

Đồ Án Môn học

**ỨNG DỤNG PID ĐIỀU KHIỂN TỐC ĐỘ
ĐỘNG CƠ DC SERVO HIỂN THỊ
LABVIEW SỬ DỤNG STM32F407VG**

SVTH: LÊ TẤN ĐẠT

MSSV: 0309191016

Bộ Môn Tự Động Hóa

Khoa Điện-Điện Lạnh

Trường Cao Đẳng Kỹ Thuật Cao Thắng

Tháng 11 năm 2022

ỨNG DỤNG PID ĐIỀU KHIỂN TỐC ĐỘ ĐỘNG CƠ DC SERVO HIỂN THỊ LABVIEW SỬ DỤNG STM32F407VG

Giáo viên hướng dẫn: **Th.S Cù Minh Phước**

Giáo viên hướng dẫn: **T.S Đặng Đức Chi**

Sinh viên thực hiện: **Lê Tấn Đạt**

MSSV: **0309191016**

NHIỆM VỤ ĐỒ ÁN

ỨNG DỤNG PID ĐIỀU KHIỂN TỐC ĐỘ ĐỘNG CƠ DC SERVO HIỂN THỊ LABVIEW SỬ DỤNG STM32F407VG.

ĐỀ BÀI

Thiết kế và thi công mô hình điều khiển tốc độ động cơ hoạt động theo yêu cầu

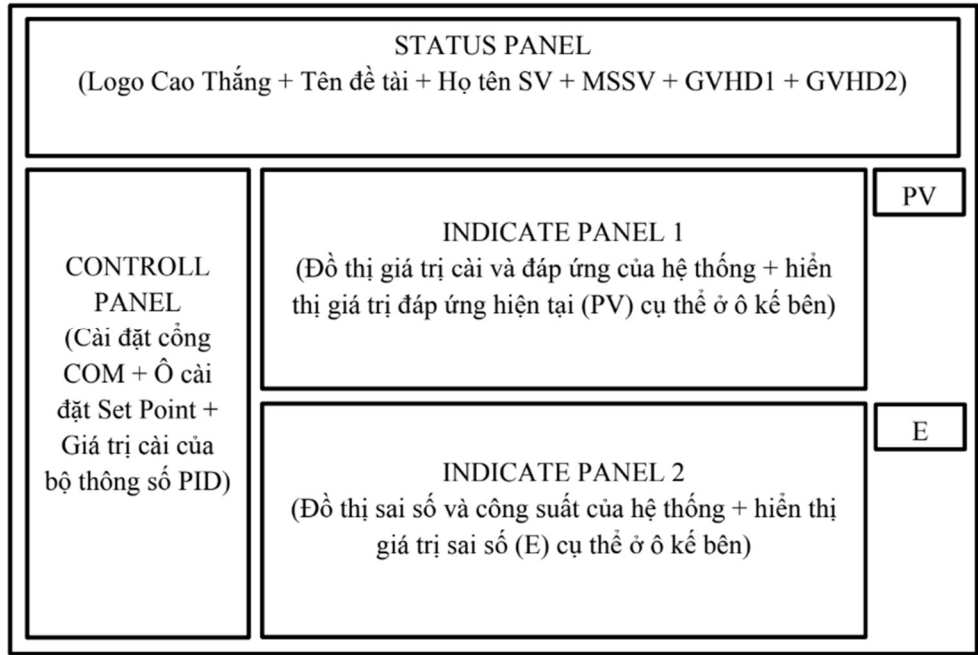
Bảng 1 và Bảng 2 với các thông số kỹ thuật sau:

- Nguồn sử dụng: 220 [VAC]
- Động cơ sử dụng điện áp 12 [VDC] hoặc 24 [VDC], có công suất từ [30 250][W]
- Dây điều khiển: [-800 800] [vòng/ phút]
- Cảm biến: Dùng encoder từ 200 [xung/vòng] trở lên
- Vi điều khiển: STM32F407VG
- Hiển thị giao diện LabVIEW
- Sử dụng 4 nút nhấn: BT1, BT2, BT3.
- Có để hoặc hộp cố định vị trí của các thiết bị sử dụng trong đề tài.

Bảng 1: Lập trình vi điều khiển STM32F407VG theo các yêu cầu sau:

STT	Nội dung
1	Cấp nguồn, đèn báo nguồn 5[VDC] sáng.
2	Cập nhật giá trị mới lên LabVIEW thỏa chu kỳ cập nhật $T_{ht}=[0.01 \ 0.2]$ [giây]
3	Sử dụng nút nhấn BT1, BT2 để tăng, giảm giá trị đặt 100 [vòng/phút] cho một lần tác động.
4	Sử dụng nút nhấn BT3 để reset về trạng thái ban đầu của chương trình, với tốc độ đặt mặc định là 0 [vòng/phút]
5	Quy ước chiều quay thuận là chiều ngược chiều kim đồng hồ (lúc này tốc độ mang dấu dương), chiều quay nghịch là chiều thuận chiều kim đồng hồ (lúc này tốc độ mang dấu âm)
6	<p>Chất lượng đáp ứng trên toàn dãy điều khiển của bộ điều khiển PID phải thỏa yêu cầu sau:</p> <p>POT : [0 10] [%]</p> <p>e_{xl} : [-20 20] [vòng] / [phút]</p> <p>T_r : [0.5 4] [giây]</p> <p>T_{xl} : [1 6] [giây]</p>

Bảng 2: Yêu cầu lập trình LabVIEW. Giao diện có nội dung và bố cục chi tiết như sau:



ỨNG DỤNG PID ĐIỀU KHIỂN TỐC ĐỘ ĐỘNG CƠ DC SERVO HIỂN THỊ LABVIEW SỬ DỤNG STM32F407VG

Sinh viên thực hiện: Lê Tấn Đạt

MSSV: 0309191016

Hội Đồng Chấm Bảo Vệ:

(Trưởng Ban)

(Thành Viên)

(Thành Viên)

(Thành Viên)

(Thành Viên)

(Thành Viên)

(Thành Viên)

Tháng 11 năm 2022

[illegible]

7

Lời Cảm Ơn

Để đồ án này được đạt kết quả tốt như hiện nay em đã nhận được rất nhiều sự hỗ trợ và hướng dẫn của thầy Cù Minh Phước. Xuất phát từ sự chân thành, em xin bày tỏ sự biết ơn sâu sắc nhất đến thầy.

Bên cạnh đó, em cũng xin gửi lời cảm ơn các thầy cô giáo và nhà trường cùng với các anh chị khóa trên đã tạo điều kiện tốt nhất để em có được môi trường học tập tốt nhất.

Em xin kính chúc quý thầy, quý cô, quý nhà trường luôn luôn mạnh khỏe. Kính chúc cho chặng đường lái đò của quý thầy cô được thuận buồm xuôi gió!

Em xin chân thành cảm ơn!

Mục Lục

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN.....	7
LỜI CẢM ƠN	8
MỤC LỤC	9
CÁC KÝ HIỆU	13
TÓM TẮT	14
CHƯƠNG 1 GIỚI THIỆU.....	16
1.1 Giới thiệu.....	16
1.2 Mục tiêu của đề tài.....	16
1.3 Cấu trúc của quyển đồ án	17
1.4 Kế hoạch thực hiện	17
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	18
2.1 Lý thuyết cơ bản	18
2.1.1 Tổng quan về bộ điều khiển PID	18
2.1.2 Lý thuyết về bộ điều khiển PID	19
2.1.3 Phương pháp điều xung PWM.....	22
2.1.4 Phương pháp đọc xung encoder.....	24
2.1.5 Nguyên lý hoạt động mạch cầu H.....	26
2.2 Các công cụ và thiết bị	28
2.2.1 Giới thiệu KIT STM32F4 Discovery.....	28
2.2.2 Động cơ DC Servo NF5475E với Encoder.....	29
2.2.3 Mạch cầu H-HI216	30
2.2.4 Mạch nguồn ổn áp sử dụng IC L7805	31
2.2.5 Phần mềm MATLAB.....	31
2.2.6 Phần mềm LABVIEW	32
CHƯƠNG 3 THIẾT KẾ SƠ ĐỒ.....	34
3.1 Sơ đồ khối tổng quát của hệ thống	34
3.2 Sơ đồ nguyên lý.....	34
3.2.1 Sơ đồ nguồn	34
3.2.2 Sơ đồ khối công suất.....	35
3.2.3 Sơ đồ khối encoder	36
3.2.4 Sơ đồ khối UART	36
3.2.5 Sơ đồ khối nút nhấn	37
3.2.6 Sơ đồ khối xử lý trung tâm	37
3.2.7 Sơ đồ khối tín hiệu điều khiển	38
3.3 Sơ đồ thiết kế PCB	38
CHƯƠNG 4 CHƯƠNG TRÌNH ĐIỀU KHIỂN	41
4.1 Lưu đồ giải thuật của chương trình chính.....	41
4.2 Lưu đồ giải thuật của chương trình nút nhấn.....	42

4.3	Lưu đồ giải thuật của chương trình UART.....	43
4.4	Lưu đồ giải thuật của chương trình đọc encoder	44
4.5	Lưu đồ giải thuật của chương trình tính PID.....	45
4.6	Lưu đồ giải thuật của chương trình đọc xuất tín hiệu điều khiển.....	46
	CHƯƠNG 5 KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN	47
5.1	Kết quả đạt được.....	47
5.1.1	Kết quả thiết kế thi công phần cứng	47
5.1.2	Kết quả thiết kế thi công phần mềm	47
5.1.3	Kết quả vận hành	48
5.2	Nhận xét về kết quả	48
5.3	Hướng phát triển.....	49
	TÀI LIỆU THAM KHẢO.....	49
	Chương trình tham khảo	50

Danh Sách Các Hình

Hình 2. 1 Sơ đồ khối bộ điều khiển PID	18
Hình 2. 2 Sơ đồ khối thiết kế bộ điều khiển PID dùng phương pháp Zeigler-Nichols 2.....	21
Hình 2. 3 Đáp ứng của hệ kín ở biên giới ổn định	22
Hình 2. 4 Xung PWM.....	23
Hình 2. 5 Các thành phần của bộ mã hóa quang	25
Hình 2. 6 Tín hiệu đầu ra A và B của bộ mã hóa cầu phương	26
Hình 2. 7 Cấu tạo cầu H	27
Hình 2. 8 Nguyên lý hoạt động cầu H.....	28
Hình 2. 9 KIT STM32F407VG Discovery	28
Hình 2. 10 Hình dáng bên ngoài động cơ NF5475E	29
Hình 2. 11 Kích thước động cơ NF5475E.....	30
Hình 2. 12 Cách đấu nối mạch cầu H-Hi216.....	30
Hình 2. 13 Phần mềm MATLAB	32
Hình 2. 14 Phần mềm LABVIEW	33
Hình 3. 1 Sơ đồ khối tổng quát.....	34
Hình 3. 2 Hình ảnh thiết kế PCB 3D	39
Hình 3. 3 Hình ảnh thiết kế đi dây PCB	40

Danh Sách Các Bảng

Bảng 1. 1 Bảng kế hoạch thực hiện	17
Bảng 2. 1 Phương pháp Ziegler-Nichols	22

Các Từ Viết Tắt

VDK	Vi điều khiển
GUI	Graphical User Interface

Các Ký hiệu

Ký hiệu	Ý nghĩa	Đơn vị
K_p :	Hệ số tỷ lệ bộ điều khiển PID	
K_i :	Hệ số tích phân bộ điều khiển PID	
K_d :	Hệ số đạo hàm bộ điều khiển PID	
<i>Setpoint</i> :	Giá trị điều khiển mong muốn	
e :	Tín hiệu sai số	
e_{ss} :	Sai số xác lập	

ỨNG DỤNG PID ĐIỀU KHIỂN TỐC ĐỘ ĐỘNG CƠ DC SERVO HIỂN THỊ LABVIEW SỬ DỤNG STM32F407VG

SVTH: LÊ TẤN ĐẠT

Ngành Công Nghệ Điều Khiển và Tự Động Hóa

Khoa Điện - Điện Tử

Trường Cao Đẳng Kỹ Thuật Cao Thắng

Tóm Tắt

Điều khiển tốc độ động cơ DC bằng bộ điều khiển PID là thuật toán điều khiển phổ biến nhất được sử dụng trong công nghiệp và đã được chấp nhận rộng rãi trong điều khiển công nghiệp. Một trong những ứng dụng được sử dụng ở đây là điều khiển tốc độ của động cơ DC. Kiểm soát tốc độ của động cơ DC là rất quan trọng vì bất kỳ thay đổi nhỏ nào cũng có thể dẫn đến sự mất ổn định của hệ thống vòng kín.

Mục đích của đồ án này là thiết kế bộ điều khiển tốc độ động cơ DC bằng cách sử dụng bộ điều khiển PID với giao diện LABVIEW.

DC Motor sẽ được giao tiếp với LabVIEW bằng Vi điều khiển STM32F407VG. Tốc độ của động cơ DC sẽ được thiết lập bằng cách tạo Giao diện người dùng đồ họa (GUI) cho Bộ điều khiển PID trong LabVIEW. LabVIEW sẽ gửi lệnh nối tiếp đến động cơ DC bằng các chân PWM trên bo mạch Vi điều khiển. Động cơ DC sẽ di chuyển với tốc độ do người dùng thiết lập trong LabVIEW. Tốc độ của động cơ dc sẽ được cảm nhận bằng cách sử dụng cảm biến Encoder. Từ cảm biến, đầu ra được gửi trở lại Bộ điều khiển PID trong VDK. Bộ điều khiển PID so sánh tốc độ thực tế của động cơ DC với tốc độ cài đặt. Nếu tốc độ của nó không giống nhau, bộ điều khiển PID sẽ cố gắng giảm thiểu sai số và đưa động cơ về giá trị điểm đặt.

Khi cung cấp một nguồn điện, một động cơ DC sẽ bắt đầu quay cho đến khi điện được loại bỏ. Hầu hết các động cơ DC chạy ở tốc độ RPM cao (số vòng quay

mỗi phút), ví dụ như làm mát máy tính quạt, hoặc bánh xe ô tô điều khiển bằng radio. Tốc độ của động cơ DC được điều khiển bằng điều chế độ rộng xung(PWM). DC Motor sẽ được giao tiếp với LabVIEW sử dụng Vi điều khiển STM32.

Vai trò của Vi điều khiển là chuyển tốc độ đã đặt cho động cơ DC sử dụng các chân PWM và để lấy dữ liệu (tốc độ) từ động cơ sử dụng cảm biến Encoder.

Ý tưởng cơ bản đằng sau bộ điều khiển PID là đọc một cảm biến, sau đó tính toán đầu ra bộ truyền động mong muốn bằng cách tính toán phản ứng tỷ lệ, tích phân và đạo hàm và tổng hợp ba thành phần đó để tính toán đầu ra.

Để điều khiển và giám sát được động cơ bằng một giao diện trên phần mềm LabVIEW, qua phần mềm này thì người dùng có thể gửi các giá trị tham số hệ thống và tốc độ chạy mong muốn xuống vi điều khiển STM32F407 để vi điều khiển này xuất tín hiệu điều khiển động cơ DC.

Dựa trên sự hướng dẫn từ giáo viên hướng dẫn, các kiến thức học được về Lý thuyết điều khiển tự động, Đo lường điều khiển trên máy tính, Vi điều khiển và một phần kiến thức từ internet đã đúc kết kiến thức để thực hiện đề tài.

Sau quá trình làm thì kết quả thu được đúng theo như mong muốn ban đầu là điều khiển động cơ chạy theo đáp ứng tốt. Không những vậy mà còn tiếp xúc được kiến thức mới về viết chương trình điều khiển STM32 bằng phần mềm MATLAB.

CHƯƠNG 1 GIỚI THIỆU

1.1 Giới thiệu

Động cơ điện một chiều vẫn còn phù hợp trong ngành công nghiệp hiện đại, mặc dù chúng là một trong những thiết kế động cơ điện lâu đời nhất lý do chính là bởi vì chúng khả năng điều khiển tốt của chúng. Cổ máy đơn giản này biến đổi dòng điện một chiều thành quay cơ học, có thể điều khiển bằng cách đơn giản là thay đổi điện áp đầu vào hoặc đảo chiều các dây dẫn của nó. Sự đơn giản của động cơ DC đã dẫn đến nhiều bộ điều khiển động cơ DC, thường được thiết kế đơn giản và cung cấp hiệu suất phù hợp với giá thành của chúng.

Đối với lĩnh vực điều khiển tự động hiện đại, điều khiển chính xác là một phần không thể thiếu. Phần lớn các loại máy móc, thiết bị dân dụng hay công nghiệp sử dụng động cơ điện, từ động cơ điện trong các máy công cụ, máy CNC, cánh tay robot,... đến những thiết bị như máy giặt, điều hòa, máy vi tính. Tất cả đều yêu cầu độ chính xác cao, tiết kiệm năng lượng, tuổi thọ, chu kỳ bảo dưỡng dài. Để đáp ứng được những yêu cầu đó yêu cầu bộ điều khiển tốc độ động cơ một cách ổn định, đáp ứng nhanh, vận hành trơn tru khi xác lập và chuyển trạng thái.

Việc ứng dụng các thuật toán kinh điển vào vấn đề điều khiển tốc độ động cơ đã đạt được nhiều kết quả khả quan. Ví dụ như sử dụng bộ điều khiển PI, PID cho kết quả tốt ở một số động cơ. Trong khuôn khổ đề tài, em xin phép trình bày về thuật toán PID, xây dựng bộ điều khiển trên nền vi điều khiển STM32F407VG, các kết quả thu được và các hướng phát triển đề tài.

1.2 Mục tiêu của đề tài

Việc hoàn thành đề tài có ý nghĩa rất lớn về mặt kiến thức thu được về điều khiển hệ thống và giải thuật điều khiển cũng như lựa chọn thiết kế một hệ thống điện tử. Về mô hình điều khiển tốc độ động cơ DC Servo có thể ứng dụng vào rất nhiều hệ thống trong thực tế một cách tự động và chính xác có thể điều khiển bằng giao diện LABVIEW.

Các phương pháp thực hiện đề tài là:

- Tính toán tốc độ động cơ từ số xung đọc về từ encoder
- Ứng dụng phương pháp điều độ rộng xung PWM
- Ứng dụng thuật toán điều khiển PID tốc độ động cơ
- Ứng dụng phần mềm LABVIEW vào điều khiển giám sát

- Ứng dụng viết chương trình điều khiển bằng phần mềm MATLAB.

1.3 Cấu trúc của quyển đồ án

Cấu trúc quyển báo cáo này gồm 5 chương

- Chương 1 Giới thiệu
- Chương 2 Cơ sở lý thuyết
- Chương 3 Thiết kế sơ đồ
- Chương 4 Chương trình điều khiển
- Chương 5 Kết quả và hướng phát triển
- Tài liệu tham khảo

1.4 Kế hoạch thực hiện

Bảng 1. 1 Bảng kế hoạch thực hiện

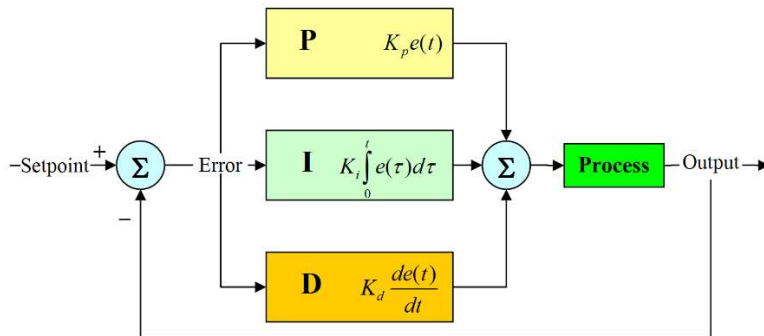
Thời gian	Công việc
07/10/2022	Nhận đề tài
8/10/2022 – 15/10/2022	Tìm hiểu về yêu cầu và phương pháp thực hiện đề tài và lựa chọn và tìm mua thiết bị
16/10/2022 – 23/10/2022	Thi công mô hình thử nghiệm
24/10/2022 – 31/10/2022	Viết chương trình điều khiển
1/11/2022 – 15/11/2022	Thiết kế giao diện người dùng
16/11/2022 – 27/11/2022	Thiết kế thi công mạch PCB và hoàn thiện mô hình
27/11/2022 – 5/12/2022	Viết báo cáo và nội dung trình chiếu
5/12/2022 – 10/12/2022	Chuẩn bị báo cáo

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1 Lý thuyết cơ bản

2.1.1 Tổng quan về bộ điều khiển PID

Bộ điều khiển PID là một bộ điều khiển được sử dụng rộng rãi trong nhiều lĩnh vực bởi có tính chính xác, tốc độ đáp ứng cao.



Hình 2. 1 Sơ đồ khối bộ điều khiển PID

Bộ điều khiển PID (bộ điều khiển vi tích phân tỉ lệ) là bộ điều khiển tổng quát được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp – bộ điều khiển PID là bộ điều khiển được sử dụng nhiều nhất trong các bộ điều khiển phản hồi. Bộ điều khiển PID sẽ tính toán giá trị "sai số" là hiệu số giữa giá trị đặt mong muốn và giá trị đo thông số biến đổi. Trong trường hợp không có mô hình toán học về hệ thống điều khiển thì bộ điều khiển PID cũng sẽ là một sự lựa chọn tốt. Tuy nhiên, để đạt được kết quả tốt nhất, các thông số của bộ điều khiển PID sử dụng trong tính toán phải điều chỉnh theo tính chất của hệ thống.

Bộ điều khiển PID bao gồm 3 thông số riêng biệt, do đó đôi khi nó còn được gọi là điều khiển ba khâu: các giá trị tỉ lệ, tích phân, vi phân, viết tắt là P, I, và D. Giá trị tỉ lệ xác định tác động của sai số hiện tại, giá trị tích phân xác định tác động của tổng các sai số quá khứ, và giá trị vi phân xác định tác động của tốc độ biến đổi sai số hiện tại. Tổng của ba khâu này dùng để điều chỉnh quá trình thông qua một phần tử điều khiển như vị trí của van điều khiển hay bộ nguồn của phần tử gia nhiệt. Khâu P phụ thuộc vào sai số hiện tại, Khâu I phụ thuộc vào tích lũy các sai số quá khứ cho đến chu kỳ tính toán hiện tại, khâu D phụ thuộc vào tốc độ thay đổi của sai số hiện tại.

Bằng cách điều chỉnh 3 hệ số tỉ lệ trong giải thuật của bộ điều khiển PID, bộ điều khiển có thể dùng trong những thiết kế có yêu cầu đặc biệt. Vài ứng dụng có thể yêu cầu có thể lược bỏ một hoặc hai khâu tùy theo hệ thống.

Bộ điều khiển PID có 2 dạng là dạng song song và dạng lý tưởng. Trong đề tài này ta sử dụng PID dạng song song:

Công thức tổng quát của bộ điều khiển PID dạng lý tưởng:

$$u_{PID}(t) = K_p \cdot \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) \cdot d\tau + T_d \cdot \frac{de(t)}{dt} \right)$$

Công thức tổng quát của bộ điều khiển PID dạng song song:

$$u_{PID}(t) = P + I + D = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) \cdot d\tau + K_d \cdot \frac{de(t)}{dt}$$

2.1.2 Lý thuyết về bộ điều khiển PID

➤ Khâu tỉ lệ (Khâu P)

Thành phần tỉ lệ làm thay đổi giá trị đầu ra, tỉ lệ với sai số hiện tại. Đáp ứng tỉ lệ có thể được điều chỉnh bằng cách nhân sai số đó với một hệ số K_p được gọi là độ lợi tỉ lệ.

$$P = K_p \cdot e(t)$$

Trong đó:

- P : Khâu tỉ lệ đầu ra.
- K_p : Thông số điều chỉnh.
- e: Giá trị sai số (sai số = SP - PV).
- t : Biến thời gian tức thời.

Đây là khâu quan trọng nhất trong bộ điều khiển PID. Tuy nhiên khâu này còn có 2 khuyết điểm là làm tín hiệu đáp ứng thay đổi nhanh và không thể triệt tiêu hết sai số xác lập (e_{xl}). Hai khuyết điểm này lần lượt sẽ được khắc phục bởi khâu D và I.

➤ Khâu tích phân (Khâu I)

Thành phần tích phân tỉ lệ thuận với cả biên độ sai số lẫn thời gian xảy ra sai số. Tổng sai số tức thời theo thời gian cho ta tích lũy bù đã được hiệu chỉnh trước đó. Tích lũy sai số sau đó được nhân với độ lợi tích phân và cộng với tín hiệu đầu ra của bộ điều khiển PID.

$$I = K_i \cdot \int_0^t e(\tau) \cdot d\tau$$

Trong đó:

- I : Tích phân đầu ra.
- Ki : Thông số điều chỉnh.
- e: Giá trị sai số (sai số=SP-PV).
- t : Biến thời gian tức thời.
- τ : Biến tích phân.

Khâu I giúp triệt tiêu sai số xác lập của hệ thống, nhưng có thể làm tăng tốc độ của tín hiệu trả về trong quá trình quá độ dẫn đến thời gian xác lập của hệ thống dài hơn.

➤ Khâu vi phân (Khâu D)

Thành phần vi phân cộng thêm tốc độ thay đổi sai số vào giá trị điều khiển ngõ ra. Nếu sai số thay đổi sẽ tạo thành phần cộng thêm vào giá trị điều khiển. Điều này cải thiện đáp ứng của hệ thống, giúp trạng thái của hệ thống thay đổi chậm lại và đạt được giá trị mong muốn. Tốc độ thay đổi của sai số quá trình được tính toán bằng cách xác định độ dốc của sai số theo thời gian (tức là đạo hàm bậc 1 của sai số theo thời gian) và nhân tốc độ này với độ lợi tỉ lệ Kd.

$$D = K_d \cdot \frac{de(t)}{dt}$$

Trong đó:

- D : Vi phân đầu ra.
- Kd : Thông số điều chỉnh.
- e : Giá trị sai số.
- t : Biến thời gian tức thời

Tỉ lệ vi phân làm chậm đáp ứng của hệ thống, giá trị càng lớn thì càng giảm độ vọt lố. Tuy nhiên nó rất nhạy cảm với nhiễu đo lường, nhiễu tần số cao.

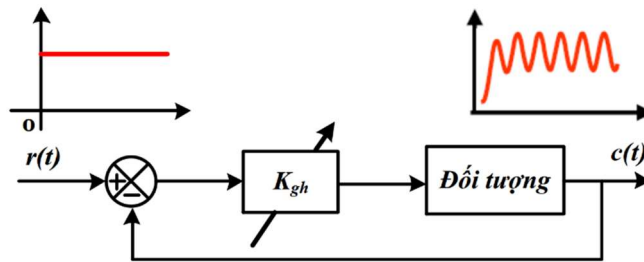
Tóm lại, thành phần tỉ lệ, tích phân, vi phân có ảnh hưởng đến tính toán đầu ra của bộ điều khiển PID. Trong đó các thông số điều chỉnh là:

- Độ lợi tỉ lệ Kp: giá trị càng lớn thì đáp ứng càng nhanh do đó sai số càng lớn. Một giá trị độ lợi tỉ lệ quá lớn sẽ dẫn đến quá trình mất ổn định và dao động.
- Độ lợi tích phân Ki: giá trị càng lớn kéo theo sai số xác lập bị khử càng nhanh. Đổi lại là độ vọt lố càng lớn bất kì sai số âm nào được tích phân

trong suốt đáp ứng quá độ phải được triệt tiêu tích phân bằng sai số dương trước khi tiến tới trạng thái xác lập.

- Độ lợi vi phân K_D : giá trị càng lớn càng giảm độ vọt lố, nhưng lại làm chậm đáp ứng quá độ và có thể dẫn đến mất ổn định do khuếch đại nhiễu tính hiệu trong phép vi phân sai số.

➤ Điều chỉnh bộ PID



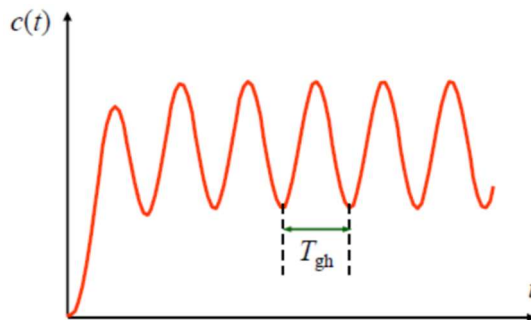
Hình 2. 2 Sơ đồ khối thiết kế bộ điều khiển PID dùng phương pháp Zeigler-Nichols 2

Là điều chỉnh các thông số điều khiển của PID tới đáp ứng điều khiển tối ưu tại một vùng điểm đặt biết trước. Độ ổn định là một yêu cầu cơ bản, nhưng ngoài ra các hệ số cũng ảnh hưởng qua lại với nhau để điều chỉnh tối ưu hóa hệ thống.

Có nhiều phương pháp khác nhau để điều chỉnh vòng lặp PID. Những phương pháp hữu hiệu nhất thường bao gồm những triển khai của vài dạng mô hình xử lý, sau đó chọn P, I và D dựa trên các thông số của mô hình động học. Các phương pháp điều chỉnh thủ công tương đối không hiệu quả lắm, đặc biệt nếu vòng lặp có thời gian đáp ứng được tính bằng phút hoặc lâu hơn.

Lựa chọn phương pháp thích hợp sẽ phụ thuộc phần lớn vào việc có hay không có vòng lặp, để có thể điều chỉnh “offline” từ đó xác định đáp ứng theo thời gian của hệ thống. Nếu hệ thống có thể thực hiện offline, phương pháp điều chỉnh tốt nhất, tín hiệu đo lường đầu ra là một hàm thời gian liên tục trên mô phỏng, sử dụng đáp ứng này để xác định các thông số điều khiển.

Ngoài phương pháp dò PID thủ công mà nhóm đã thực hiện thì nhóm còn có thể tham khảo phương pháp Ziegler-Nichols. Giống như phương pháp trên, độ lợi K_I và K_D lúc đầu được cho bằng 0. Độ lợi K_P được tăng đến khi tiến tới gần độ lợi giới hạn mà ở đó hệ thống bắt đầu xuất hiện dao động với biên độ không đổi. T_{gh} và thời gian dao động K_{gh} được gán độ lợi như sau:



Hình 2. 3 Đáp ứng của hệ kín ở biên giới ổn định

Bảng 2. 1 Phương pháp Ziegler-Nichols

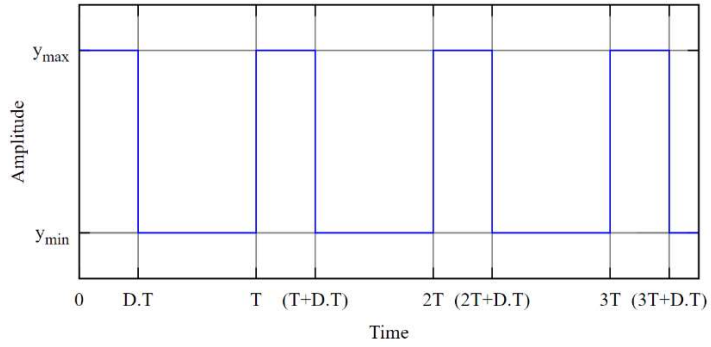
Thông số Bộ ĐK	K_p	T_i	T_D
P	$0,5 K_{gh}$	∞	0
PI	$0,45 K_{gh}$	$0,83 T_{gh}$	0
PID	$0,6 K_{gh}$	$0,5 T_{gh}$	$0,125 T_{gh}$

2.1.3 Phương pháp điều xung PWM

Phương pháp điều xung PWM là phương pháp điều chỉnh điện áp ra tải, hay nói cách khác, là phương pháp điều chế dựa trên sự thay đổi độ rộng của chuỗi xung vuông, dẫn đến sự thay đổi điện áp ra.

Các PWM khi biến đổi thì có cùng 1 tần số và khác nhau về độ rộng của sườn dương hay sườn âm.

➤ Nguyên lý:



Hình 2. 4 Xung PWM

Điều chế độ rộng xung sử dụng một sóng xung hình chữ nhật có độ rộng được điều chế dẫn đến sự biến thiên của giá trị trung bình của dạng sóng. Nếu chúng ta xem xét một sóng xung $f(t)$, với chu kỳ T , giá trị thấp y_{\min} , giá trị cao y_{\max} và chu kỳ làm việc D , giá trị trung bình của dạng sóng đó được cho bởi:

$$\bar{y} = \frac{1}{T} \int_0^T f(t) dt$$

Vì $f(t)$ là một sóng xung, giá trị của nó là y_{\max} trong khoảng $0 < t < DT$ và y_{\min} trong khoảng $DT < t < T$. Biểu thức trở thành:

$$\begin{aligned} \bar{y} &= \frac{1}{T} \int_0^T f(t) dt \\ \Leftrightarrow \bar{y} &= \frac{1}{T} \left(\int_0^{DT} y_{\max} dt + \int_{DT}^T y_{\min} dt \right) \\ \Leftrightarrow \bar{y} &= \frac{1}{T} (DT \cdot y_{\max} + T(1 - D) \cdot y_{\min}) \\ \Leftrightarrow \bar{y} &= D \cdot y_{\max} + (1 - D) \cdot y_{\min} \end{aligned}$$

Biểu thức sau này có thể khá đơn giản trong nhiều trường hợp trong đó $y_{\min} = 0$ khi $\bar{y} = D \cdot y_{\max}$. Từ đó, rõ ràng là giá trị trung bình của tín hiệu (\bar{y}) trực tiếp phụ thuộc vào chu kỳ làm việc D .

$$\text{Dutycycle} = \frac{T_{\text{on}}}{T_{\text{PWM}}} \cdot 100 = \frac{\text{Pulse Width}}{\text{Period}} \cdot 100 \quad [\%]$$

Với:

- Duty Cycle là tỷ lệ phần trăm mức cao [%].
- Period là chu kỳ xung [Hz].
- Pulse Width là thời gian mức cao trong một chu kỳ [s].

$$F_{PWM} = \frac{F_{TimerClock}}{(Prescaler + 1)(CounterPeriod + 1)}$$

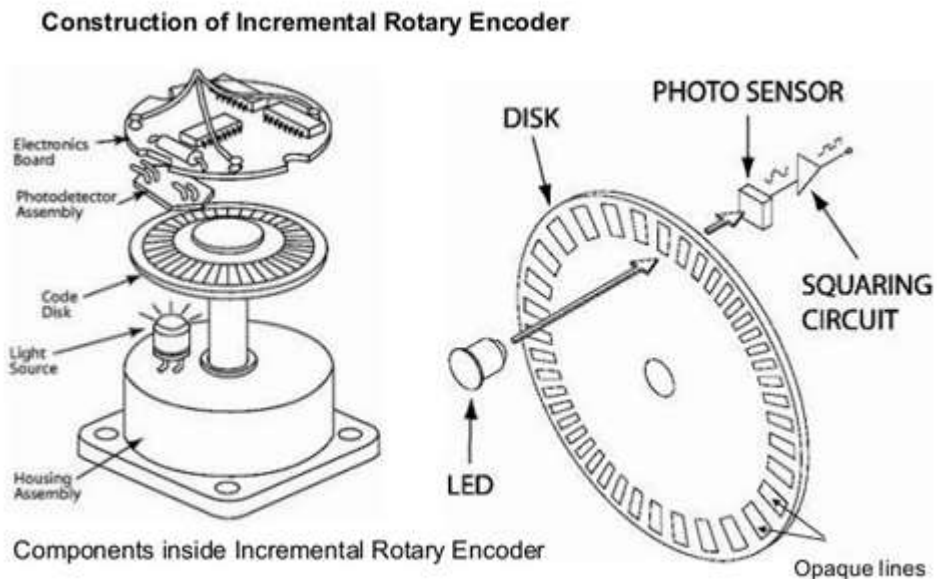
Với:

- F_{PWM} là tần số xuất xung mà ta mong muốn [Hz].
- $F_{TimerClock}$ là tần số của bộ timer ta đã cài đặt cho chip [Hz].
- Prescaler là bộ chia timer.
- CounterPeriod là giá trị đếm tối đa.

2.1.4 Phương pháp đọc xung encoder

Bộ mã hóa tín hiệu vòng quay tương đối (hay còn được gọi tắt là bộ mã hóa vòng quay, bộ mã hóa tương đối) là một thiết bị cơ điện có khả năng làm biến đổi chuyển động tuyến tính(chuyển động thẳng) hoặc chuyển động tròn thành tín hiệu số hoặc xung. Khi có sự thay đổi vị trí trục quay, các bộ mã hoá được sử dụng để kiểm tra góc lệch của trục đang làm việc

Hình 2.5 cho thấy các thành phần cơ bản của bộ mã hóa vòng quay, bao gồm đi-ốt phát quang (LED), đĩa và bộ dò ánh sáng ở mặt đối diện của đĩa. Đĩa, được gắn trên trục quay, có các mô hình khu vực mờ đục và trong suốt được mã hóa vào đĩa. Khi đĩa quay, các phân đoạn mờ đục sẽ chặn ánh sáng và ở những nơi thủy tinh trong suốt, ánh sáng được phép đi qua. Điều này tạo ra các xung sóng vuông, sau đó có thể được giải thích thành vị trí hoặc chuyển động.



Hình 2. 5 Các thành phần của bộ mã hóa quang

Bộ mã hóa thường có từ 100 đến 6.000 phân đoạn trên mỗi vòng quay. Điều này có nghĩa là các bộ mã hóa này có thể cung cấp độ phân giải 3,6 độ cho bộ mã hóa có 100 phân đoạn và độ phân giải 0,06 độ cho bộ mã hóa có 6.000 phân đoạn.

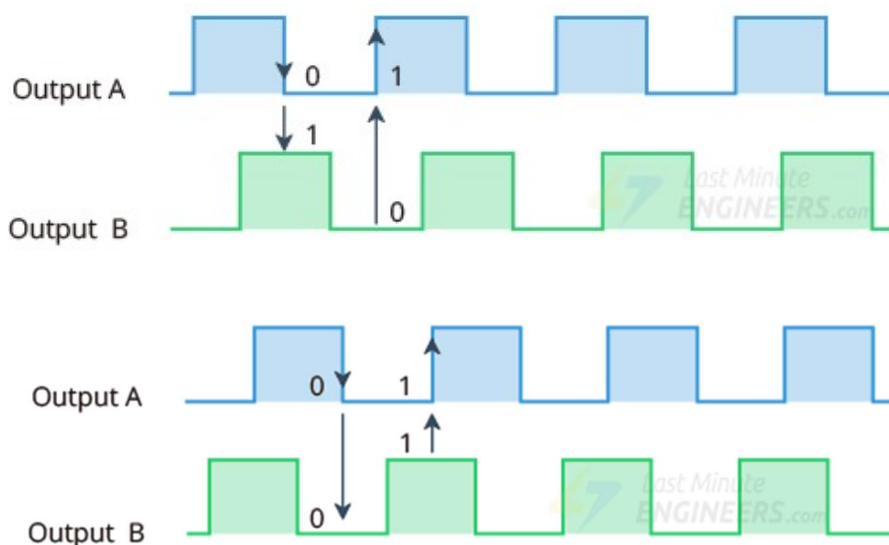
Bộ mã hóa tuyến tính hoạt động theo nguyên tắc giống như bộ mã hóa quay ngoại trừ thay vì đĩa quay, có một dải mờ cố định với các khe trong suốt dọc theo bề mặt và cụm đầu dò LED được gắn vào thân chuyển động.

Sử dụng hai rãnh mã với các cung được định vị lệch pha 90 độ (Hình 2.6), hai kênh đầu ra của bộ mã hóa cầu phương cho biết cả vị trí và hướng quay. Ví dụ, tín hiệu A xuất hiện trước tín hiệu B, thì đĩa đang quay theo chiều kim đồng hồ. Còn nếu như dãy mã hóa B đi trước A thì đĩa quay ngược chiều kim đồng hồ.

Nói cách khác khi A thay đổi trạng thái:

- Nếu B khác A thì đĩa quay thuận
- Nếu B giống A thì đĩa quay nghịch

Do đó, bằng cách theo dõi cả số lượng xung và pha tương đối của tín hiệu A và B, bạn có thể theo dõi cả vị trí và hướng quay.

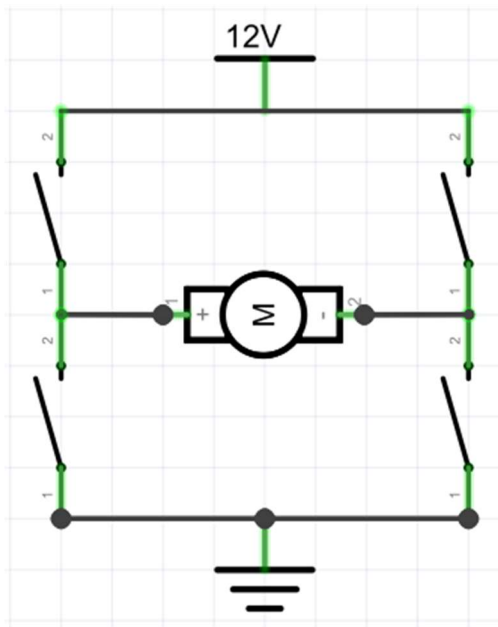


Hình 2. 6 Tín hiệu đầu ra A và B của bộ mã hóa cầu phương

Ngoài ra, một số bộ mã hóa bậc hai bao gồm kênh đầu ra thứ ba – được gọi là tín hiệu không hoặc tín hiệu tham chiếu – cung cấp một xung trên mỗi vòng quay. Bạn có thể sử dụng xung đơn này để xác định chính xác vị trí tham chiếu. Trong phần lớn các bộ mã hóa, tín hiệu này được gọi là Z-Terminal hoặc chỉ mục.

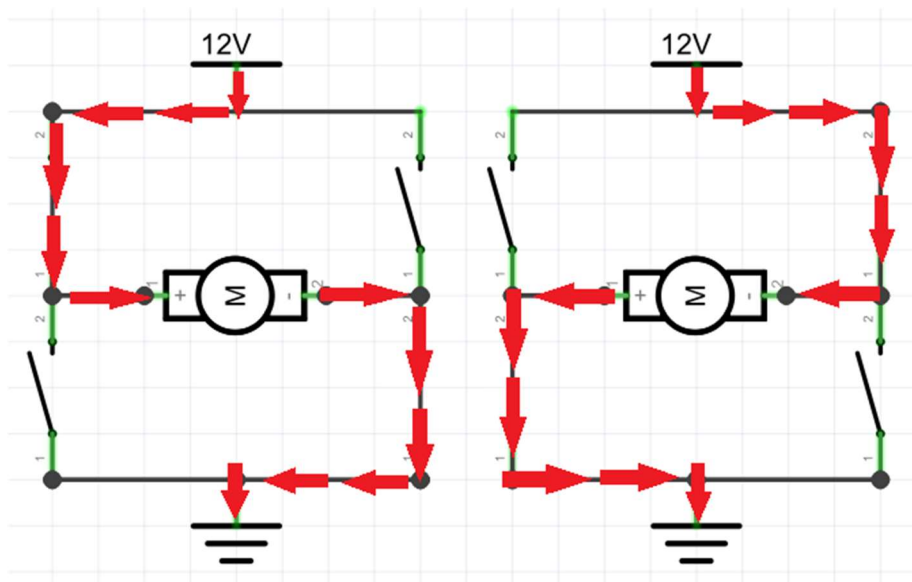
2.1.5 Nguyên lý hoạt động mạch cầu H

Xét một cách tổng quát, mạch cầu H là một mạch gồm 4 "công tắc" được mắc theo hình chữ H.



Hình 2. 7 Cấu tạo cầu H

Bằng cách điều khiển 4 "công tắc" này đóng mở, ta có thể điều khiển được dòng điện qua động cơ cũng như các thiết bị điện tương tự.



Hình 2. 8 Nguyên lý hoạt động cầu H

2.2 Các công cụ và thiết bị

2.2.1 Giới thiệu KIT STM32F4 Discovery



Hình 2. 9 KIT STM32F407VG Discovery

STM32F4 Discovery là board vi điều khiển dành cho người mới học lập trình nhúng hoặc dành cho những người muốn làm quen với lập trình trên vi điều khiển 32-bit dòng ARM... Board được tích hợp chip ARM Cortex-M4 cùng với bộ tính toán số thực (FPU), hoạt động với tần số rất cao 168 MHz, tỷ suất DMIPS/MHZ cao 1.25 giúp cho hệ thống có thể đạt được hiệu năng 210 DMIPS, board rất thích hợp cho các ứng dụng với yêu cầu tính toán xử lý nhanh, ví dụ như DSP, điều khiển robot...

Các đặc điểm của KIT STM32F4Discovery:

- Sử dụng vi điều khiển lõi ARM Cortex-M4 32-bit STM32F407VGT6, 1 MB Flash, 192 KB RAM đóng gói LQFP100 (100 chân)
- Tích hợp sẵn mạch nạp ST-LINK/V2
- Nguồn cung cấp cho board: qua USB bus hoặc từ nguồn điện ngoài 5V
- Cấp nguồn cho ứng dụng ngoài: 3V và 5V
- Cảm biến chuyển động LIS302DL, ST MEMS 3 trục gia tốc.
- Cảm biến âm thanh MP45DT02 ST-MEMS, mic cảm biến âm thanh vô hướng kỹ thuật số.
- Bộ chuyển đổi DAC âm thanh CS43L22.
- Tám đèn LED:
LD1 (đỏ / xanh lá cây) để giao tiếp USB
LD2 (màu đỏ) báo hiệu nguồn 3,3 V on

Bốn đèn LED màu: LD3 (màu cam), LD4 (màu xanh lá cây), LD5 (màu đỏ) và LD6 (màu xanh dương).

Hai USB OTG LED LD7 (màu xanh lá cây) VBUS và LD8 (màu đỏ).

- Hai nút bấm (nút bấm User màu xanh, nút bấm Reset màu đen).
- OTG FS USB với cổng nối micro-AB.
- Header mở rộng cho tất cả LQFP100 I/O.
- Phần mềm miễn phí bao gồm một loạt các ví dụ, sử dụng thư viện chuẩn của ST

Các công cụ và thiết bị được sử dụng trong đề tài: Motor, Nguồn, Các linh kiện điện tử cơ bản, Các IC đo lường, Các IC khó tìm bằng tiếng việt. Các IC quan trọng trong đề tài... Giới thiệu các phần mềm hay chương trình sử dụng trong đề tài như: Labview, C++, Tia Portal,... Những phần mềm nào không được học trong trường thì giới thiệu kỹ. Phần mềm nào được học thì giới thiệu ít.

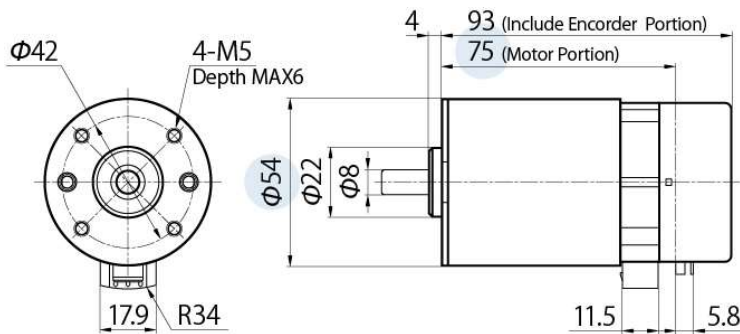
2.2.2 Động cơ DC Servo NF5475E với Encoder



Hình 2. 10 Hình dáng bên ngoài động cơ NF5475E

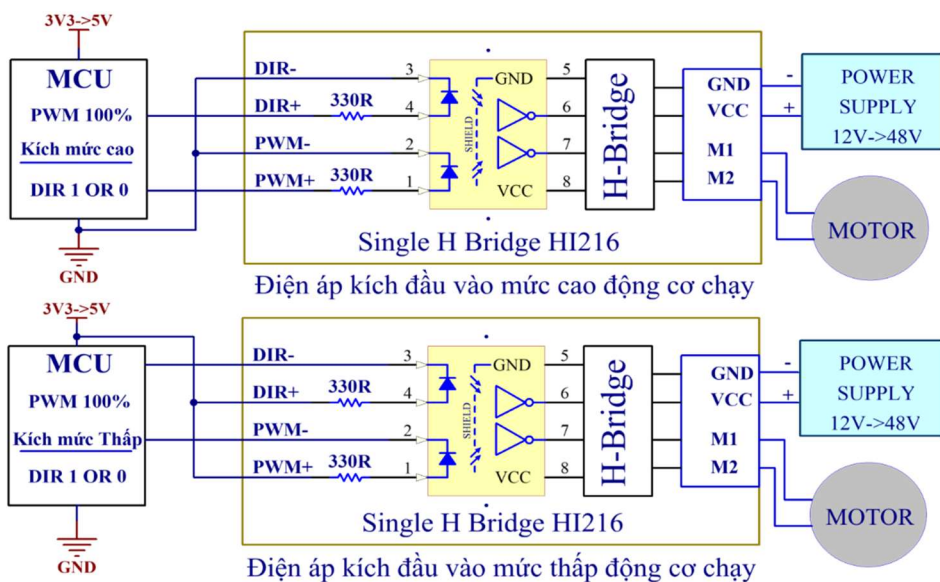
Thông số kỹ thuật:

- Điện áp hoạt động : 12VDC – 24VDC
- Cân nặng: 700g
- Số cực từ: 4 cực
- Mô men xoắn : 74.83
- Tốc độ không tải: 4884 rpm
- Cường độ dòng điện: 0.218(không tải), 7.246(khởi động)
- Công suất tối đa: 38.9W
- Độ phân giải encoder: 200 P/R
- Điện áp hoạt động encoder: 5VDC
- Tần số phản hồi tối đa: 20kHz
- Loại encoder: quang



Hình 2. 11 Kích thước động cơ NF5475E

2.2.3 Mạch cầu H-HI216

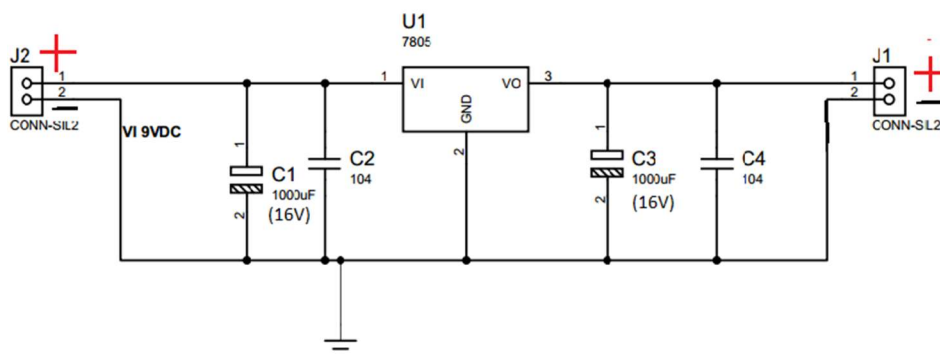


Hình 2. 12 Cách đấu nối mạch cầu H-Hi216

Thông số kỹ Thuật:

- Dòng liên tục 15A, dòng đỉnh 20A công suất 600W, tại 25oC.
- Điện áp công suất từ 12V đến 48V.
- Điện áp kích từ 3V3 -> 5V.
- Có cầu chì bảo vệ ngắn mạch.
- Có Led báo nguồn (POW), tín hiệu xung đưa vào (PWM) và tín hiệu đảo chiều (DIR).
- Board được thiết kế nhỏ gọn với kích thước 52x64x22mm.
- Tần số hoạt động lên tới 100 Khz, sử dụng Opto HCPL-0631 cho tần số hoạt động cao.
- Độ rộng xung 100%.

2.2.4 Mạch nguồn ổn áp sử dụng IC L7805

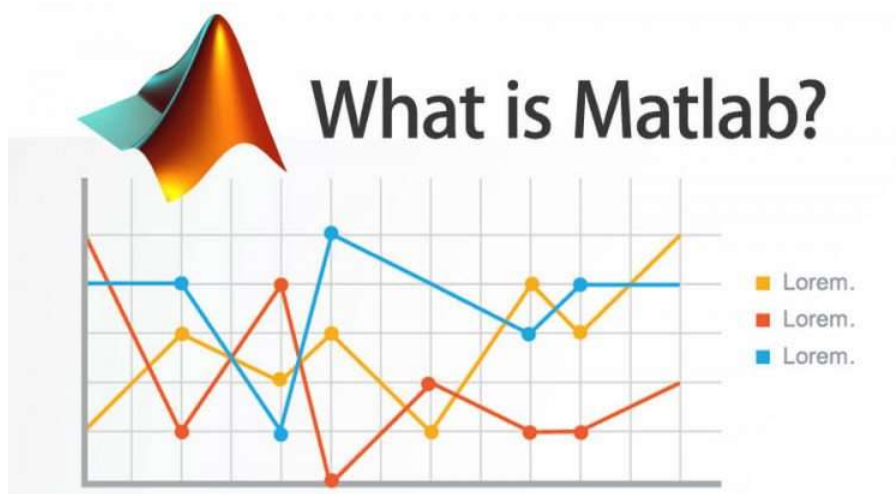


Chúng ta sẽ cấp điện áp đầu vào qua J2 (tương ứng theo các chân âm dương) và điện áp 5V ở ngõ ra sẽ được lấy qua chân J1.

Tụ C1 và C2 để lọc điện áp đầu vào cấp cho chân Vi của IC 7805, tụ C1 có các dụng cung cấp điện áp tạm thời cho chân Vi khi nguồn đột ngột bị sụt áp, tụ C2 là tụ gốm nên trở kháng lớn, C2 có tác dụng ngăn nguồn đầu vào tăng áp đột ngột làm dạng sóng điện áp đầu vào có hình răng cưa.

Tụ C3 và C4 để lọc điện áp cấp cho tải tiêu thụ lấy từ chân Vo của IC 7805, tụ C3 có các dụng cung cấp điện áp tạm thời cho tải khi điện áp tải đột ngột bị sụt áp, tụ C4 trở kháng lớn, C4 có tác dụng lọc nhiễu điện áp đầu ra (nhiều là các điện áp không mong muốn làm cho dạng sóng điện áp ngõ ra có hình răng cưa).

2.2.5 Phần mềm MATLAB



Hình 2. 13 Phần mềm MATLAB

Matlab là tên viết tắt của Matrix laboratory phần mềm được MathWorks thiết kế để cung cấp môi trường lập trình và tính toán kỹ thuật số.

Matlab cho phép bạn sử dụng ma trận để tính toán các con số, vẽ thông tin cho các hàm và đồ thị, chạy các thuật toán, tạo giao diện người dùng và liên kết với các chương trình máy tính được viết bằng nhiều ngôn ngữ lập trình khác.

Matlab được sử dụng để giải quyết các vấn đề trong phân tích số, xử lý tín hiệu kỹ thuật số và xử lý đồ họa mà không cần lập trình cổ điển.

Matlab hiện có hàng nghìn lệnh và chức năng tiện ích. Ngoài các chức năng có sẵn của chính ngôn ngữ, Matlab còn có các lệnh ứng dụng đặc biệt và các chức năng hộp công cụ (Toolbox) để mở rộng môi trường Matlab nhằm giải quyết một số loại vấn đề nhất định.

Hộp công cụ rất quan trọng và hữu ích cho người sử dụng toán học sơ cấp, xử lý tín hiệu kỹ thuật số, xử lý hình ảnh, xử lý giọng nói, ma trận thưa, logic mờ...

Là một ngôn ngữ lập trình bậc cao (Script) với các lệnh điều khiển, chức năng, cấu trúc dữ liệu, đầu vào/ đầu ra và khả năng lập trình hướng đối tượng. Cho phép bạn nhanh chóng tạo và phá hủy phần mềm trong “lập trình quy mô nhỏ” hoặc tạo các chương trình lớn và phức tạp trong “lập trình quy mô lớn”.

Tính ứng dụng của Matlab khá rộng rãi, được sử dụng như công cụ tính toán trong lĩnh vực khoa học và kỹ thuật: công nghệ, toán học, hóa học, vật lý...

Matlab được sử dụng hầu hết trong các việc như:

- Xử lý tín hiệu và truyền thông.
- Xử lý chất lượng hình ảnh, video.
- Ứng dụng tính toán tài chính, sinh học.
- Ứng dụng trong kiểm tra, tính toán và đo lường.
- Hệ thống điều khiển

2.2.6 Phần mềm LABVIEW



Hình 2. 14 Phần mềm LABVIEW

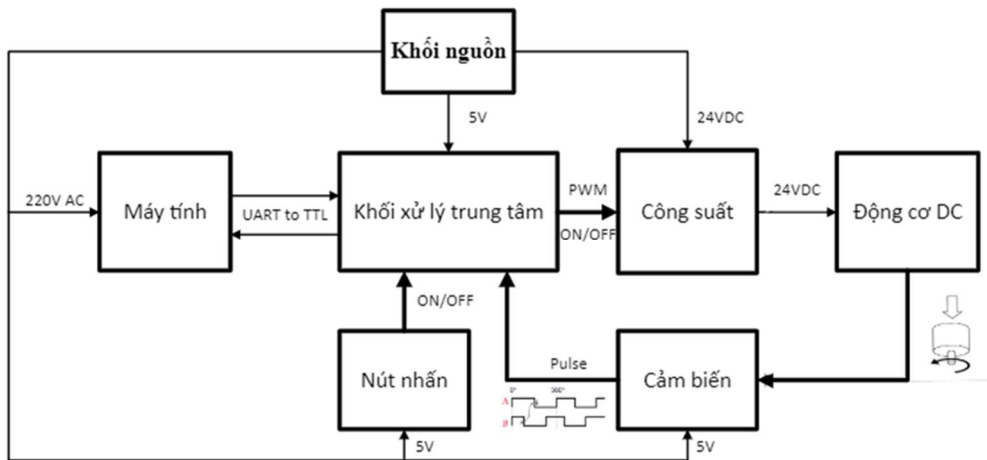
Phần mềm LabVIEW được dùng nhiều trong các phòng thí nghiệm, lĩnh vực khoa học kỹ thuật như tự động hóa, điều khiển, điện tử, cơ điện tử, hàng không, hóa sinh, điện tử y sinh... Hiện tại ngoài phiên bản LabVIEW cho các hệ điều hành Windows, Linux, hãng NI đã phát triển các mô đun LabVIEW cho máy hỗ trợ cá nhân (PDA).

Các chức năng chính của LabVIEW có thể tóm tắt như sau:

- Thu thập tín hiệu từ các thiết bị bên ngoài như cảm biến nhiệt độ, hình ảnh từ webcam, vận tốc của động cơ...
- Mô phỏng và xử lý các tín hiệu thu nhận được để phục vụ các mục đích nghiên cứu hay mục đích của hệ thống mà người lập trình mong muốn.
- Xây dựng các giao diện người dùng một cách nhanh chóng và thẩm mỹ hơn nhiều so với các ngôn ngữ khác như Visual Basic, MATLAB..
- Giao tiếp với các thiết bị ngoại vi thông qua các chuyển truyền thông giao tiếp như: RS232, RS485, Ethernet...
- Cho phép thực hiện các thuật toán điều khiển như PID, Logic mờ (Fuzzy Logic), một cách nhanh chóng thông qua các chức năng tích hợp sẵn trong LabVIEW.
- Cho phép kết hợp với nhiều ngôn ngữ lập trình truyền thống như C, C++...

CHƯƠNG 3 THIẾT KẾ SƠ ĐỒ

3.1 Sơ đồ khối tổng quát của hệ thống



Hình 3. 1 Sơ đồ khối tổng quát

Trong khối nguồn công suất biến đổi điện áp từ 220VAC thành 24VDC cung cấp nguồn cho mạch cầu H cấp nguồn cho động cơ khi nhận được tín hiệu điều khiển từ bộ xử lý trung tâm

Khi động cơ quay, 1 encoder được gắn đồng trục sẽ phát ra chuỗi xung vuông 0-5VDC. Tần số xung tỉ lệ với tốc độ quay trục động cơ.

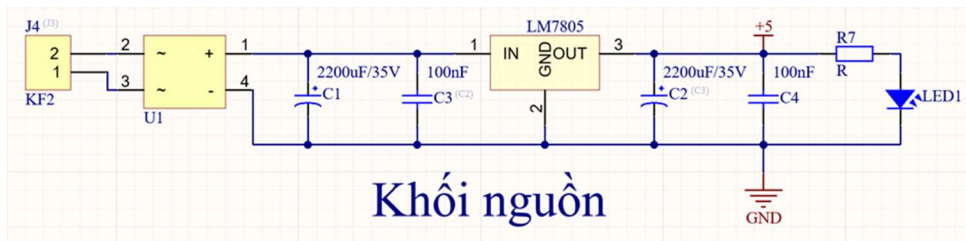
Khối xử lý trung tâm nhận tín hiệu từ khối encoder trong khoảng thời gian 50ms bằng ngắt timer. Sau đó tính toán tốc độ và thông qua thuật toán PID tính toán giá trị điều độ rộng xung PWM điều khiển driver làm cho động cơ hoạt động với tốc độ mong muốn.

Khối nút nhấn sẽ xuất tín hiệu về khối xử lý trung tâm mỗi lần nhấn để tăng hoặc giảm giá trị tốc độ mong muốn.

Máy tính giao tiếp với khối xử lý trung tâm và hiển thị các giá trị hệ thống lên giao diện người dùng thông qua chuẩn giao tiếp RS232

3.2 Sơ đồ nguyên lý

3.2.1 Sơ đồ nguồn



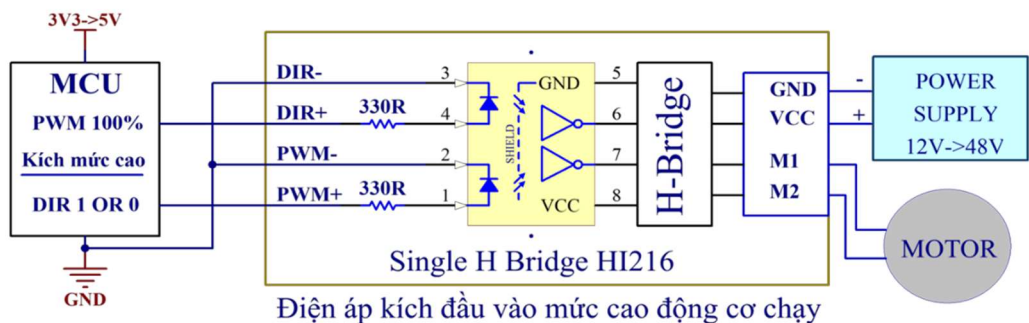
Nguồn khối nguồn, điện áp 220VAC được biến đổi thành 15VAC bằng biến áp sau đó thông qua cầu diode được chuyển thành 15VDC.

Tụ C1 và C2 để lọc điện áp đầu vào cấp cho chân Vi của IC 7805, tụ C1 có các dụng cung cấp điện áp tạm thời cho chân Vi khi nguồn đột ngột bị sụt áp, tụ C2 là tụ gốm nên trở kháng lớn, C2 có tác dụng ngăn nguồn đầu vào tăng áp đột ngột làm dạng sóng điện áp đầu vào có hình răng cưa.

Tụ C3 và C4 để lọc điện áp cấp cho tải tiêu thụ lấy từ chân Vo của IC 7805, tụ C3 có các dụng cung cấp điện áp tạm thời cho tải khi điện áp tải đột ngột bị sụt áp, tụ C4 trở kháng lớn, C4 có tác dụng lọc nhiễu điện áp đầu ra (nhiều là các điện áp không mong muốn làm cho dạng sóng điện áp ngõ ra có hình răng cưa).

3.2.2 Sơ đồ khối công suất

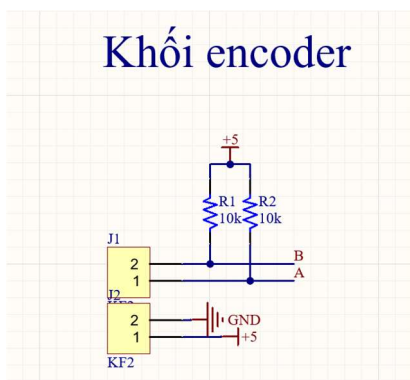
Mạch cầu H



Mạch cầu Hi216 có cùng nguyên lý hoạt động với mạch cầu H.

Sử dụng Opto HCPL-0631 cho tần số hoạt động cao và IC kích Fet chuyên dụng cho Fet hoạt động tốt nhất tránh hiện tượng trùng dẫn giúp đạt được hiệu năng cao.

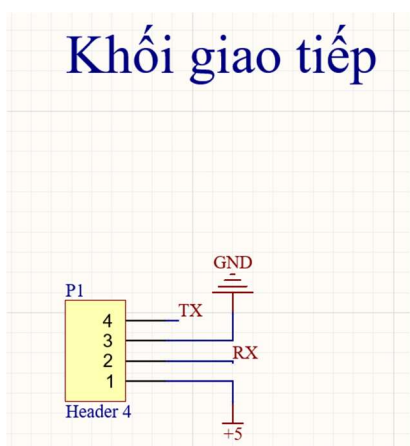
3.2.3 Sơ đồ khối encoder



Vì cấu tạo của encoder là mạch Open Collector nên cần phải có trở treo để VDK có thể xử lý dữ liệu chính xác.

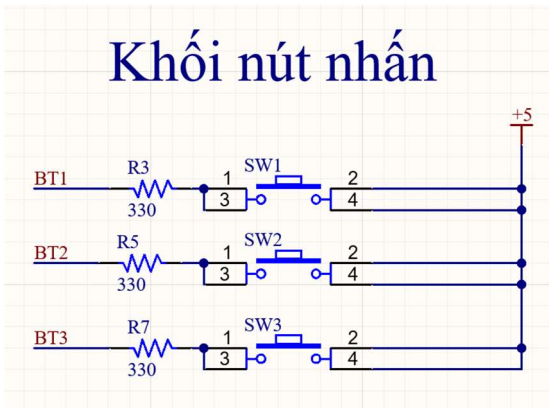
Và 2 kênh A,B của encoder được đọc bằng 2 chân E9,E11 của VDK

3.2.4 Sơ đồ khối UART



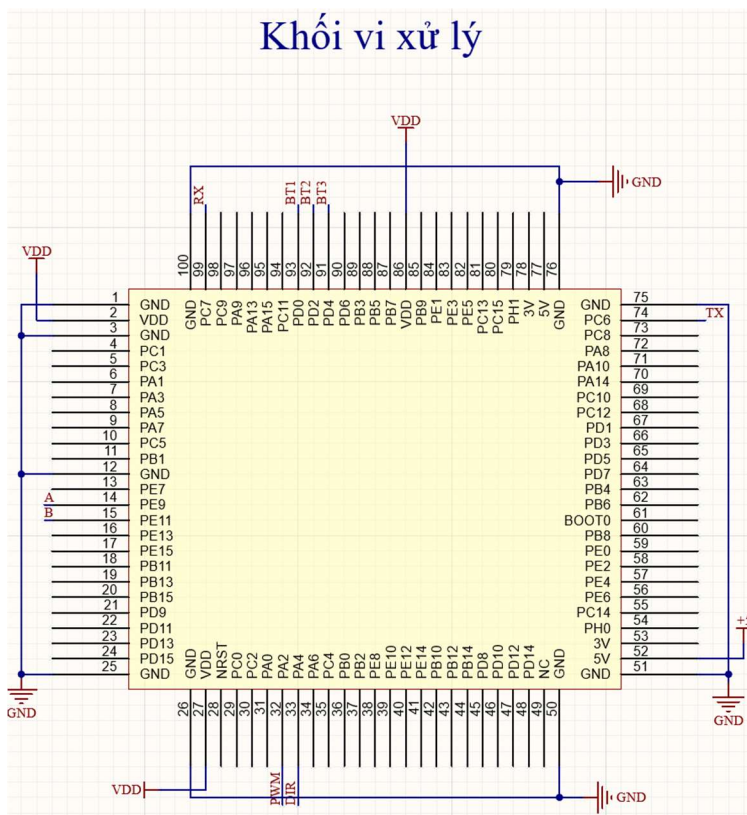
Vì điều khiển sử dụng chức năng UART tại 2 chân C6/C7 thông qua mạch Uart to TTL để giao tiếp với COM máy tính.

3.2.5 Sơ đồ khối nút nhấn



Vi điều khiển nhận tín hiệu nút nhấn từ các chân PD0/PD2/PD4 được cài đặt Pull-down. Các trở R3,R5,R7 có tác dụng hạn dòng trước khi đi vào chân IO của VDK.

3.2.6 Sơ đồ khối xử lý trung tâm

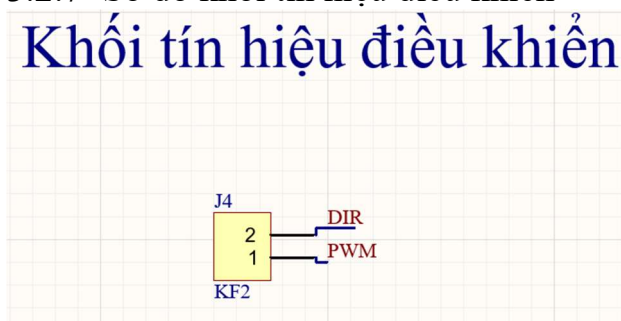


Vi điều khiển Stm32f407 được sử dụng làm bộ xử lý trung tâm nhận và xuất tín hiệu điều khiển cho các khối ngoại vi.

Được cấp nguồn bằng mạch ổn áp 220VAC-5VDC. Trình bày chức năng và cách giao tiếp giữa khối đang giới thiệu và các khối xung quanh.

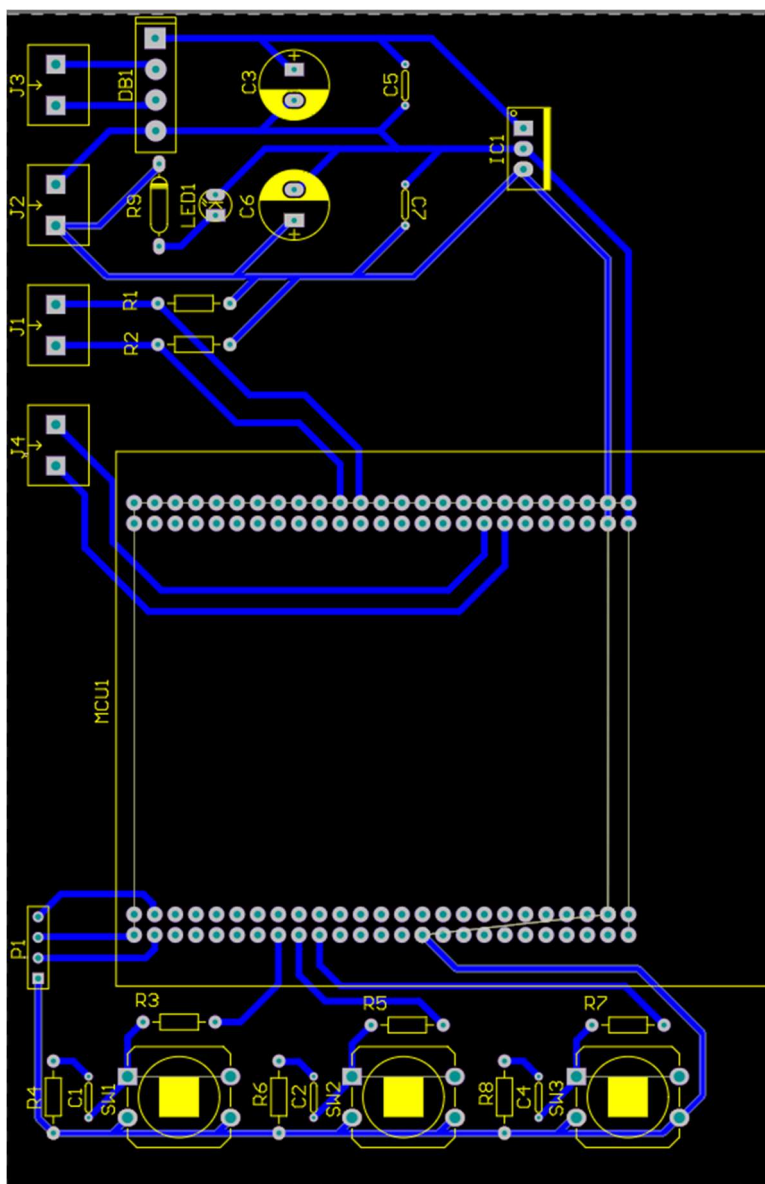
3.2.7 Sơ đồ khối tín hiệu điều khiển

Khối tín hiệu điều khiển



Tín hiệu điều khiển bao gồm tín hiệu PWM được xuất từ chân PA2 và tín hiệu chiều quay động cơ DIR xuất từ chân PA4 của VDK.

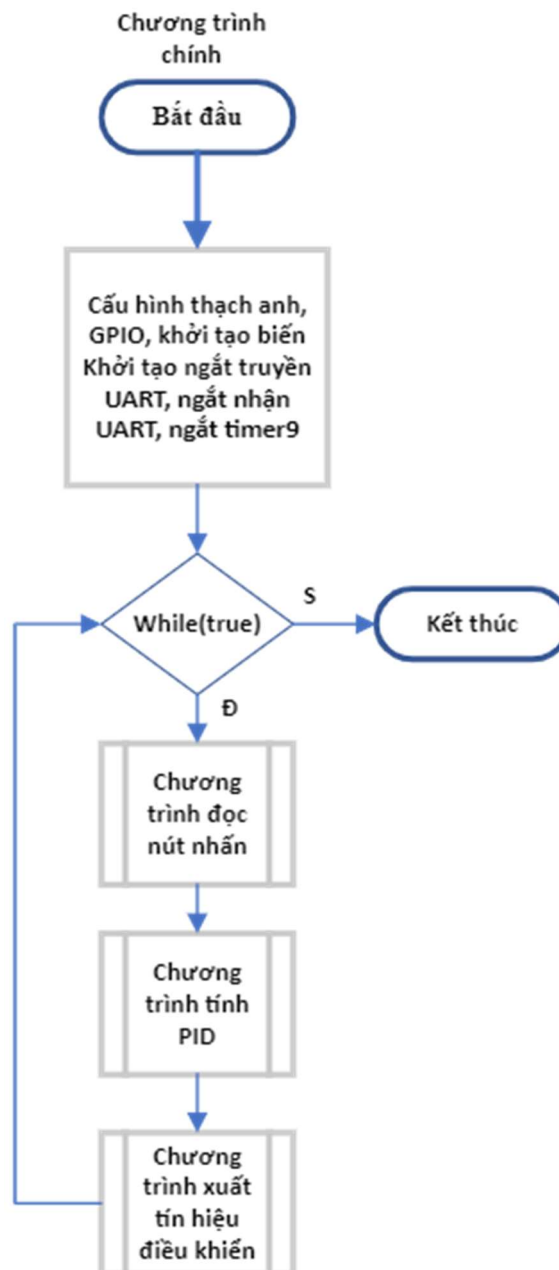
3.3 Sơ đồ thiết kế PCB



Hình 3. 3 Hình ảnh thiết kế đi dây PCB

CHƯƠNG 4 CHƯƠNG TRÌNH ĐIỀU KHIỂN

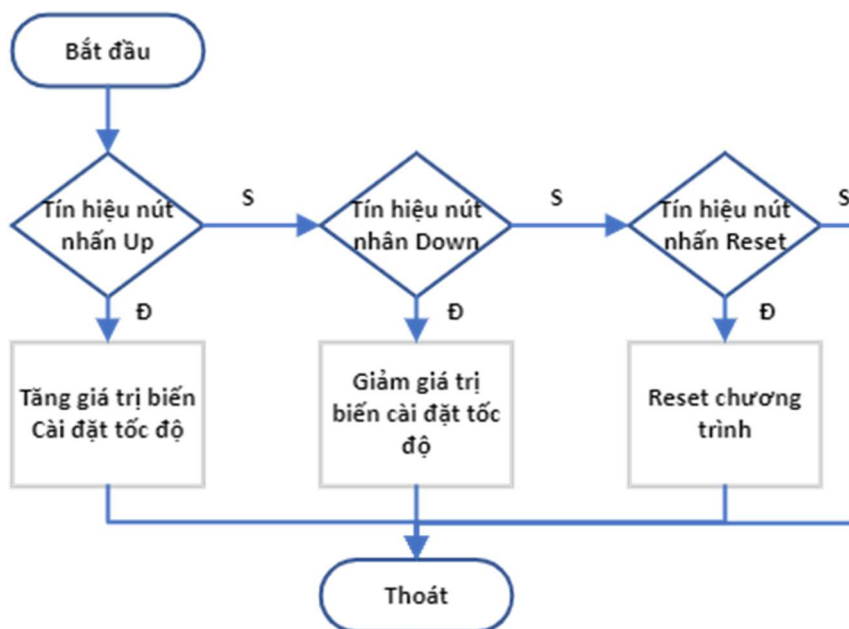
4.1 Lưu đồ giải thuật của chương trình chính



Chương trình chính: Khi bắt đầu chạy chương trình chính sẽ khai báo cấu hình các chân GPIO, các ngắt truyền, nhận UART và ngắt timer.

4.2 Lưu đồ giải thuật của chương trình nút nhấn

Chương trình nút
nhấn



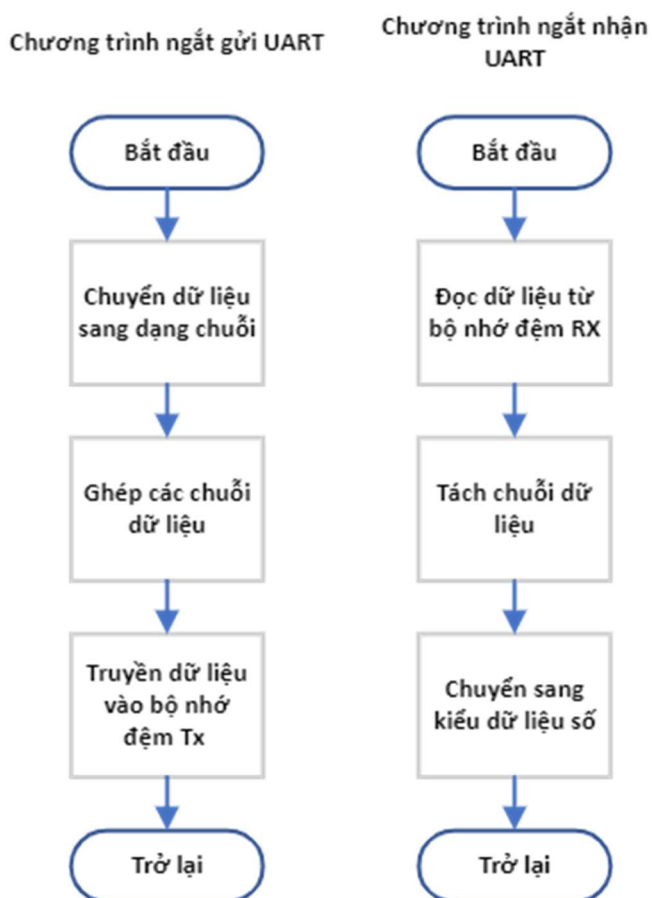
Chương trình nút nhấn: khi nhận được tín hiệu nhấn nút vi điều khiển sẽ căn cứ vào chân nhận được tín hiệu mà xử lý:

Nếu tín hiệu nhận được là nút Up: biến cài đặt tốc độ được tăng 100rpm và giới hạn tối đa 800rpm.

Nếu tín hiệu nhận được là nút Down: biến cài đặt tốc độ được giảm 100rpm và giới hạn tối thiểu là -800rpm.

Nếu tín hiệu nhận được là nút Reset: Giá trị của các biến trong chương trình sẽ được đặt về mặc định.

4.3 Lưu đồ giải thuật của chương trình UART

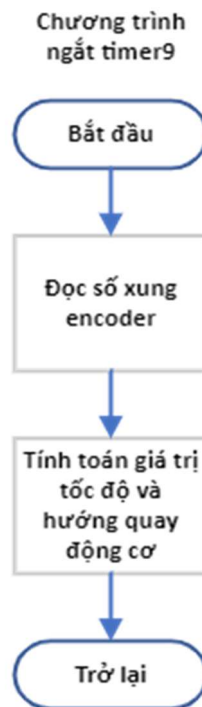


Chương trình UART được thực hiện liên tục việc truyền nhận dữ liệu giữa VDK và máy tính.

Giá trị nhận về bao gồm: giá trị tốc độ chạy, thông số bộ PID (K_p, K_i, K_d).

Giá trị gửi bao gồm: giá trị cài đặt tốc độ (chương trình nút nhấn), giá trị tốc độ (chương trình đọc encoder), giá trị sai số và giá trị tín hiệu điều khiển (chương trình tính PID).

4.4 Lưu đồ giải thuật của chương trình đọc encoder



Chương trình đọc encoder: được gọi mỗi khi có tín hiệu từ chương trình ngắt với chu kỳ là thời gian lấy mẫu(50ms).

Với số xung encoder đọc được trong mỗi chu kỳ ngắt giá trị tốc độ được tính theo công thức:

$$\text{Delta xung} / 800 \text{ xung} * 60 / \text{thời gian lấy mẫu} = \text{tốc độ động cơ}$$

4.5 Lưu đồ giải thuật của chương trình tính PID



Chương trình tính PID: Giá trị điều khiển được tính bằng PID dựa vào giá trị tốc độ chạy, thông số bộ PID (Chương trình UART) và giá trị tốc độ động cơ (Chương trình đọc encoder).

4.6 Lưu đồ giải thuật của chương trình đọc xuất tín hiệu điều khiển

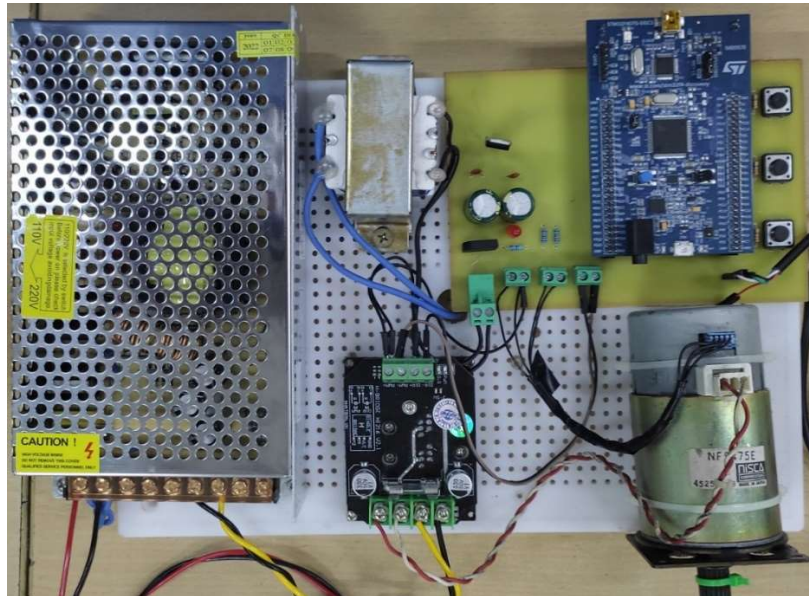


Chương trình xuất tín hiệu điều khiển: Sau khi tín hiệu điều khiển được tính toán ở chương trình tính PID chương trình xuất tín hiệu sẽ quyết định chiều chạy của động cơ bằng chân DIR và tốc độ động cơ bằng PWM.

CHƯƠNG 5 KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN

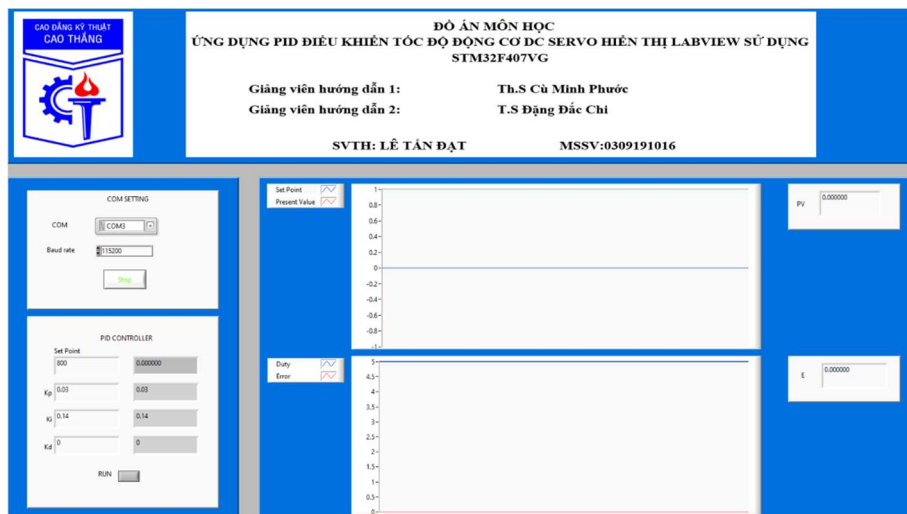
5.1 Kết quả đạt được

5.1.1 Kết quả thiết kế thi công phần cứng



Hình 5. 1 Hình ảnh mô hình thực tế

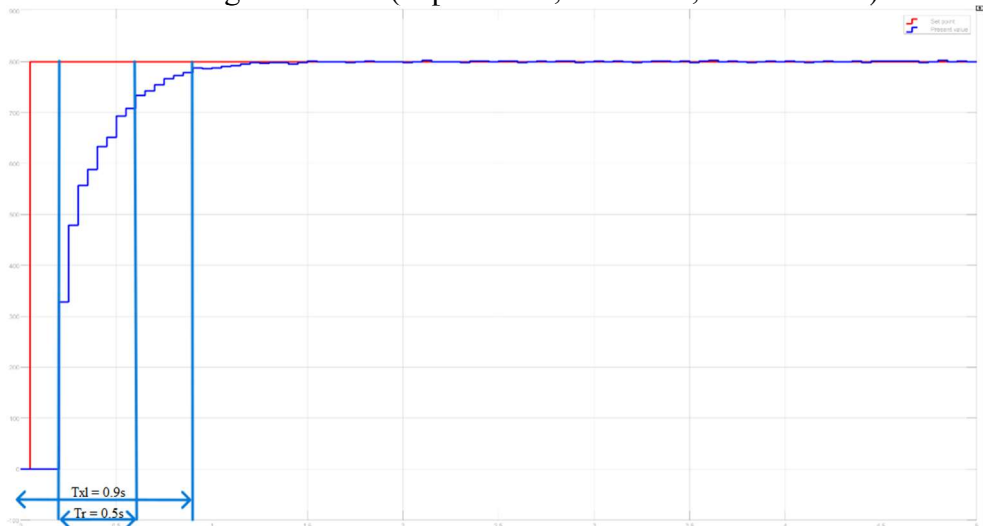
5.1.2 Kết quả thiết kế thi công phần mềm



Hình 5. 2 Hình ảnh giao diện người dùng thực tế

5.1.3 Kết quả vận hành

Với thông số cài PID ($K_p = 0.023$, $K_i = 0.14$, $K_d = 0.0001$)



Hình 5. 3 Đáp ứng hệ thống ở tốc độ cài 800rpm

Nhận xét :

- Độ vọt lố 0%
- Sai số xác lập = 0
- Thời gian lên T_r (từ 10% - 90%): 0.5s
- Thời gian xác lập (5%): 0.9s

5.2 Nhận xét về kết quả

- Hoàn thành đề tài đúng thời hạn đề ra
- Hệ thống hoạt động đáp ứng được các yêu cầu của đề tài

Kết quả đạt được trong quá trình làm:

- Tìm hiểu được phương pháp điều xung
- Phương pháp điều khiển dùng giải thuật PID
- Thiết kế và vẽ mạch điện tử PCB

Điểm còn hạn chế: Hệ thống hoạt động chưa thực sự tốt ở tốc độ thấp do cấu tạo cơ khí chưa tốt

5.3 Hướng phát triển

- Sử dụng encoder có độ phân giải cao hơn, thời gian lấy mẫu cao hơn để có được phản hồi chính xác nhất
- Sử dụng động cơ tốt hơn để hệ thống điều khiển sát với hệ tuyến tính cho ra kết quả điều khiển tốt hơn
- Áp dụng phương pháp điều khiển vị trí cho động cơ

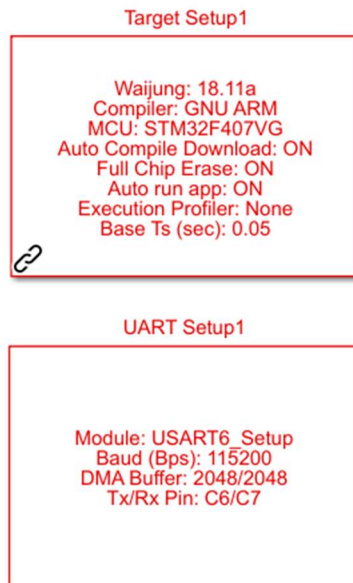
Tài Liệu Tham Khảo

<https://www.instructables.com/Speed-Control-of-DC-Motor-Using-PID-Algorithm-STM3/>

<https://starfujii.vn/hi216-mach-cau-h-600w-15a>

<https://lastminuteengineers.com/rotary-encoder-arduino-tutorial/>

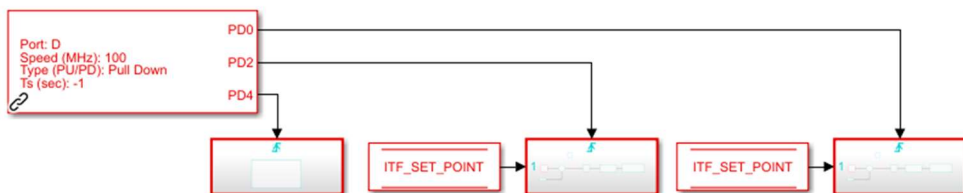
Chương trình tham khảo



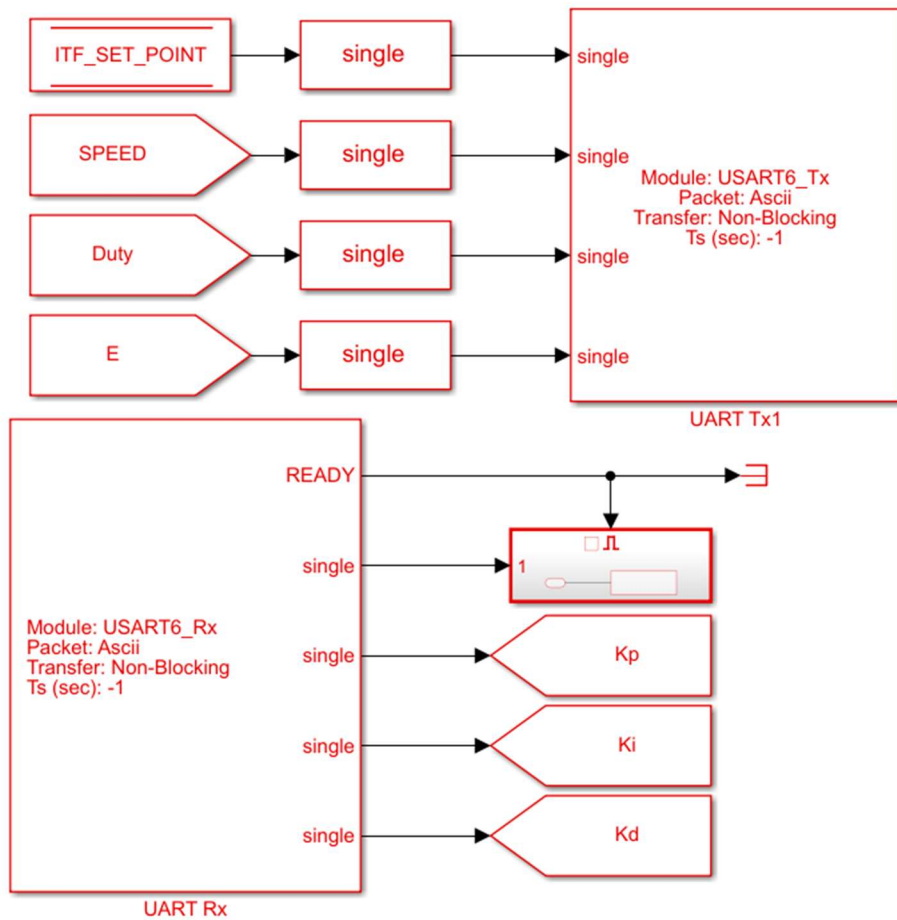
Cài đặt chọn dòng vi điều khiển STM32F407VG

Chọn tần số lấy mẫu là 50ms

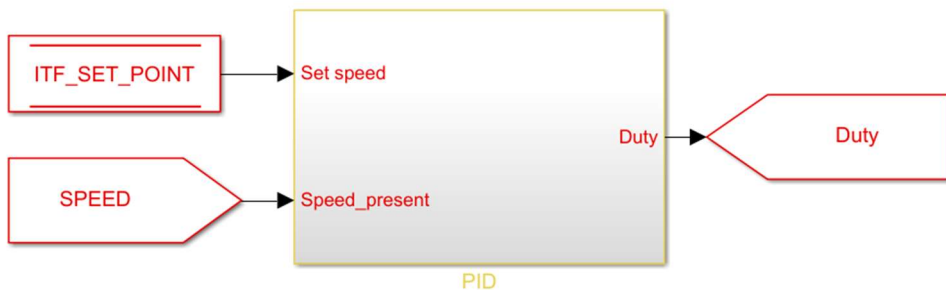
Cài đặt UART với tốc độ là 115200 với chân Tx/Rx là C6/C7



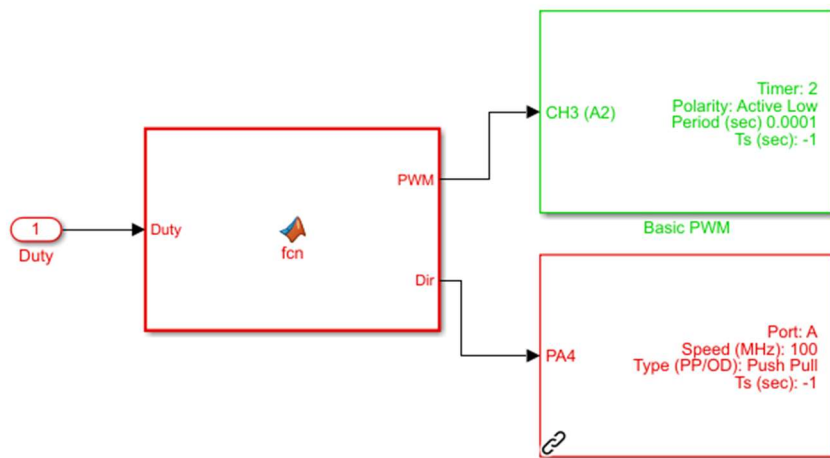
Cài đặt nút nhấn UP/DOWN/RESET ở các chân PD0/PD2/PD4 với cấu hình điện trở kéo xuống (tích cực mức cao)



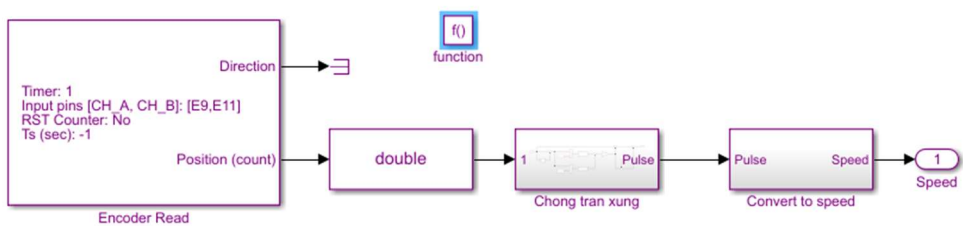
Gửi nhận các giá trị hệ thống thông qua UART



Dựa vào giá trị tốc độ cài và giá trị tốc độ phản hồi từ encoder thông qua PID tính toán giá trị Duty cần thiết để điều khiển động cơ



Dựa vào giá trị Duty được tính toán từ thuật toán PID mà cho ra xung PWM ở chân PA2 và chiều hoạt động của động cơ ở chân PA4.



Dựa vào ngắt mỗi 50ms của timer 9 chương trình đọc số xung encoder và chuyển đổi thành giá trị tốc độ.