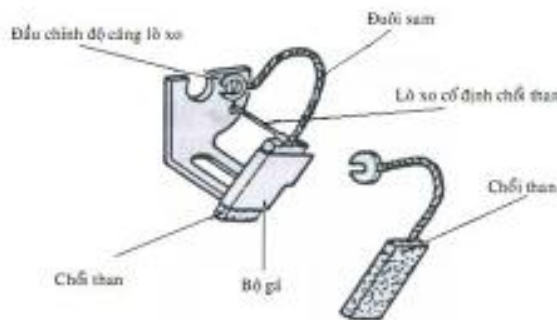
Chổi than và cổ góp: có cấu tạo như Hình 2. 3.



Hình 2. 3. Cấu tạo cổ góp động cơ điện một chiều

Cổ góp (vành góp) là tập hợp nhiều phiến đồng hình đuôi nhận được ghép thành một khối hình trụ, cách điện với nhau, được gắn trên trục máy và cách điện với trục máy.

Chổi than: cấu tạo từ bột than granit, có độ dẫn điện cao và khả năng chống mài mòn tốt. Tỳ lên các chổi than là các lò xo, các lò xo này có thể điều chỉnh lực căng để tăng tiếp xúc giữa chổi than và cổ góp. Chổi than của máy điện một chiều được thể hiện như Hình 2. 4.

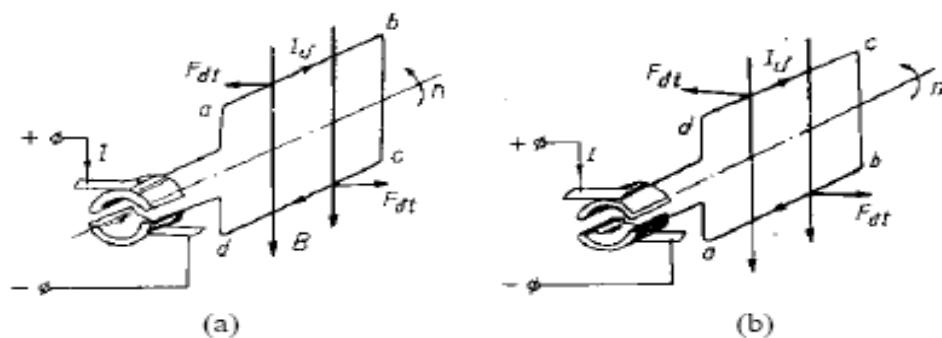


Hình 2. 4. Cấu tạo chổi than động cơ điện một chiều

2.1.2. Nguyên lý làm việc và phân loại động cơ điện một chiều

a. Nguyên lý làm việc động cơ điện một chiều:

Khi cho điện áp một chiều U vào hai chổi điện A và B, trong dây quấn phần ứng có dòng điện. Các thanh dẫn ab và cd mang dòng điện nằm trong từ trường sẽ chịu lực tác dụng tương hỗ lên nhau tạo nên mômen tác dụng lên rôto, làm quay rôto. Chiều lực tác dụng được xác định theo quy tắc bàn tay trái Hình 2. 5.



Hình 2. 5. Mô tả nguyên lý làm việc của động cơ điện một chiều

Khi phản ứng quay được nửa vòng, vị trí thanh dẫn ab, cd đổi chỗ nhau Hình 2. 5, nhờ có phiến góp đổi chiều dòng điện, nên dòng điện một chiều biến đổi thành dòng điện xoay chiều đưa vào dây quấn phần ứng, giữ cho chiều lực tác dụng không đổi, do đó lực tác dụng lên rôto cũng theo một chiều nhất định, đảm bảo động cơ có chiều quay không đổi.

b. Phân loại động cơ điện một chiều

Dựa theo phương pháp kích từ, động cơ một chiều có các loại như sau:

Động cơ một chiều kích từ độc lập.

Động cơ một chiều kích từ song song.

Động cơ một chiều kích từ nối tiếp.

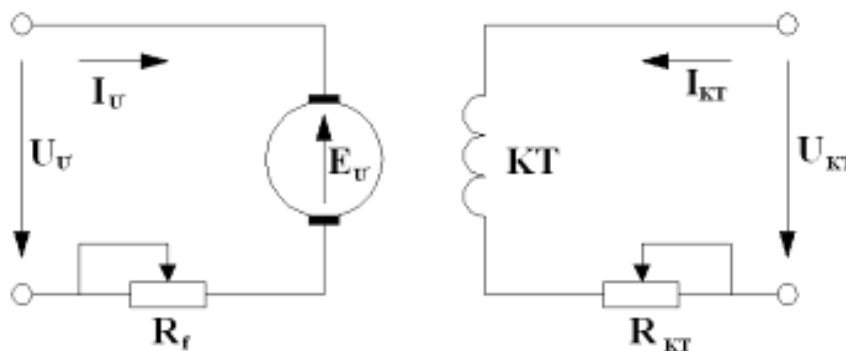
Động cơ một chiều kích từ hỗn hợp.

2.1.3. Phương trình đặc tính cơ động cơ điện một chiều

Do đối tượng sử dụng trong luận văn là động cơ điện một chiều kích từ độc lập nên dưới đây tác giả tập trung trình bày cách xây dựng đặc tính cơ của động cơ một chiều kích từ độc lập.

+ Xây dựng phương trình đặc tính cơ

Để điều khiển được tốc độ động cơ điện một chiều kích từ độc lập thì ta phải phân tích, tìm các mối quan hệ giữa tốc độ với các thông số khác của động cơ để từ đó đưa ra phương pháp điều khiển. Động cơ điện một chiều kích từ độc lập thì dòng kích từ độc lập với dòng phần ứng và được cấp bởi hai nguồn một chiều độc lập với nhau [2].



Hình 2. 6. Sơ đồ động cơ điện một chiều kích từ độc lập

Theo hình 2.6 ta viết phương trình cân bằng điện áp của mạch phần ứng.

$$U_{ur} = E_{ur} + (R_{ur} + R_f).I_{ur} \quad 2.1$$

Trong đó: U_{ur} : Điện áp đặt lên phần ứng động cơ (V).
 E_{ur} : Sức điện động phần ứng (V)
 R_{ur} : Điện trở của mạch phần ứng (Ω)
 R_f : Điện trở phụ trong mạch phần ứng (Ω)
 I_{ur} : Dòng điện mạch phần ứng (A)

Với: $R_{ur} = r_{ur} + r_{cf} + r_b + r_{ct}$

Trong đó : r_{ur} : điện trở cuộn dây phần ứng (Ω)
 r_{cf} : điện trở cuộn cực từ phụ (Ω)
 r_b : điện trở cuộn bù (Ω)
 r_{ct} : điện trở tiếp xúc chổi điện (Ω)

Sức điện động E_{ur} của phần ứng động cơ được xác định theo biểu thức (2.2).

$$E_{o,,} = K. \omega = \frac{P.N}{2.\pi.a} . \Phi. \omega \quad 2.2$$

Trong đó : $K = \frac{P.N}{2.\pi.a}$ hệ số cấu tạo của động cơ

P : Số đôi cực
N : Số thanh dẫn tác dụng
a : Số đôi mạch nhánh song song
 Φ : Từ thông kích từ dưới mỗi cực từ (wb)
 ω : Tốc độ góc (rad/s)

Nếu biểu diễn sức điện động theo tốc độ quay n (vòng/ phút):

$$E_{ur} = Ke. \Phi. n \quad 2.3$$

Mà

$$\omega = \frac{2.\pi.n}{60} \quad 2.4$$

Thay (2.4) vào (2.2) ta có:

$$E_{ur} = \frac{P.N}{60.a} . \Phi. n$$

$$\Rightarrow K_e = \frac{P.N}{60.a} = \frac{K}{9,55} \approx 0,105$$

K : Hệ số sức điện động của động cơ

Thay (2.1) vào (2.2) và biến đổi ta được:

$$\omega = \frac{U}{k.\phi} - \frac{R_u + R_f}{k.\phi} . I_u \quad 2.5$$

Biểu thức (2.6) là phương trình đặc tính cơ điện của động cơ:

Mặt khác, mômen điện từ:

$$M_{đt} = K . \Phi . I_v \quad 2.6$$

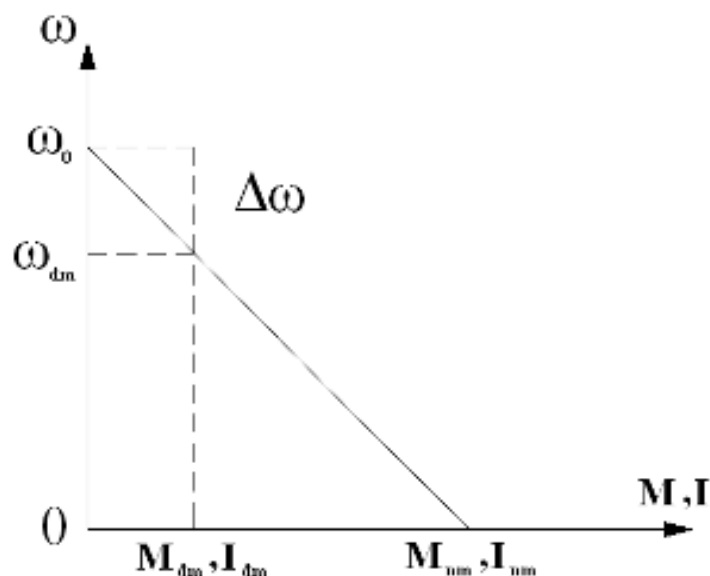
Nếu bỏ qua mọi tổn hao thì mômen cơ trên trục của động cơ chính bằng mômen điện từ, ta ký hiệu M tức là:

$$M_{đt} = M_{cơ} = M$$

Thay $M_{đt} = M_{cơ} = M$ vào phương trình (2.4) ta thu được phương trình đặc tính cơ của động cơ:

$$\omega = \frac{U}{k.\phi} - \frac{R_u + R_f}{(k.\phi)^2} . M \quad 2.7$$

Biểu thức (2.7) là phương trình đặc tính cơ của động cơ điện một chiều kích từ độc lập. Nếu ta xem $\Phi = \text{const}$ thì quan hệ $\omega = f(M, I)$ là tuyến tính.



Hình 2. 7. Đồ thị đặc tính cơ của động cơ điện một chiều kích từ độc lập

Từ hình 2.7 ta thấy : khi $I_v = 0$ hoặc $M = 0$ ta có:

$$\omega = \frac{U}{k.\phi} = \omega_0 \quad 2.8$$

Với ω_0 : tốc độ không tải lý tưởng của động cơ điện một chiều kích từ độc lập

Khi $\omega = \omega_0$ ta có:

$$I_u = \frac{U}{R_u + R_f} = I_{nm} \quad 2.9$$

Và

$$M = k. \Phi.I_{nm} = M_{nm} \quad 2.10$$

I_{nm}, M_{nm} : là dòng điện ngắn mạch và mômen ngắn mạch.

Mặt khác ta có: $I_u = \frac{M}{k.\phi}$, $R = R_u + R_f$, $\omega_0 = \frac{U}{k.\phi}$ nên phương trình (2.7) có thể

viết dưới dạng:

$$\omega = \frac{U}{k.\phi} - \frac{R_u + R_f}{(k.\phi)^2}.M = \omega_0 - \Delta\omega \quad 2.11$$

2.2. Các phương pháp điều khiển tốc độ động cơ điện một chiều

Từ biểu thức (2.7) ta có mối quan hệ $\omega = f(R, \Phi, U)$ do đó để điều chỉnh tốc độ động cơ điện một chiều ta có thể thực hiện bằng ba phương pháp điều khiển sau:

- + Điều khiển tốc độ bằng cách thay đổi điện trở phụ phản ứng.
- + Điều khiển tốc độ bằng cách thay đổi từ thông kích từ.
- + Điều khiển tốc độ bằng cách thay đổi điện áp phản ứng.

2.2.1. Phương pháp điều khiển tốc độ bằng cách thay đổi điện trở phụ phản ứng

Nguyên lý điều chỉnh: Nối thêm điện trở phụ R_f vào mạch phản ứng ta có mối quan hệ: $\omega = f(R_f, \Phi, U)$ với giả thiết rằng : Nếu giữ $\Phi = \Phi_{dm} = const$; $U = U_{dm} = const$; $R_u = const$ thì tốc độ của động cơ chỉ phụ thuộc vào điện trở phản ứng $\omega = f(R_f)$.

Để thực hiện thay đổi giá trị R_f của mạch phản ứng ta thực hiện bằng cách nối tiếp một điện trở phụ (R_f) có thể thay đổi giá trị vào mạch phản ứng.

Khi thêm điện trở R_f vào mạch phản ứng ta có : $R = R_u + R_f$

Theo phương trình đặc tính cơ :

$$\omega = \frac{U}{k.\phi} - \frac{R_u + R_f}{(k.\phi)^2} . M \quad 2.12$$

Từ (2.11) ta thấy: khi ta điều chỉnh tăng giá trị của R_f thì tốc độ của động cơ giảm và ngược lại

Lúc này ta có tốc độ không tải lý tưởng:

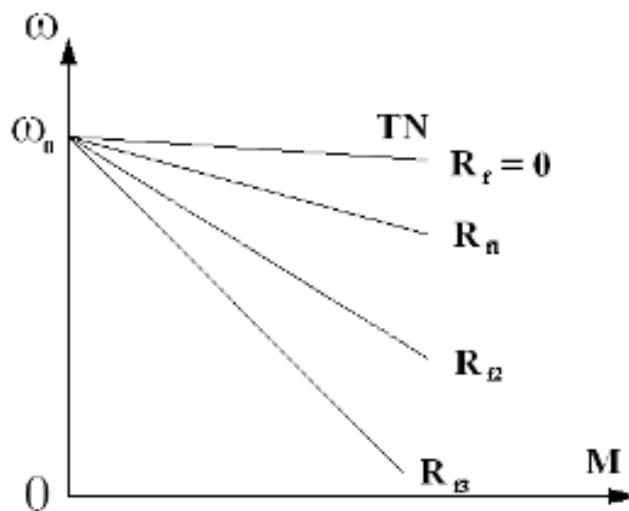
$$\omega_0 = \frac{U_{dm}}{k.\phi_{dm}} = \text{const} \quad 2.13$$

Và độ cứng của đặc tính cơ:

$$\beta = \frac{\Delta M}{\Delta \omega} = - \frac{(k.\phi_{nm})^2}{R_u + R_f} = \text{var} \quad 2.14$$

Khi R_f càng lớn, β càng nhỏ nghĩa là đặc tính cơ càng dốc. Ứng với $R_f = 0$ ta có đặc tính cơ tự nhiên. β tự nhiên có giá trị lớn nhất nên có độ cứng hơn tất cả các đường đặc tính có điện trở phụ.

Như vậy khi thay đổi R_f cho ta một họ đặc tính như sau:



Hình 2. 8. Đặc tính cơ của động cơ điện một chiều kích từ độc lập khi thay đổi điện trở phụ mạch phản ứng

Nếu giá trị điện trở R_f càng lớn thì tốc độ động cơ càng giảm, đồng thời dòng điện ngắn mạch (I_{nm}) và mô men ngắn mạch (M_{nm}) cũng giảm. Phương pháp này được dùng để hạn chế dòng khởi động của động cơ.

2.2.2. Phương pháp điều khiển tốc độ động cơ bằng cách thay đổi từ thông kích từ

Nguyên lý điều chỉnh: Phương pháp điều khiển tốc độ động cơ bằng cách thay đổi từ thông kích từ của động cơ điện một chiều là thực hiện điều chỉnh mô men điện từ của động cơ $M = k \cdot \Phi \cdot I_u$ và sức điện động quay của động cơ $E_u = k \cdot \Phi \cdot \omega$.

Theo phương trình đặc tính cơ $\omega = f(U, \phi_{kt}, R_f)$, thực hiện giữ $U = U_{dm} = \text{const}$, $R_u = \text{const}$ ($R_f = 0$) thì lúc này tốc độ của động cơ chỉ phụ thuộc vào từ thông kích từ $\omega = f(\Phi_{kt})$.

Vậy để điều chỉnh từ thông kích từ Φ_{kt} ta thực hiện mắc thêm biến trở R_v vào mạch kích từ của động cơ, khi điều chỉnh từ thông kích từ Φ_{kt} ta phải tuân theo điều kiện sau: không được tăng dòng kích từ I_{kt} lớn hơn dòng định mức của cuộn kích từ.

Khi thay đổi tốc độ bằng cách thay đổi từ thông kích từ ta có:

Tốc độ không tải lý tưởng của động cơ:

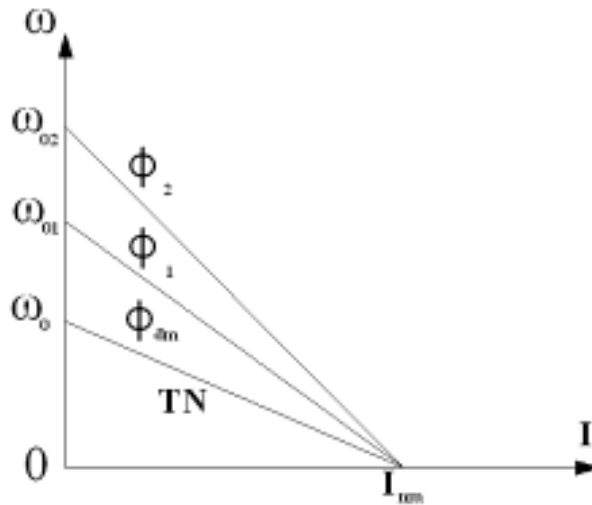
$$\omega_0 = \frac{U_{dm}}{k \cdot \phi_x} = \text{var} \quad 2.15$$

Độ cứng của đặc tính cơ :

$$\beta = \frac{(k \cdot \phi_x)^2}{R_u} = \text{var} \quad 2.16$$

Do cấu tạo của động cơ điện, khi từ thông kích từ: $\Phi_{kt} = \Phi_{dm}$ đã bão hòa, nếu muốn thực hiện tăng dòng kích từ I_{kt} thì Φ_{kt} tăng cũng không tăng đáng kể (do từ thông đã bão hòa) nên thực tế thường ta điều chỉnh bằng cách giảm từ thông kích từ Φ_{kt} . Khi từ thông kích từ giảm thì tốc độ của động cơ (ω_x) tăng,

còn độ cứng đặc tính cơ (β) sẽ giảm. Ta có đồ thị đặc tính cơ (hình 2.9) với ω_0 tăng dần và độ cứng của đặc tính giảm dần khi giảm từ thông.



Hình 2. 9. Đặc tính cơ điện của động cơ điện một chiều kích từ độc lập khi giảm từ thông.

Từ hình 2.9 ta thấy khi từ thông kích từ thay đổi với $\Phi_{đm} > \Phi_1 > \Phi_2$ ta có:

$$\text{Dòng điện ngắn mạch: } I_{nm} = \frac{U_{đm}}{R_u} = \text{const}$$

$$\text{Mô men ngắn mạch: } M_{nm} = K \cdot \Phi_x \cdot I_{nm} = \text{var} (M_{nm} > M_{nm1} > M_{nm2})$$

Từ đồ thị đặc tính (hình 2.9) ta thấy $\omega_0 < \omega_{01} < \omega_{02}$.

2.2.3. Phương pháp điều khiển tốc độ động cơ bằng cách thay đổi điện áp phản ứng

$$\text{Từ phương trình đặc tính cơ (2.7): } \omega = \frac{U}{k \cdot \phi} - \frac{R_u + R_f}{(k \cdot \phi)^2} \cdot M$$

Thực hiện điều chỉnh để giữ: $\Phi = \Phi_{đm} = \text{const}$, $R_u = \text{const}$, ($R_f = 0$), $M = \text{const}$. Lúc này tốc độ của động cơ chỉ phụ thuộc vào điện áp phản ứng: $\omega = f(U_r)$

Khi thực hiện thay đổi điện áp phản ứng của động cơ thì tốc độ không tải lý tưởng sẽ thay đổi, độ cứng đặc tính cơ là không đổi.

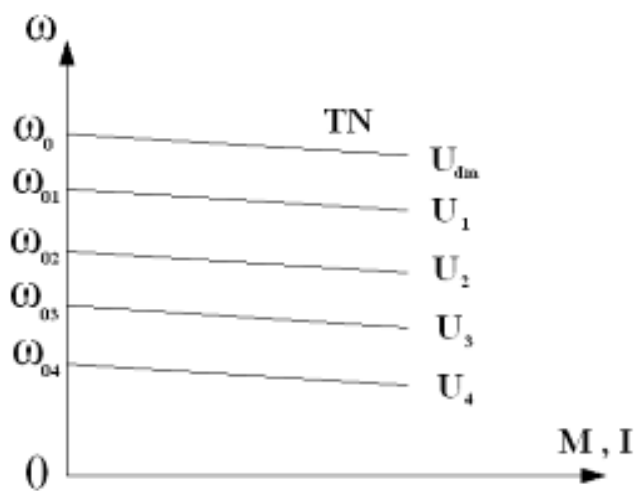
$$\text{Tốc độ không tải lý tưởng của động cơ: } \omega_0 = \frac{U_u}{k \cdot \phi} = \text{var}$$

(2.16)

$$\text{Độ cứng của đặc tính cơ của động cơ: } \beta = -\frac{(k \cdot \phi)^2}{R_u} = \text{const}$$

(2.17)

Như vậy khi thay đổi điện áp phần ứng của động cơ ta thu được một họ các đặc tính cơ như hình 2.10

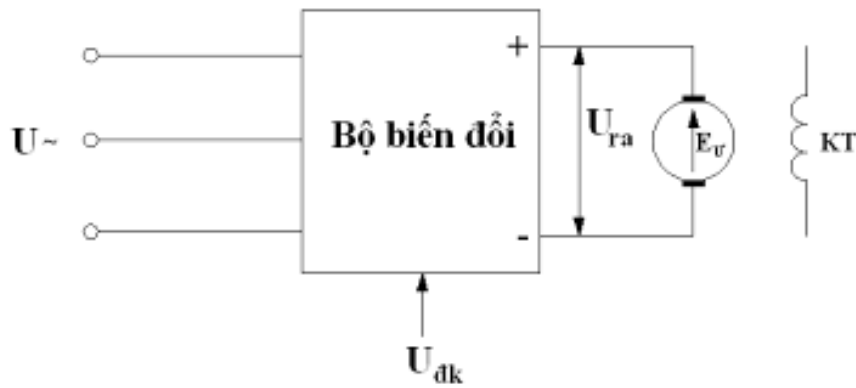


Hình 2. 10. Đồ thị đặc tính cơ của động cơ điện một chiều kích từ độc lập khi điện áp phần ứng thay đổi

Từ hình 2.10 ta thấy khi thay đổi điện áp phần ứng của động cơ ta được một họ đặc tính cơ nhân tạo song song với đặc tính tự nhiên.

Cũng từ hình 2.10 ta thấy rằng khi thay đổi điện áp phần ứng của động cơ thì mômen ngắn mạch (M_{nm}), dòng điện ngắn mạch (I_{nm}) của động cơ giảm và tốc độ động cơ cũng giảm. Do đó phương pháp điều chỉnh điện áp phần ứng được dùng để thực hiện điều chỉnh tốc độ động cơ và hạn chế dòng khởi động.

Để thực hiện điều chỉnh điện áp phần ứng, ta phải sử dụng một bộ biến đổi để thực hiện điều chỉnh điện áp đầu ra cấp cho mạch phần ứng của động cơ (hình 2.11).



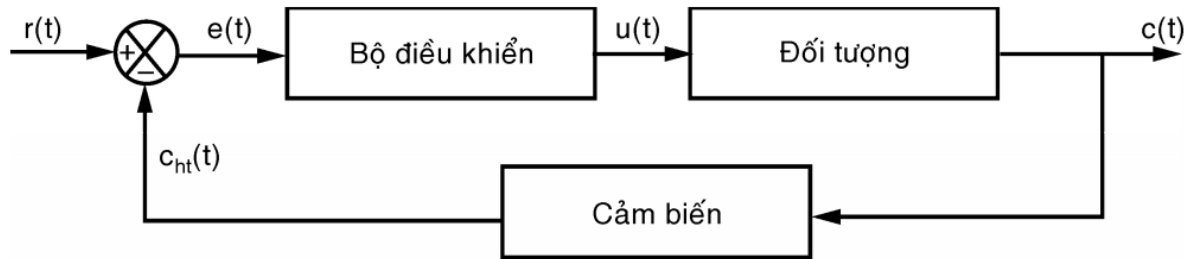
Hình 2. 11. Sơ đồ nguyên lý điều chỉnh động cơ

Bộ biến đổi dùng để biến đổi điện áp xoay chiều của lưới điện thành điện áp một chiều với giá trị điện áp đầu ra có thể điều chỉnh được theo yêu cầu của người điều khiển.

Kết luận : Sau khi phân tích ba phương pháp điều khiển tốc độ động cơ điện thì tác giả thấy rằng phương pháp điều khiển bằng thay đổi điện áp phản ứng là tốt hơn cả. Vì vậy tác giả lựa chọn phương pháp điều chỉnh tốc độ động cơ điện một chiều bằng cách thay đổi điện áp phản ứng làm phương pháp nguyên cứu cho đề tài.

Chương 3 Thiết kế và lập trình chương trình điều khiển động cơ 1 chiều trên STM32F407

3.1. Các thành phần cơ bản của hệ thống điều khiển

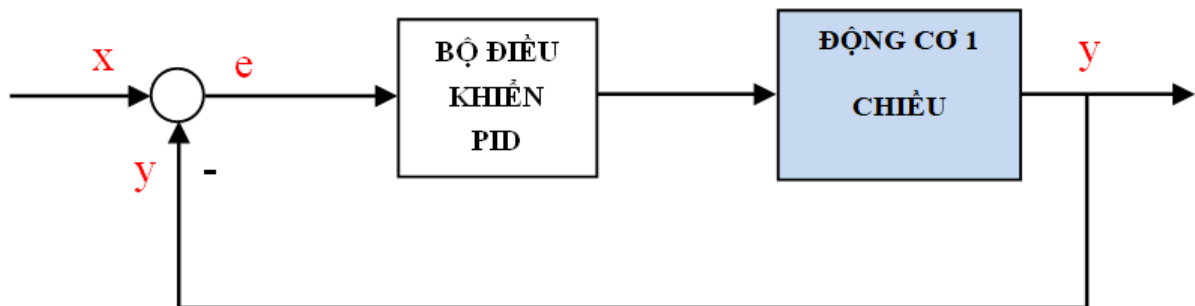


Hình 3. 1. Sơ đồ khối hệ thống điều khiển

Chú thích các ký hiệu viết tắt:

- + $r(t)$ (reference input): tín hiệu vào, tín hiệu chuẩn
- + $c(t)$ (controlled output): tín hiệu ra
- + $c_{ht}(t)$: tín hiệu hồi tiếp
- + $e(t)$ (error): sai lệch
- + $u(t)$: tín hiệu điều khiển

3.1.1. Sơ đồ điều khiển sử dụng bộ điều khiển PID



Hình 3. 2. Sơ đồ cấu trúc điều khiển sử dụng bộ điều khiển PID

Các tín hiệu vào/ra của sơ đồ:

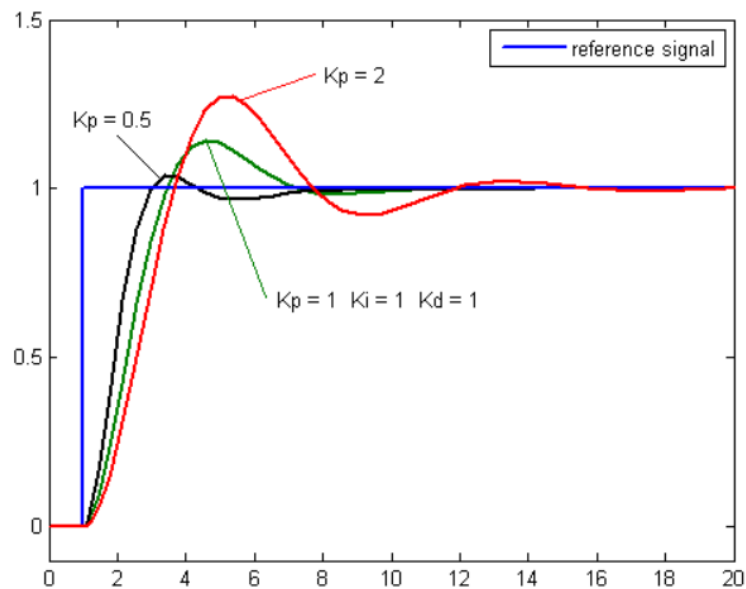
- + x là tín hiệu đặt
- + y là tín hiệu phản hồi về
- + e là sai lệch

Biểu thức tính sai lệch được cho như sau:

$e = x - y$	3.1
-------------	-----

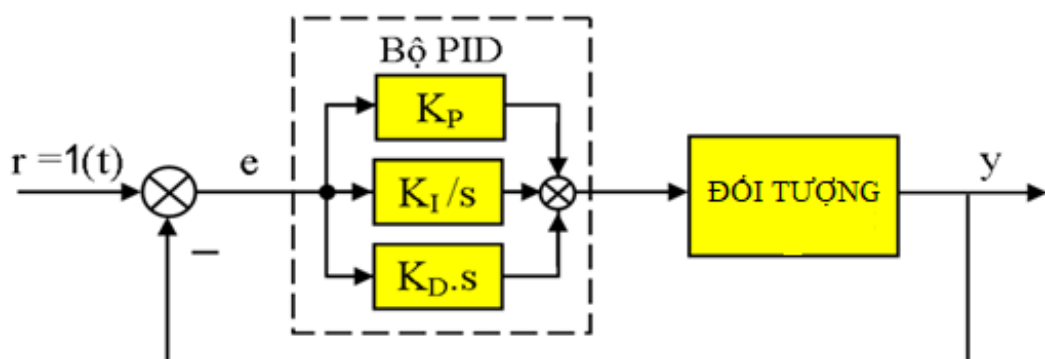
Có nhiều phương pháp xác định tham số của bộ điều khiển PID:

- + Phương pháp Ziegler-Nichols
- + Phương pháp Chien-Hrones-Reswick
- + Phương pháp tổng T của Kuhn
- + Phương pháp tối ưu modul và phương pháp tối ưu nổi xứng
- + Phương pháp tối ưu theo sai lệch bám



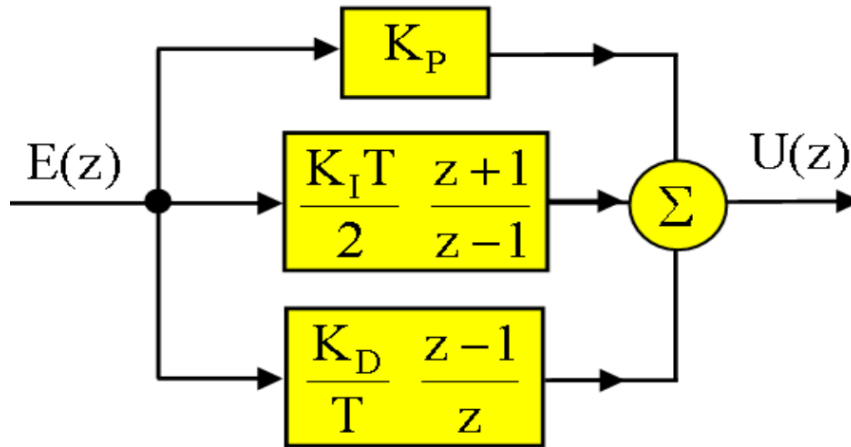
Hình 3. 3. Đồ thị theo thời gian, ba giá trị K_p , K_i , K_d

- + Bộ điều khiển PID liên tục:



Hình 3. 4. PID liên tục

+ Bộ điều khiển PID số (rời rạc):



Hình 3. 5. PID rời rạc

3.1.2. Một số bộ điều khiển khác

Một số bộ điều khiển khác bao gồm:

Bộ điều khiển mờ, Neuron

Bộ điều khiển LQR

3.2. Thiết kế phần cứng

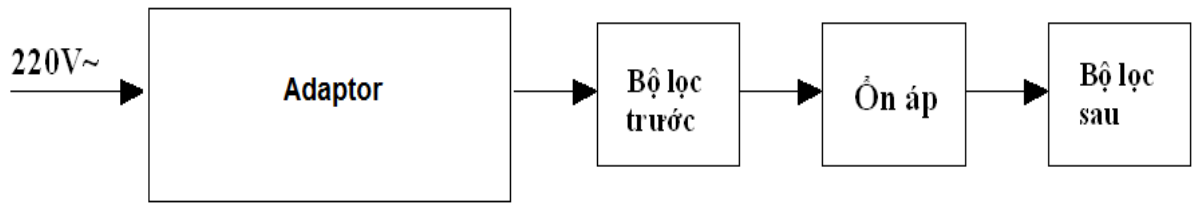
3.2.1. Khối mạch cấp nguồn

Khối mạch cấp nguồn

gồm có 2 mạch: cấp nguồn 5V một chiều và cấp nguồn 12V-2A một chiều.

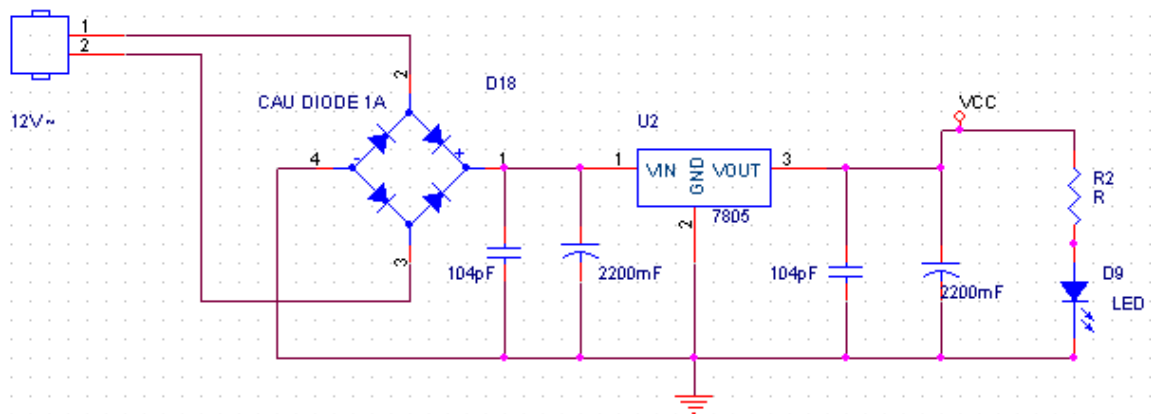
Mạch cấp nguồn 5VDC: để tạo ra điện áp một chiều 5VDC cấp cho vi điều khiển STM32F407 và IC MAX232 sử dụng IC ổn áp LM7805.

Sơ đồ khối của mạch cấp nguồn như sau:



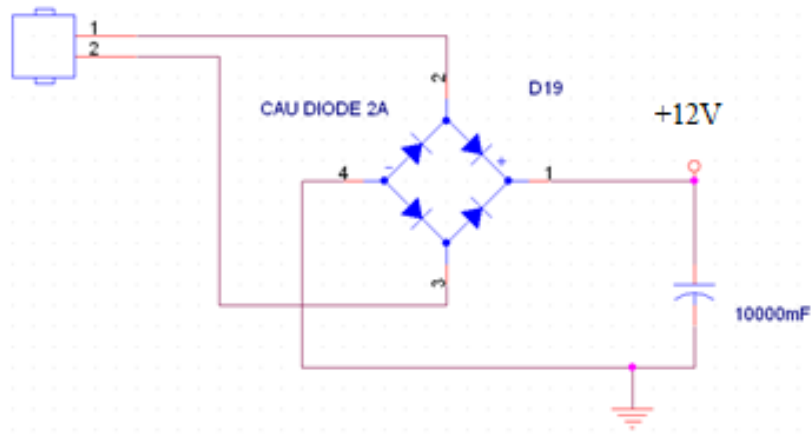
Hình 3. 6. Sơ đồ khối mạch cấp nguồn

Nguyên lý hoạt động của mạch như sau: điện áp 220VAC đưa qua bộ Adaptor hạ áp xuống 12VAC. Điện áp xoay chiều này đưa tới cầu chỉnh lưu để có được điện áp một chiều. Điện áp một chiều đưa tới chân Vin của LM7805 và đầu ra Vout ta có được điện áp 5VDC. Đôi khi đường cấp nguồn bị nhiễu, để san phẳng các nhiễu này ta sử dụng các bộ lọc trước và bộ lọc sau IC ổn áp để có được nguồn 5VDC ổn định. Các tụ sử dụng trong mạch : tụ gốm 104pF và tụ hóa 2200mF. Ngoài ra còn có 1 LED báo cấp nguồn.



Hình 3. 7. Sơ đồ nguyên lý mạch cấp nguồn 5VDC

Mạch cấp nguồn 12VDC cho động cơ: do chất lượng điện áp cấp cho động cơ không đòi hỏi quá cao nên ta sử dụng một mạch đơn giản như sau:

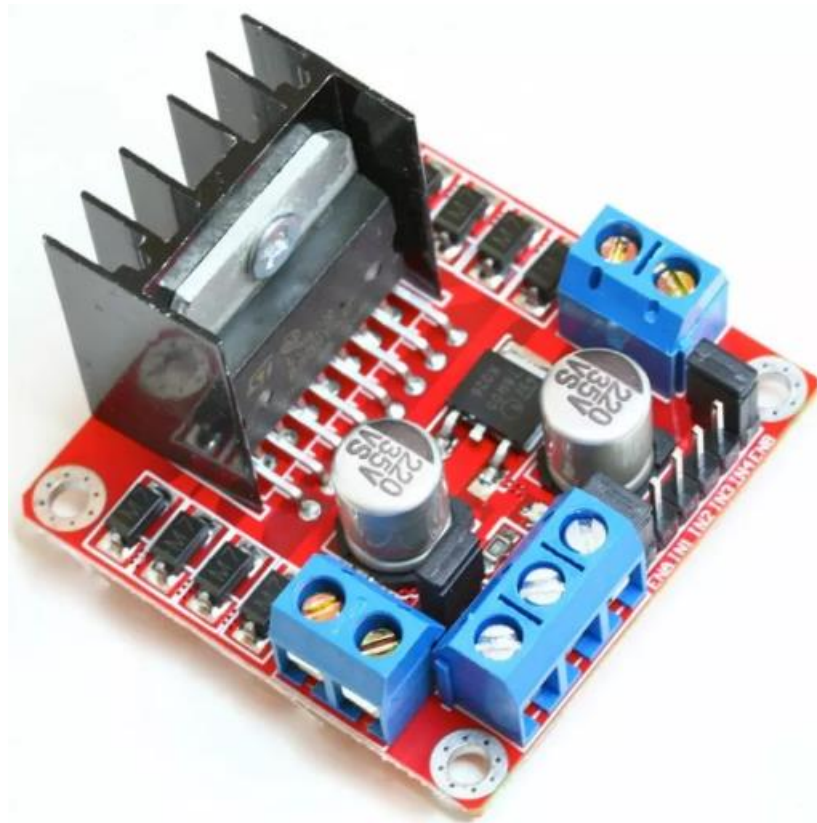


Hình 3. 8. Sơ đồ nguyên lý mạch cấp nguồn 12VDC

Nguyên lý hoạt động: điện áp 220VAC đưa qua biến áp hạ áp xuống 12VAC. Điện áp xoay chiều này đưa tới cầu chỉnh lưu để có được điện áp một chiều. Điện áp sau cầu chỉnh lưu vẫn nhấp nhô, để san phẳng điện áp ra ta sử dụng một tụ hóa có dung lượng 10000mF. Điện áp này cấp cho đối tượng là động cơ cần điều khiển.

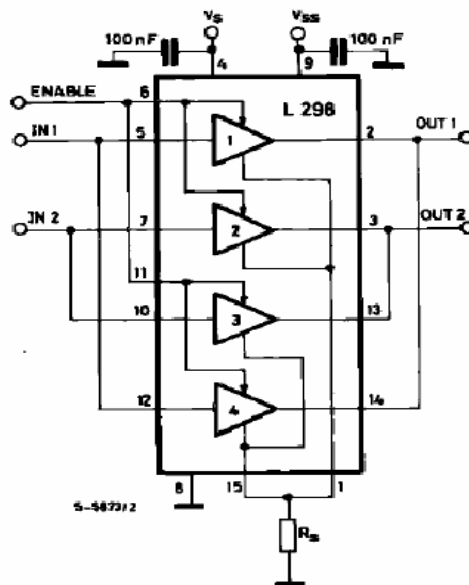
3.2.2. Mạch driver

L298N là một driver chip tích hợp sẵn hai mạch cầu H bên trong với chuẩn điều khiển TTL, không có diode nội bảo vệ Mosfet. Chịu tải tối đa trên mỗi cầu là 2A, điện áp 40VDC. Logic '0' ở ngõ vào lên tới 1.5V (khả năng khử nhiễu cao). Sử dụng dạng đóng gói Multiwatt15. Các thông số cần thiết khác của L298N có thể được tìm thấy tại trang web của hãng STMicroelectronics



Hình 3. 9 : Hình dạng IC L298N

Thiết kế một bộ điều khiển mở có thể sử dụng cho các động cơ lên đến 4A bằng cách nối song song 2 cầu H bên trong của L298N. Hình dạng của IC L298N được thể hiện như hình 3.5.



Hình 3. 10. Nối song song hai mạch cầu H của L298N

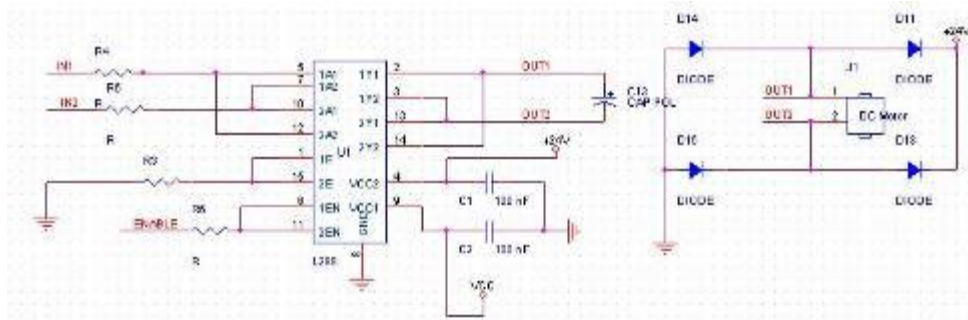
Việc hạn dòng cho động cơ là rất cần thiết, nhất là khi động cơ hoạt động, có rất nhiều tình huống không mong đợi sẽ xảy ra, vì vậy cần phải có chế độ hạn dòng bằng phần mềm. Trong các trường hợp nguy hiểm như ngắn mạch do va chạm, hoặc tuột dây nối, cần có mạch bảo vệ chống ngắn mạch bằng phần cứng để đáp ứng kịp thời.

Cần điều khiển PWM ở tần số cao để tránh tiếng ồn do động cơ gây ra, nhất là những tiếng kêu nghe rất rõ ở khoảng tần số 1KHz đến 3KHz. Bộ điều khiển PWM thông thường được dùng ở 5KHz. Tuy nhiên, trong ứng dụng này, ta dùng ở mức 10.8KHz.

Hai chế độ dừng của L298N rất thuận lợi. Dừng tự do (free stop) được thực hiện khi chân “enable” được kéo xuống mass. Dừng nhanh hay phanh (fast stop – break) được thực hiện khi hai ngõ vào có cùng logic. Trường hợp dừng tự do chỉ được dùng khi điện áp hồi tiếp đo dòng tải ở chân “sense” vượt quá cho phép và được dùng bởi chương trình điều khiển. Tất cả các trường hợp dừng khác đều dùng biện pháp dừng nhanh. Như vậy, trong trường hợp ngắn mạch, động cơ sẽ được duy trì ở trạng thái phanh.

L298N không có diode nội để bảo vệ các mosfet nằm bên trong, do đó, cần có các diode ngoài để bảo vệ (xem Mạch nguyên lý). Để đảm bảo điều khiển ở tần số PWM cao, cần dùng diode nhanh có điện trở thấp. Diode chuyên dụng để điều khiển động cơ là các diode fast recovery Schottky. Nhưng ở tần số thấp như trong ứng dụng này, vẫn có thể dùng loại 1N4007.

Một lưu ý rằng L298N khi hoạt động rất nóng, nhiệt độ có thể làm phỏng tay khi chạm vào L298N. Do vậy, cần có một miếng tản nhiệt để giải nhiệt cho L298N. Khi không có tản nhiệt, L298N sẽ nóng rất nhanh và tự động ngắt điều khiển. Trong các thí nghiệm, L298N sẽ ngắt mạch trong vòng 45 đến 60 giây khi không có tản nhiệt.

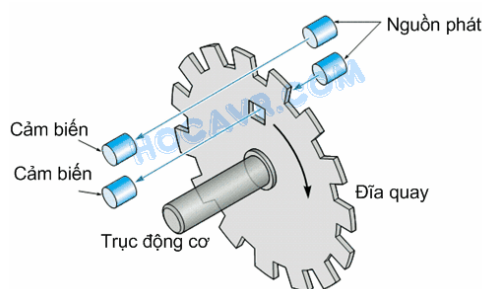


Hình 3. 11. Sơ đồ nguyên lý mạch công suất

3.2.3. Khối phản hồi tốc độ

Để điều khiển số vòng quay hay vận tốc động cơ thì chúng ta nhất thiết phải đọc được góc quay của motor. Một số phương pháp có thể được dùng để xác định góc quay của motor bao gồm tachometer (máy phát tốc), dùng biến trở xoay hoặc dùng encoder. Trong đó 2 phương pháp đầu tiên là phương pháp analog và dùng optical encoder (encoder quang) thuộc nhóm phương pháp digital. Hệ thống optical encoder bao gồm một nguồn phát quang (thường là hồng ngoại – infrared), một cảm biến quang và một đĩa có chia rãnh.

Optical encoder lại được chia thành 2 loại: encoder tuyệt đối (absolute optical encoder) và encoder tương đối (incremental optical encoder). Trong đa số các DC Motor, encoder tương đối được dùng và mô hình động cơ trong đồ án này cũng không ngoại lệ.



Hình 3. 12. Mô hình của encoder tương đối

Encoder thường có 3 kênh (3 ngõ ra) bao gồm kênh A, kênh B và kênh I (Index).

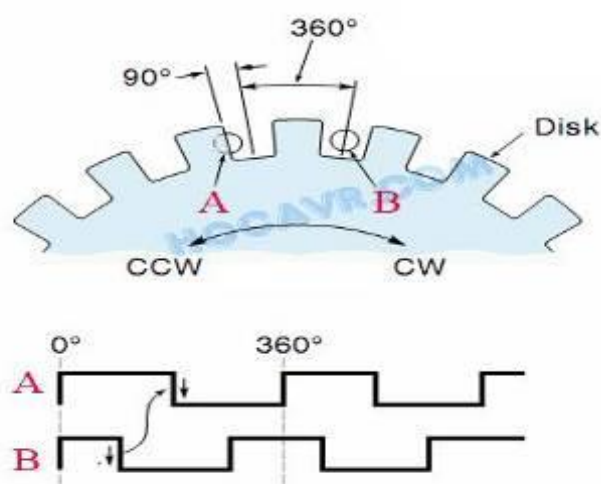
Kênh I: Trong hình trên ta thấy một lỗ nhỏ bên phía trong của đĩa quay và một cặp phát-thu dành riêng cho lỗ nhỏ này. Cứ mỗi lần motor quay được một vòng, lỗ nhỏ xuất hiện tại vị trí của cặp phát-thu, hồng ngoại từ nguồn phát sẽ

xuyên qua lỗ nhỏ đến cảm biến quang, một tín hiệu xuất hiện trên cảm biến. Như thế kênh I xuất hiện một “xung” mỗi vòng quay của motor.

Kênh A: bên ngoài đĩa quay được chia thành các rãnh nhỏ và một cặp thu-phát khác dành cho các rãnh này. Hoạt động của kênh A cũng tương tự kênh I, điểm khác nhau là trong 1 vòng quay của motor, có n “xung” xuất hiện trên kênh A với n là số rãnh trên đĩa và được gọi là độ phân giải (resolution) của encoder. Mỗi loại encoder có độ phân giải khác nhau, có khi trên mỗi đĩa chỉ có vài rãnh nhưng cũng có trường hợp đến hàng nghìn rãnh được chia. Để điều khiển động cơ, ta phải biết độ phân giải của encoder đang dùng. Độ phân giải ảnh hưởng đến độ chính xác điều khiển và cả phương pháp điều khiển. Trong đồ án sử dụng loại encoder có độ phân giải 200 xung/vòng.

Kênh B: trên các encoder còn có một cặp thu phát khác được đặt trên cùng đường tròn với kênh A nhưng lệch một chút (lệch $M+0,5$ rãnh). Tín hiệu xung từ kênh B có cùng tần số với kênh A nhưng lệch pha 90° . Bằng cách phối hợp kênh A và B người đọc sẽ biết chiều quay của động cơ.

Hình 3.13 thể hiện sự bố trí của 2 cảm biến kênh A và B lệch pha nhau. Khi cảm biến A bắt đầu bị che thì cảm biến B hoàn toàn nhận được hồng ngoại xuyên qua, và ngược lại. Hình thấp là dạng xung ngõ ra trên 2 kênh.



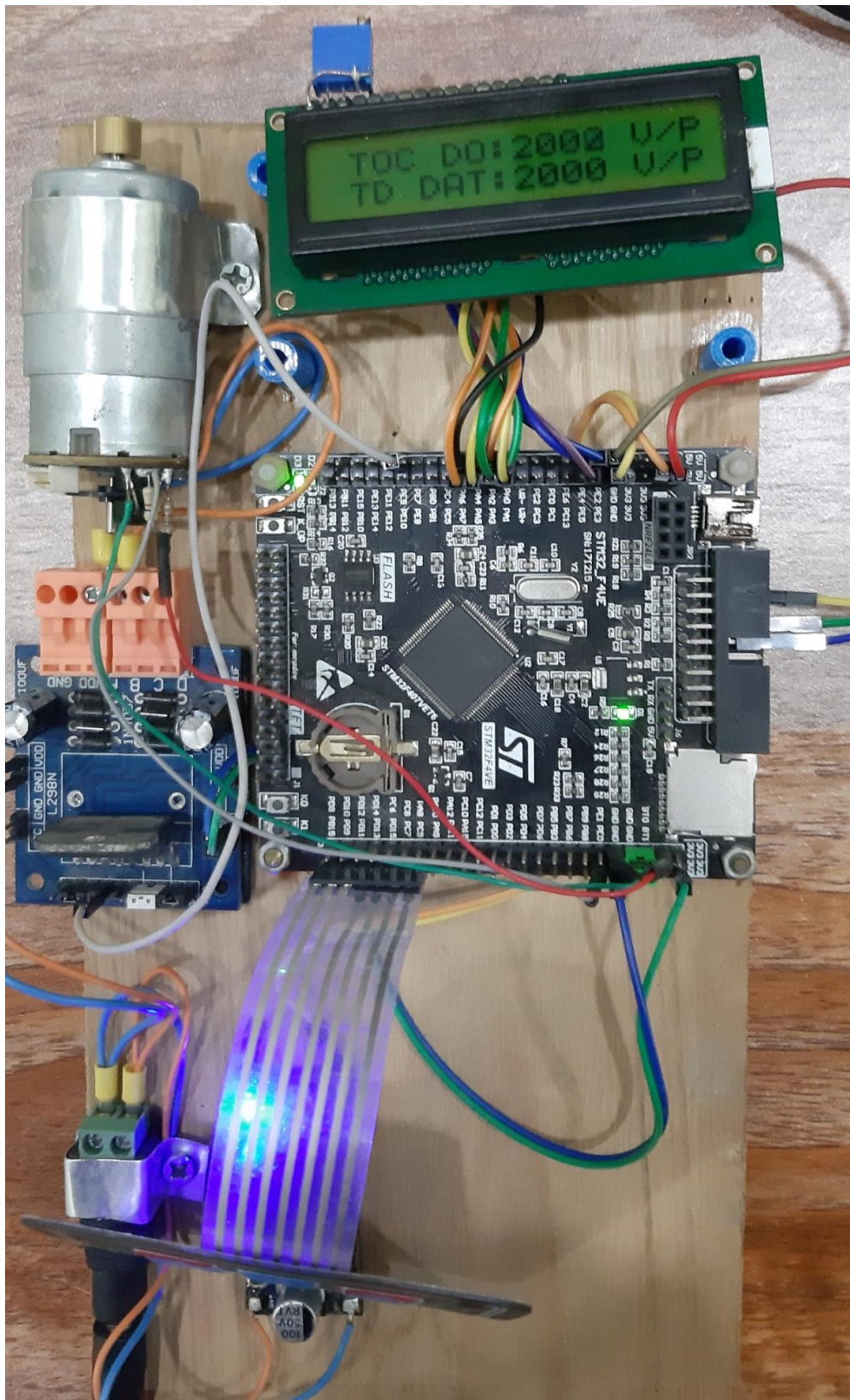
Hình 3. 13. Hai kênh A và B lệch pha trong encoder

Xét trường hợp motor quay cùng chiều kim đồng hồ, tín hiệu “đi” từ trái sang phải. Quan sát thấy lúc tín hiệu A chuyển từ mức cao mức xuống thấp (cạnh xuống) thì kênh B đang ở mức thấp.

Ngược lại, nếu động cơ quay ngược chiều kim đồng hồ, tín hiệu “đi” từ phải qua trái. Lúc này, tại cạnh xuống của kênh A thì kênh B đang ở mức cao. Như vậy, bằng cách phối hợp 2 kênh A và B chúng ta không những xác định được góc quay (thông qua số xung) mà còn biết được chiều quay của động cơ (thông qua mức của kênh B ở cạnh xuống của kênh A).

Phương pháp đọc encoder bằng STM32: Tùy theo mục đại lượng điều khiển (vị trí hay vận tốc) và đặc điểm encoder (độ phân giải) chúng ta có 3 phương pháp để đọc encoder bằng STM32: Input Capture, Timer với chức năng counter và ngắt ngoài. Mỗi phương pháp sử dụng đều có ưu và nhược điểm riêng. Tuy nhiên trong đồ án sử dụng phương pháp sử dụng Timer với chức năng counter. Đặt các kênh của encoder vào các chân đếm của Timer 1 ta sẽ đếm được số lượng xung của các kênh, sau đó chỉ chọn chế độ xung nguồn cấp của Timer 1. Mỗi khi có xung từ encoder, timer sẽ tăng giá trị lên 1. Đây là phương pháp sử dụng ít tài nguyên nhất (ít tốn thời gian cho encoder).

3.3. Hình ảnh thực tế của sơ đồ điều khiển



Hình 3. 14. Hình ảnh mạch sử dụng STM32F407

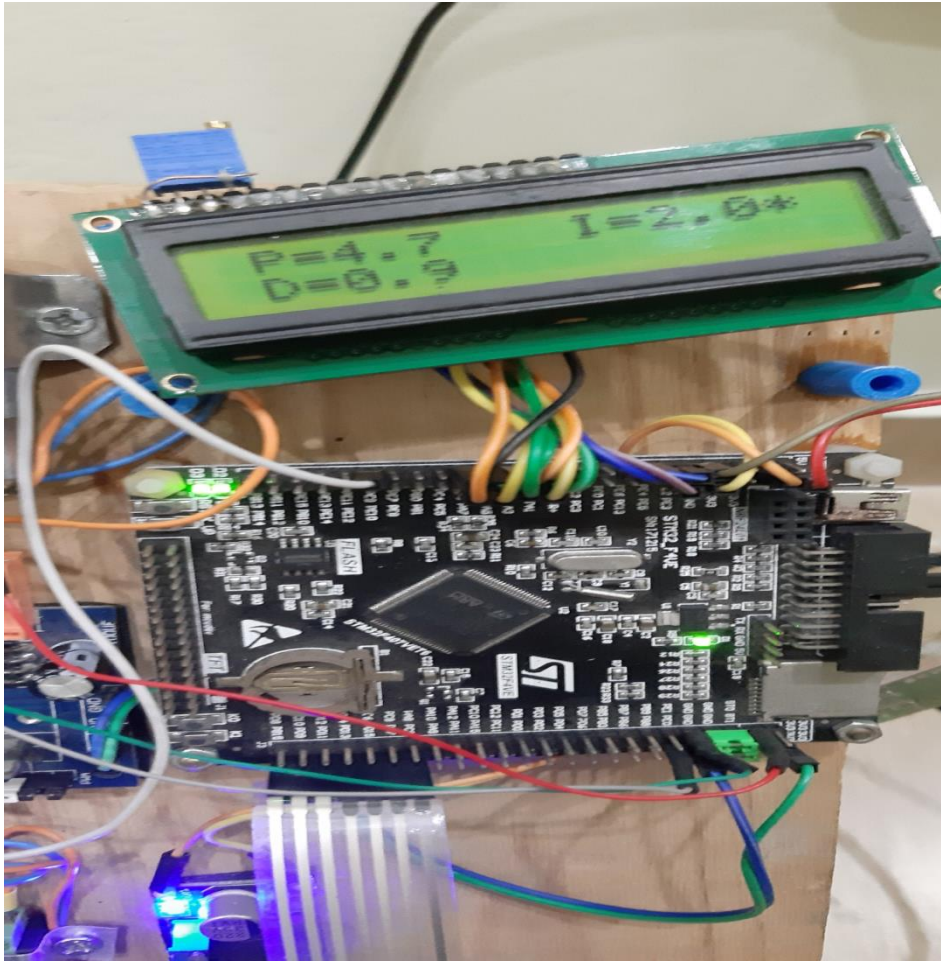
Ở sơ đồ mạch theo hình 3.14 được giải thích như sau:

TD Đặt (tốc độ đặt):

TOC DO (tốc độ thực):

Ở đây ta đặt tốc độ ở dòng TD Đặt với giá trị trong dải từ 0000 đến 3400 vòng/phút.

Hệ số K_p, K_i, K_d được lựa chọn theo kinh nghiệm của tác giả theo như hình 3.15:



Hình 3. 15. Khai báo các hệ số K_p, K_i, K_d

3.4. Phần code của STM32F407

```
/* USER CODE BEGIN Header */  
  
/**
```

* @file stm32f4xx_it.c

* @brief Interrupt Service Routines.

* @attention

*

* <h2><center>© Copyright (c) 2020 STMicroelectronics.

* All rights reserved.</center></h2>

*

* This software component is licensed by ST under BSD 3-Clause
license,

* the "License"; You may not use this file except in compliance with the

* License. You may obtain a copy of the License at:

* opensource.org/licenses/BSD-3-Clause

*

*/


```
/* USER CODE END Header */

/* Includes -----*/

#include "main.h"

#include "stm32f4xx_it.h"

/* Private includes -----*/

/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* Private typedef -----*/

/* USER CODE BEGIN TD */

/* USER CODE END TD */

/* Private define -----*/

/* USER CODE BEGIN PD */

/* USER CODE END PD */

/* Private macro -----*/

/* USER CODE BEGIN PM */
```

```
/* USER CODE END PM */
```

```
/* Private variables -----*/
```

```
/* USER CODE BEGIN PV */
```

```
/* USER CODE END PV */
```

```
/* Private function prototypes -----*/
```

```
/* USER CODE BEGIN PFP */
```

```
/* USER CODE END PFP */
```

```
/* Private user code -----*/
```

```
/* USER CODE BEGIN 0 */
```

```
/* USER CODE END 0 */
```

```
/* External variables -----*/
```

```
extern TIM_HandleTypeDef htim2;
```

```
extern TIM_HandleTypeDef htim3;
```

```
/* USER CODE BEGIN EV */
```

```
/* USER CODE END EV */
```

```
/**
 *
 */
```

```
/*          Cortex-M4 Processor Interruption and Exception Handlers
 */
```

```
/**
 *
 */
```

```
/**
```

```
 * @brief This function handles Non maskable interrupt.
```

```
 */
```

```
void NMI_Handler(void)
```

```
{
```

```
/* USER CODE BEGIN NonMaskableInt_IRQn 0 */
```

```
/* USER CODE END NonMaskableInt_IRQn 0 */
```

```
/* USER CODE BEGIN NonMaskableInt_IRQn 1 */
```

```
/* USER CODE END NonMaskableInt_IRQn 1 */
```

```
}
```

```

/**
 * @brief This function handles Hard fault interrupt.
 */

void HardFault_Handler(void)
{
    /* USER CODE BEGIN HardFault_IRQn 0 */

    /* USER CODE END HardFault_IRQn 0 */

    while (1)
    {
        /* USER CODE BEGIN W1_HardFault_IRQn 0 */

        /* USER CODE END W1_HardFault_IRQn 0 */

    }
}

/**
 * @brief This function handles Memory management fault.
 */

void MemManage_Handler(void)
{

```

```

/* USER CODE BEGIN MemoryManagement_IRQn 0 */

/* USER CODE END MemoryManagement_IRQn 0 */

while (1)

{

/* USER CODE BEGIN W1_MemoryManagement_IRQn 0 */

/* USER CODE END W1_MemoryManagement_IRQn 0 */

}

}

/**

* @brief This function handles Pre-fetch fault, memory access fault.

*/

void BusFault_Handler(void)

{

/* USER CODE BEGIN BusFault_IRQn 0 */

/* USER CODE END BusFault_IRQn 0 */

while (1)

{

/* USER CODE BEGIN W1_BusFault_IRQn 0 */

```

```

/* USER CODE END W1_BusFault_IRQn 0 */

}

}

/**

* @brief This function handles Undefined instruction or illegal state.

*/

void UsageFault_Handler(void)

{

/* USER CODE BEGIN UsageFault_IRQn 0 */

/* USER CODE END UsageFault_IRQn 0 */

while (1)

{

/* USER CODE BEGIN W1_UsageFault_IRQn 0 */

/* USER CODE END W1_UsageFault_IRQn 0 */

}

}

/**

* @brief This function handles System service call via SWI instruction.

```

```

*/

void SVC_Handler(void)

{

/* USER CODE BEGIN SVCcall_IRQn 0 */

/* USER CODE END SVCcall_IRQn 0 */

/* USER CODE BEGIN SVCcall_IRQn 1 */

/* USER CODE END SVCcall_IRQn 1 */

}

/**

* @brief This function handles Debug monitor.

*/

void DebugMon_Handler(void)

{

/* USER CODE BEGIN DebugMonitor_IRQn 0 */

/* USER CODE END DebugMonitor_IRQn 0 */

/* USER CODE BEGIN DebugMonitor_IRQn 1 */

```

```

/* USER CODE END DebugMonitor_IRQn 1 */

}

/**

 * @brief This function handles Pendable request for system service.

 */

void PendSV_Handler(void)

{

/* USER CODE BEGIN PendSV_IRQn 0 */

/* USER CODE END PendSV_IRQn 0 */

/* USER CODE BEGIN PendSV_IRQn 1 */

/* USER CODE END PendSV_IRQn 1 */

}

/**

 * @brief This function handles System tick timer.

 */

void SysTick_Handler(void)

{

```



```
/* USER CODE BEGIN SysTick_IRQn 0 */
```

```
/* USER CODE END SysTick_IRQn 0 */
```

```
HAL_IncTick();
```

```
/* USER CODE BEGIN SysTick_IRQn 1 */
```

```
/* USER CODE END SysTick_IRQn 1 */
```

```
}
```

```
/*  
*****  
*****  
*/
```

```
*****  
*/
```

```
/* STM32F4xx Peripheral Interrupt Handlers */
```

```
/* Add here the Interrupt Handlers for the used peripherals. */
```

```
/* For the available peripheral interrupt handler names, */
```

```
/* please refer to the startup file (startup_stm32f4xx.s). */
```

```
/*  
*****  
*****  
*/
```

```
*****  
*/
```

```
/**
```

```
 * @brief This function handles EXTI line0 interrupt.
```

```
 */
```

```
void EXTI0_IRQHandler(void)
```

```

{

/* USER CODE BEGIN EXTI0_IRQn 0 */

/* USER CODE END EXTI0_IRQn 0 */

HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);

/* USER CODE BEGIN EXTI0_IRQn 1 */

/* USER CODE END EXTI0_IRQn 1 */

}

/**
 * @brief This function handles TIM2 global interrupt.
 */
void TIM2_IRQHandler(void)
{
/* USER CODE BEGIN TIM2_IRQn 0 */

/* USER CODE END TIM2_IRQn 0 */

HAL_TIM_IRQHandler(&htim2);

/* USER CODE BEGIN TIM2_IRQn 1 */

```

```

/* USER CODE END TIM2_IRQn 1 */

}

/**
 * @brief This function handles TIM3 global interrupt.
 */

void TIM3_IRQHandler(void)
{
/* USER CODE BEGIN TIM3_IRQn 0 */

/* USER CODE END TIM3_IRQn 0 */

HAL_TIM_IRQHandler(&htim3);

/* USER CODE BEGIN TIM3_IRQn 1 */

/* USER CODE END TIM3_IRQn 1 */

}

/* USER CODE BEGIN 1 */

/* USER CODE END 1 */

```

/***** (C) COPYRIGHT STMicroelectronics

*****/

Kết luận

Nghiên cứu ban đầu cho đã cho thấy kết quả khả quan tạo tiền đề về ứng dụng với STM32F4 . Đề tài phát triển này tôi xin đưa ra một số ưu điểm sau :

❖ Ưu nhược điểm :

1. Ưu điểm :

Giá thành rẻ so với các dòng chip khác

Tốc độ xử lý cao, ổn định

Tiết kiệm năng lượng

Số tài nguyên phù hợp với các ứng dụng khác nhau

2. Nhược điểm

Nhiều thanh ghi câu lệnh khá dài , gây khó nhớ cho người dùng , dễ nhầm lẫn

Thị trường ở Việt Nam chưa mở rộng , gây khó trong việc tìm kiếm tài liệu và khó khăn trong việc đặt mua chip , do vậy nghiên cứu này chưa được sâu

Sau thời gian 3 tháng làm đồ án với sự hướng dẫn tận tình của thầy giáo TS. Ngô Quang Vĩ , em đã hoàn thành đề tài được giao .Quá trình thực hiện đồ án đã giúp em củng cố lại kiến thức mà mình đã học , ngoài ra quá trình tìm hiểu đồ án đã giúp em có thêm kiến thức thực tế . Do thời gian làm đồ án ngắn và kiến thức còn hạn chế nên trong đồ án còn nhiều thiếu sót nhất định . Vì vậy em rất mong được sự góp ý , bổ sung của các thầy cô giáo để đồ án em được hoàn thiện hơn .

Em xin chân thành cảm ơn !

Hải phòng , ngày.....tháng.....năm 2020

Sinh viên

Đinh Hữu Mạnh

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. Vũ Gia Hanh (*Trần Khánh Hà, Phan Tử Thu, Nguyễn Văn Sáu*)(2006). “*Máy điện tập 2*”. Nhà xuất bản KHKT.
2. Bùi Quốc Khánh (*Nguyễn Văn Liễn, Nguyễn Thị Hiền*)(2006). “*Truyền Động điện*”. Nhà xuất bản KHKT.
3. Nguyễn Phùng Quang (2004). “*Matlab dùng cho kỹ sư điều khiển tự động*”. Nhà xuất bản KHKT.
4. GS.TSKH.Thân Ngọc Hoàn /TS.Nguyễn Trọng Thắng (2016).“ *Nguyên lý hoạt động của máy điện* ”. Nhà xuất bản xây dựng.
5. Kiều Xuân Thực/Vũ Thị Thi Hương/Vũ Trung Kiên.“*Vi điều khiển*” . Nhà xuất bản giáo dục Việt Nam.
6. Nguyễn Văn Hóa .“*Cơ sở lý thuyết điều khiển tự động*”. Nhà xuất bản KHKT.
7. Nguyễn Duy Anh .“*Lý thuyết điều khiển hiện đại*”. Nhà xuất bản KHKT.
8. Châu Ngọc Thạch/Trịnh Xuân Thu. “*Hướng dẫn tự lắp ráp 100 mạch điện - điện tử*”. Nhà xuất bản KHKT.
- 9.TS. Nguyễn Vũ Quỳnh/KS. Phạm Quang Huy. “*Giáo trình điện tử thực hành*”. Nhà xuất bản thanh niên.
- 10.TS. Nguyễn Vũ Quỳnh/KS. Phạm Quang Huy. “*Giáo trình vi điều khiển ARM – Hướng dẫn sử dụng STM32*”. Nhà xuất bản thanh niên.

Một số trang Web:

11. <http://www.dientuvietsam.net/forums>
12. <http://www.thaieasyelec.net>

13. <http://www.tailieu.vn>

14. <https://hochatlamthat.edu.vn>

15. <http://luanvan.net.vn>

16. <https://123doc.net>

Mục lục

Lời mở đầu	1
Chương 1 Giới thiệu dòng vi điều khiển STM32	2
1.1. Khái niệm về Vi xử lý và Vi điều khiển	2
1.1.1. Vi xử lý.....	2
1.1.2. Vi điều khiển(microcontroller)	3
1.2. Đặc điểm nổi bật của STM32	4
1.2.1. Sự tinh vi	5
1.2.2. Sự an toàn.....	6
1.2.3. Tính bảo mật.....	6
1.2.4. Phát triển phần mềm.....	7
1.2.5. Dòng Performance và Access của STM32.....	7
Chương 2 Các phương pháp điều khiển động cơ 1 chiều	9
2.1. Tổng quan về động cơ điện một chiều.....	9
2.1.1. Cấu tạo động cơ điện một chiều	9
2.1.2. Nguyên lý làm việc và phân loại động cơ điện một chiều.....	11
2.1.3. Phương trình đặc tính cơ động cơ điện một chiều.....	12
2.2. Các phương pháp điều khiển tốc độ động cơ điện một chiều.....	15
2.2.1. Phương pháp điều khiển tốc độ bằng cách thay đổi điện trở phụ phản ứng	15
2.2.2. Phương pháp điều khiển tốc độ động cơ bằng cách thay đổi từ thông kích từ	17
2.2.3. Phương pháp điều khiển tốc độ động cơ bằng cách thay đổi điện áp phản ứng.....	18
Chương 3 Thiết kế và lập trình chương trình điều khiển động cơ 1 một chiều trên STM32F407.....	21
3.1. Các thành phần cơ bản của hệ thống điều khiển	21
3.1.1. Sơ đồ điều khiển sử dụng bộ điều khiển PID.....	21
3.1.2. Một số bộ điều khiển khác	23
3.2. Thiết kế phần cứng	23
3.2.1. Khối mạch cấp nguồn.....	23

Mạch cấp nguồn 12VDC cho động cơ: do chất lượng điện áp cấp cho động cơ không đòi hỏi qua cao nên ta sử dụng một mạch đơn giản như sau:.....	24
3.2.2. Mạch driver	25
3.2.3. Khối phản hồi tốc độ	28
3.3. Hình ảnh thực tế của sơ đồ điều khiển	30
3.4. Phần code của STM32F407.....	32
TÀI LIỆU THAM KHẢO.....	47