

LỜI MỞ ĐẦU

Ngày nay trong mọi lĩnh vực khoa học kỹ thuật luôn xuất hiện khái niệm kỹ thuật vi xử lý và điều khiển, với sự trợ giúp của máy tính kỹ thuật vi xử lý và điều khiển đã có sự phát triển mạnh mẽ đặc biệt là sự phát triển nhanh chóng của các họ vi xử lý và điều khiển với những tính năng mới.

Và xuất phát từ những sự phát triển đó em đã nghiên cứu và thiết kế một mạch dùng vi điều khiển đó là: “Xây dựng mô hình điều khiển động cơ DC servo bằng vi điều khiển”

Động cơ DC servo đã xuất hiện nhiều trong công nghiệp và trong sản xuất. Hầu như trong bất cứ dây chuyền sản xuất nào hay trong các xưởng, nhà máy, xí nghiệp...đều cũng sử dụng động cơ DC servo. Một số ứng dụng cơ bản của động cơ DC servo trong công nghiệp như là điều khiển vị trí, vận tốc, gia tốc, trong các cơ cấu servo máy CNC, băng tải, cơ cấu robot...

Đề tài này giúp em hiểu rõ hơn về vi điều khiển, đồng thời tích lũy kiến thức đặc biệt là những kinh nghiệm trong quá trình lắp mạch thực tế song do thời gian và kiến thức có hạn, nên mạch thiết kế còn nhiều thiếu sót. Em rất mong nhận được sự góp ý của các thầy cô để có thể nâng cao chất lượng của đề tài tốt nghiệp này.

Em xin chân thành cảm ơn !

CHƯƠNG 1: GIỚI THIỆU ĐỘNG CƠ SERVO

1.1. PHÂN LOẠI ĐỘNG CƠ SERVO

1.1.1. Động cơ Servo DC

- Điều khiển động cơ 1 chiều: Dẫn động chạy dao máy công cụ điều khiển số NC/CNC đòi hỏi hệ điều khiển phải có khả năng điều khiển đồng thời cả tốc độ và vị trí. Mặc dù với sự phát triển của công nghiệp điện tử, động cơ xoay chiều điều khiển tốc độ bằng biến tần ngày càng phát triển mạnh mẽ nhưng động cơ Servo DC vẫn được sử dụng phổ biến trong các máy công cụ điều khiển số. Những năm trước 1995 của thế kỉ trước 95% động cơ dùng trong xích chuyên động chạy dao máy động cơ NC/CNC đều được sử dụng động cơ DC điều khiển Servo. Động cơ Servo DC có 2 loại: động cơ 1 chiều có chổi than và động cơ 1 chiều không có chổi than.

1.1.1.1. Động cơ Servo DC có chổi than

Động cơ servo dòng một chiều DC chổi than được trình bày trên (hình 1.1) gồm 4 thành phần cơ bản: stator của động cơ DC là một nam châm vĩnh cửu, cuộn dây phần ứng lắp trên roto. Trong quá trình hoạt động, từ trường cố định được sinh ra từ nam châm vĩnh cửu gắn trên stator tương tác với dòng từ sinh ra từ cuộn dây trên roto khi có dòng điện chạy qua nó. Quá trình tương tác đó sinh ra moment tác động lên trục roto. Moment này biểu diễn theo phương trình

$$T_m = k_e \cdot \Phi \cdot I_a \cdot \sin \Theta \quad (1)$$

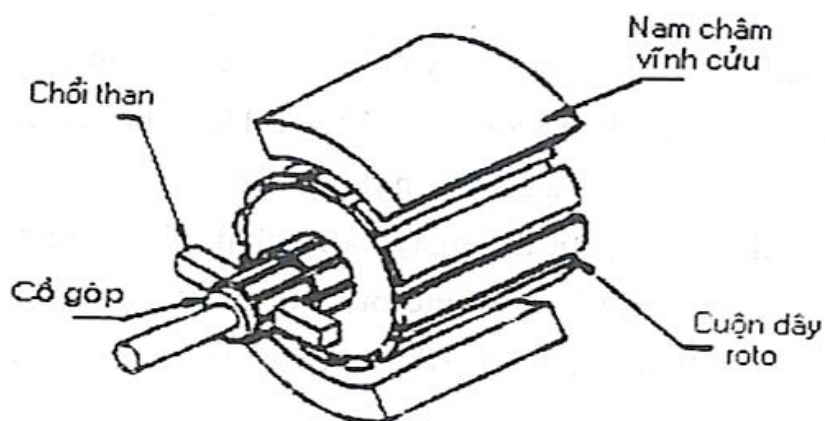
Trong đó : T_e = moment động cơ ;

K_e = hệ số động cơ ;

Φ = mật độ dòng từ

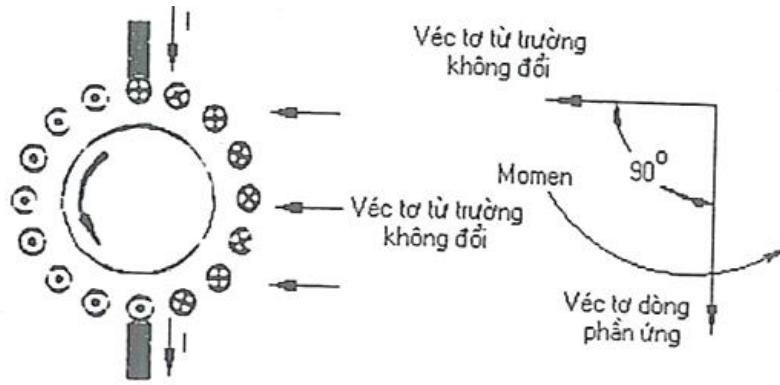
I_a = dòng phần ứng

Θ = góc giữa vectơ từ trường cố định và vectơ dòng



Hình 1.1: Cấu tạo động cơ Servo DMC chổi than

Công thức (1) cho thấy phần tử $\sin\Theta$ ảnh hưởng tới momen trên trục động cơ. Hình 1.2 chỉ ra quan hệ giữa vector từ trường cố định và vector dòng qua phần ứng. moment trên trục động cơ tăng dần từ $\Theta = 0^\circ$ và lớn nhất khi góc $\Theta = 90^\circ$ có nghĩa là khi vector từ trường cố định vuông góc với vector dòng phần ứng, moment trên trục động cơ là lớn nhất khi và khi $\Theta = 0^\circ$ vector dòng phần ứng song song với vector từ trường cố định, tại đó moment trên trục là nhỏ nhất. Để đảm bảo moment trên trục động cơ luôn đạt được giá trị lớn nhất cần thiết phải điều khiển chuyển mạch cấp điện cho cuộn dây roto sao cho vector dòng phần ứng luôn luôn vuông góc với từ trường cố định. Với cách điều khiển quá trình cấp điện như trên, mômen động cơ sẽ biến thiên tỉ lệ với dòng cấp cho cuộn dây phần ứng.



Hình 1.2: Vectơ từ trường cố định và vectơ dòng qua phần ứng

Một mối liên hệ khác giữa các thông số của động cơ một chiều là tốc độ quay của rôto tỷ lệ với sức điện động phản điện động phản điện sinh ra trong cuộn dây phần ứng

Mômen và tốc độ của động cơ Servo DC điều khiển có thể mô tả bằng hai phương trình sau:

$$T_{dc} = K_m \cdot I_u \quad (2)$$

$$E_b = K_b \cdot \omega \quad (3)$$

Trong đó: T_{dc} - là mômen từ, Nm

I_u - dòng điện trong cuộn dây phần ứng, A

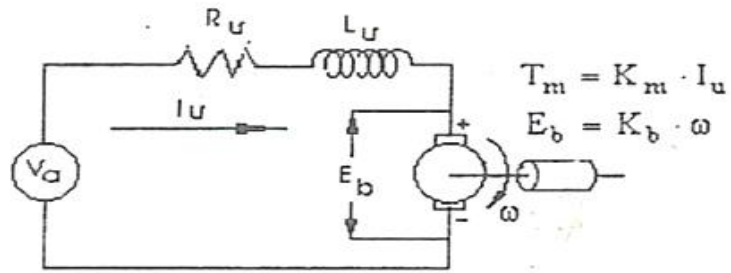
E_b - điện áp phản điện (emf), V

K_m - hệ số mômen, kgm/A

K_b - hệ số điện, đơn vị đo vôn trên vòng trên phút

ω - vận tốc quay của động cơ, vòng/phút

Mạch động cơ Servo DC chỉ ra trên hình 1.3



Hình 1.3: Mạch động cơ Servo DC

Từ định luật Kirchhoff ta có phương trình mạch

$$V_u = K_b \cdot \omega + R_u \cdot I_u + L_u \cdot \left(\frac{dI_u}{dt}\right) \quad (4)$$

Thành phần L_u nhỏ hơn so với R_u nên có thể bỏ qua L_u . Bỏ qua L_u phương trình sẽ là:

$$V_u - R_u I_u = K_b \omega \quad (5)$$

Phương trình mômen tải T_m đặt trên trục động cơ :

$$T_m = T_d + T_s + T_c \quad (6)$$

Và

$$T_d = J_{dc} (d\omega / dt)$$

$$T_s = f_{dc} \omega$$

$$T_c = J_m (d\omega / dt) + f_m$$

Trong đó :

T_d - mômen động

T_s - mômen tĩnh

T_c - mômen cản

J_{dc} - mômen quán tính roto động cơ

f_{dc} - hệ số sức cản nhớt của tải

J_m - mômen quán tính tải

f_m - hệ số sức cản nhớt của tải

Để động cơ quay thì mômen động cơ phải bằng với mômen tải:

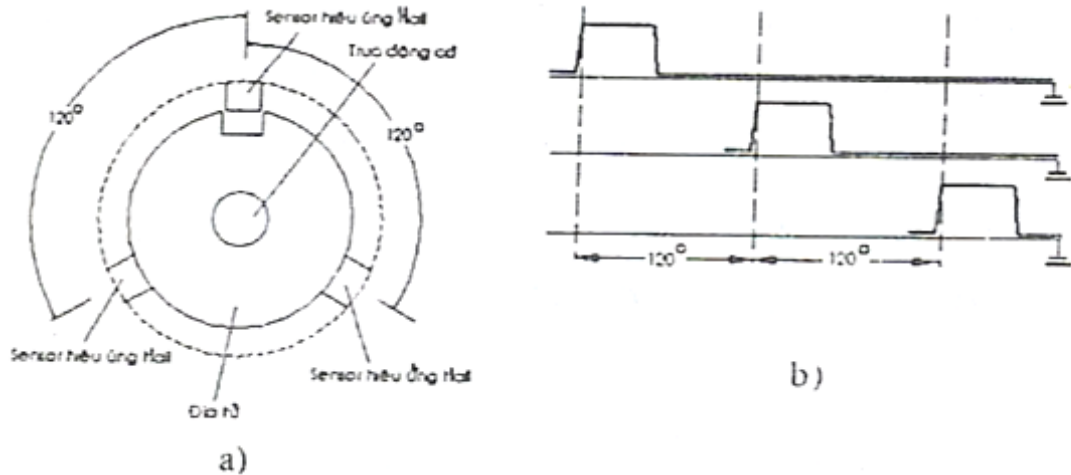
$$T_m = T_{dc} = K_m \cdot I_u \quad (7)$$

Ưu điểm của động cơ Servo DC chổi than là đơn giản trong điều khiển và giá thành sản phẩm rẻ. Tuy nhiên sử dụng chổi than cơ khí gây ra ồn, tăng nhiệt độ trên vành góp và quán tính rô to cao khi giảm tốc độ. Để khắc phục các nhược điểm trên người ta đã sử dụng động cơ Servo DC không chổi than.

1.1.1.2. Động cơ Servo DC không có chổi than

Động cơ Servo DC không có chổi than được sử dụng phổ biến trong máy công cụ điều khiển số. Cấu trúc của nó về cơ bản giống như động cơ Servo DC chổi than nhưng khác ở chỗ các cuộn pha của động cơ lắp trên Stato và Rôto là nam châm vĩnh cửu. Rôto được chế tạo từ vật liệu ferit hoặc samari coban. Rôto làm từ vật liệu samari coban có khả năng tập trung từ cao và từ dư thấp. Nhưng giá thành rôto loại này cao hơn nhiều so với khi rôto làm từ vật liệu ferit. Vì vậy, nó chỉ dùng để chế tạo rôto cho động cơ công suất lớn. Tương tự như động cơ xoay chiều, từ trường quay trong động cơ DC không chổi than được sinh ra nhờ mạch điều khiển thứ tự cấp dòng cho các cuộn pha. Cuộn dây pha của động cơ không chuyển động vì vậy có thể sử dụng chuyển mạch bằng điện tử nên loại trừ bằng những nhược điểm tồn tại trong động cơ DC Servo chổi than.

Điều khiển các trục máy công cụ điều khiển số đòi hỏi điều khiển chính xác cả về vị trí và tốc độ. Vì vậy, động cơ Servo DC không chổi than cần phải có mạch phản hồi, tín hiệu phản hồi là tốc độ quay trục động cơ hoặc vị trí góc trục. Để đảm bảo chính xác chuyển động bàn máy, tín hiệu phản hồi phải được cấp liên tục cho mạch điều khiển. Trong công nghiệp thiết bị mạch phản hồi của động cơ Servo DC thường sử dụng là cảm biến tốc độ (Tachometer) chổi than hoặc không có chổi than, sensor hiệu ứng Hall, resolver, synchro và encoder.



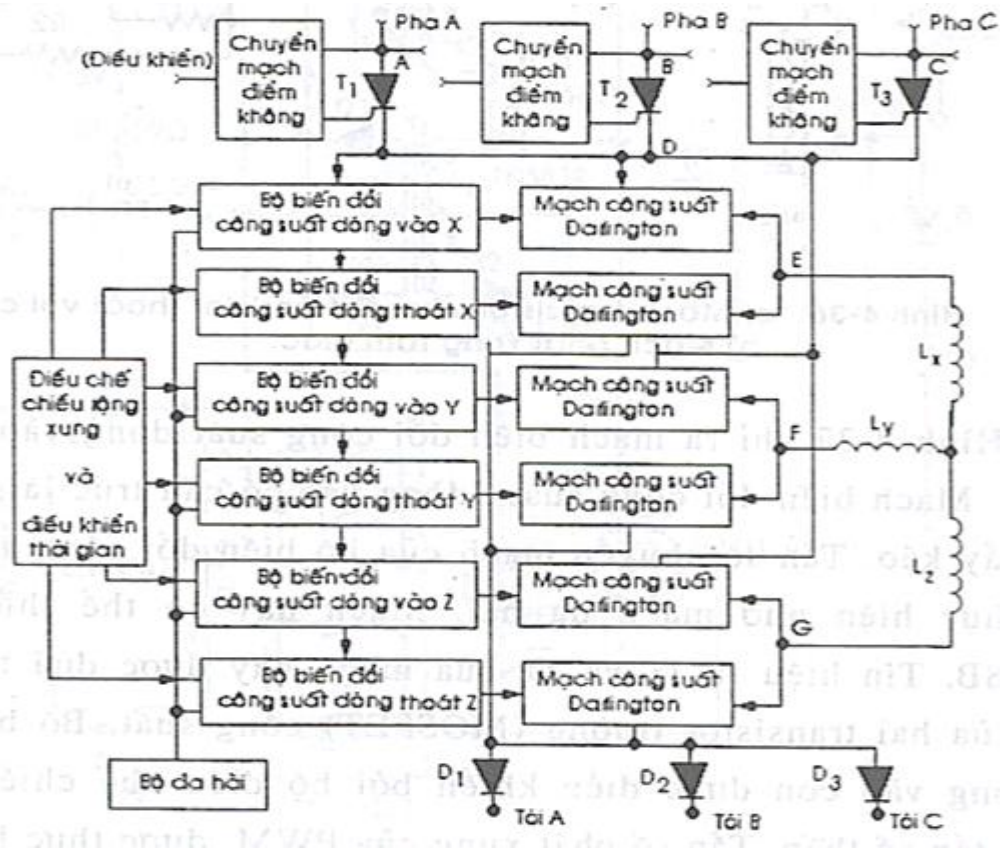
Hình 1.4: a) Sensor hiệu ứng Hall và đĩa từ lắp ở đuôi động cơ

b) Tín hiệu chuyển mạch sensor hiệu ứng Hall sinh ra trong một vòng

Phương pháp chuyển mạch hiệu ứng Hall được sử dụng khá phổ biến trong điều khiển động cơ Servo DC. Trong động cơ Servo DC 3 pha không chổi than người ta đặt cố định 3 sensor hiệu ứng Hall lên vỏ phía đuôi động cơ và cách đều 120° quanh trục động cơ. Để lấy tín hiệu sensor hiệu ứng Hall, một đĩa từ như chỉ ra trên (hình 1.4a) được lắp trên đuôi trục động cơ và trên đĩa người ta cắt một rãnh. Khi một trong 3 sensor hiệu ứng Hall đi qua rãnh, trong khoảng thời gian ngắn dòng từ bị mất và kết quả là trên đầu ra của sensor hiệu ứng Hall V_H không có điện áp V_h (V_h – điện áp hiệu ứng Hall). Tín hiệu ra từ sensor thường được đưa qua mạch Trigger Smith để hiệu chỉnh lại thành xung chữ nhật.

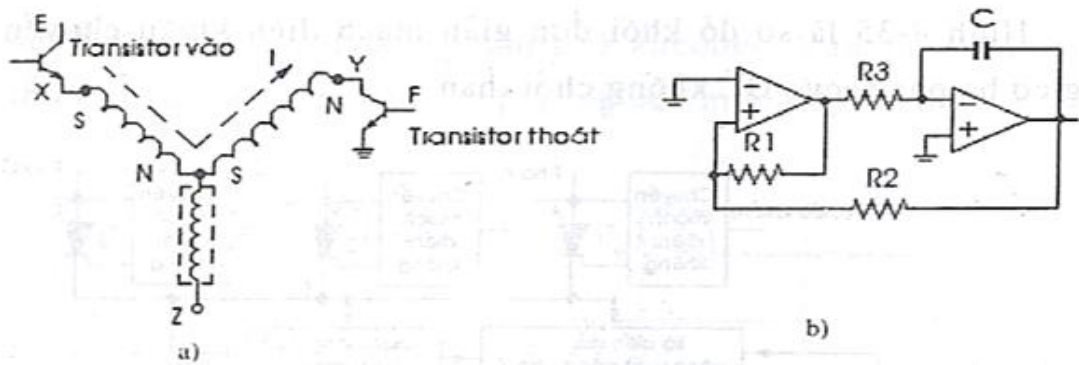
Hình 1.4b chỉ ra tín hiệu đưa ra từ sensor hiệu ứng Hall trong 1 vòng quay của trục động cơ. Tín hiệu này có thể dùng để điều khiển chuyển mạch Transistor công suất ở tín hiệu ra của điều khiển động cơ. Đồng thời cũng có thể dùng để xác định vị trí của động cơ.

Hình 1.5 là sơ đồ khối đơn giản mạch điều khiển chuyển mạch động cơ 3 pha động cơ Servo DC không chổi than.



Hình 1.5: Sơ đồ khối mạch điều khiển chuyển mạch cho động cơ ba pha

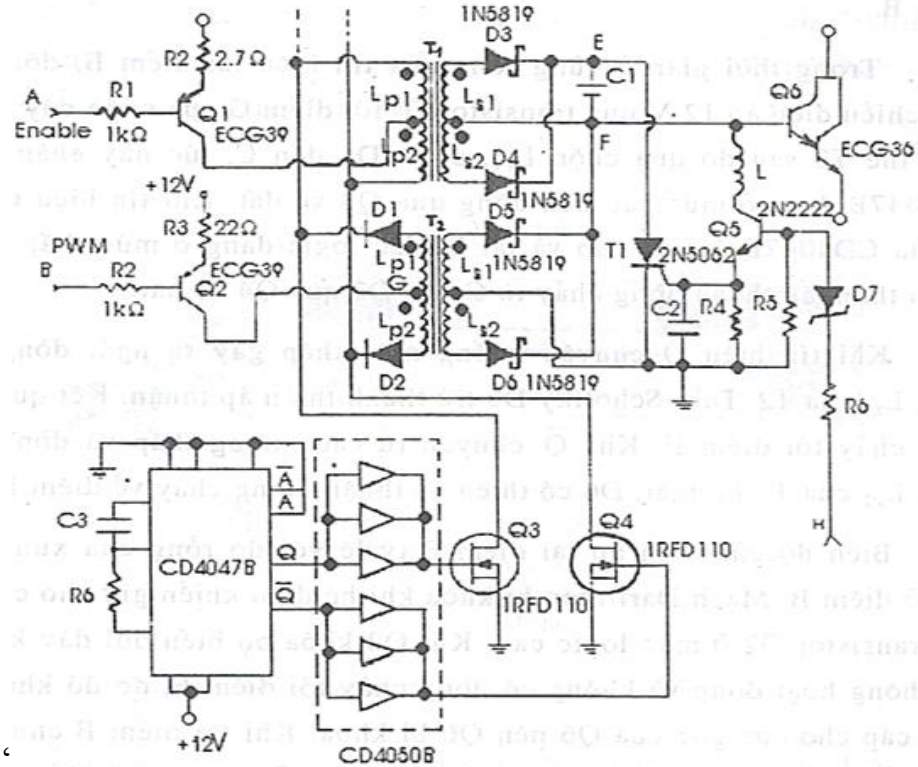
Hệ gồm 6 bộ biến đổi công suất dòng vào và dòng thoát được điều khiển bởi mạch điều chế chiều rộng xung PWM (Pul Width Modulator). Mục đích của bộ biến đổi này là khống chế dòng điện cấp cho 1 trong 3 cuộn dây L_x , L_y , L_z . Tín hiệu chuyển mạch điều khiển động cơ gởi tới chân điều khiển Transitor công suất dòng vào và Transitor công suất thoát lắp theo kiểu Darlington. Hình 1.6a chỉ ra mạch Transitor dòng vào, dòng thoát, cuộn pha L_x , L_y và tương tự như thế với cuộn L_x và L_z hoặc L_y và L_z



Hình 1.6: a) Mạch transistor vào và transistor thoát với các cuộn pha
b) Mạch phát xung tam giác

Hình 1.7 chỉ ra mạch biến đổi công suất dòng vào và mạch tín hiệu ra. Mạch biến đổi công suất 3 dòng vào có cấu trúc là mạch biến áp xung đẩy kéo. Tần số chuyển mạch của bộ biến đổi công suất dòng vào được thực hiện nhờ mạch đa hài. Mạch này có thể thiết lập từ IC CD4078B. Tín hiệu ra Q và $Q_{bù}$ của mạch này được đưa tới chân điều khiển của 2 chân Transistor trường (mosfet) công suất. Bộ biến đổi công suất dòng vào còn được điều khiển bởi bộ điều chế chiều rộng xung (PWM) tần số thấp. Tần số phát xung của PWM được thực hiện nhờ máy phát xung tam giác như chỉ ra trên hình 1.5b

Hình 1.7 là sơ đồ mạch của một trong 6 bộ biến đổi dòng. Điều khiển mạch đa hài và mạch biến đổi đẩy kéo hoạt động như sau: Khi chân tín hiệu ra Q của IC CD4047B ở mức cao và tín hiệu Enable (A) ở mức thấp, dòng chảy từ nguồn điện áp 1 chiều 12V qua Transistor Q_1 tới cuộn L_{p1} của biến áp T1 về C qua Transistor Q_3 và đất. ở thời điểm này không xuất hiện dòng trong cuộn L_{s1} chảy qua cuộn cảm L, D3 biến thiên áp ngược. Khi Q chuyển từ mức logic cao xuống mức logic thấp và Enable (A) không thay đổi mức tín hiệu, dòng chảy qua L_{p1} bị ngắt. Trong cuộn dây L_{s1} xuất hiện dòng chảy qua D3 hướng tới điểm E nạp điện cho tụ C1. Tại thời điểm này tín hiệu ra Q bù từ mức thấp chuyển lên mức cao.



Hình 1.7: Một trong sáu tầng biến đổi của hệ điều khiển động cơ DC không chổi than

Dòng chảy từ nguồn 12V qua cuộn L_{p2} của T1 hướng tới điểm D qua Q_4 về đất trong cuộn L_{s2} xuất hiện dòng điện chảy qua L_{s2} tới điểm E nạp điện cho tụ C1. Như vậy với tần số thấp của tín hiệu Enable, tụ C1 nhanh chóng được nạp đến mức xác định vì xung dòng ở điểm C và D có tần số di tri ổn định cho nên nạp điện áp tại điểm E gần như không thay đổi. Điện thế ở tại điểm E là điện áp cho Anôt của Triristor T1.

Điện áp tại điểm F điều khiển biên độ dòng gốc của khuếch đại công suất Dalington và điện áp này là hàm của tín hiệu chuyển mạch ở điểm B

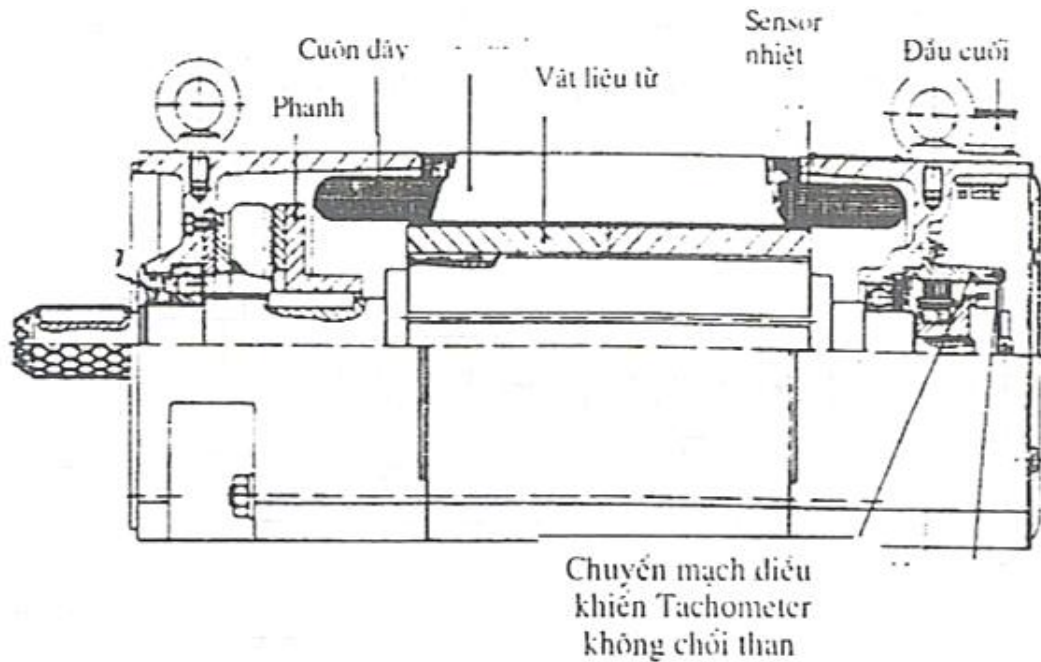
Trong thời gian ở vùng rỗng của tín hiệu ở điểm B dòng điện 1 chiều điện áp 12V qua Tristor Q_2 tới điểm G của cuộn dây L_{p1} của biến thế T2 sau đó qua cuộn L_{p1} , diode D1 đến C, lúc này chân Q của CD4047B ở mức cao và tại B

mức logic thấp D2 trở thành điện áp thuận dòng chảy từ G qua D2 qua Q4 về đất.

Khi tín hiệu Q chuyển xuống mức thấp gây ra ngắt dòng chảy trong L_{p1} của T2 diode Schottky D5 trở thành điện áp thuận. Kết quả là có dòng chảy tới điểm F. Khi Q_{bu} chuyển từ cao xuống và dòng chảy trong L_{p2} của T2 bị ngắt D6 có thiên áp thuận dòng chảy về điểm F

Biên độ của điện áp tại điểm F tỉ lệ với độ rộng của xung chữ nhật điểm B. Mạch Darlington bị khóa khi hệ điều khiển giữ cho cực gốc của Transitor Q2 ở mức logic cao. Khi q2 khóa bộ biến đổi đẩy kéo thứ 2 không hoạt động và không có chảy tới điểm F, do đó không có dòng cấp cho cực gốc của Q6 nên Q6 bị khóa. Khi tại điểm B chuyển từ logic cao sang logic thấp Transitor Q2 mở. Độ rộng xung tại điểm B tăng lên làm cho dòng gốc của Transitor Q6 tăng lên và khi độ rộng của xung vào B giảm xuống dòng gốc của Q6 cũng giảm xuống. Như vậy dòng collector và emitter của Darlington là hàm của độ rộng tín hiệu chuyển mạch.

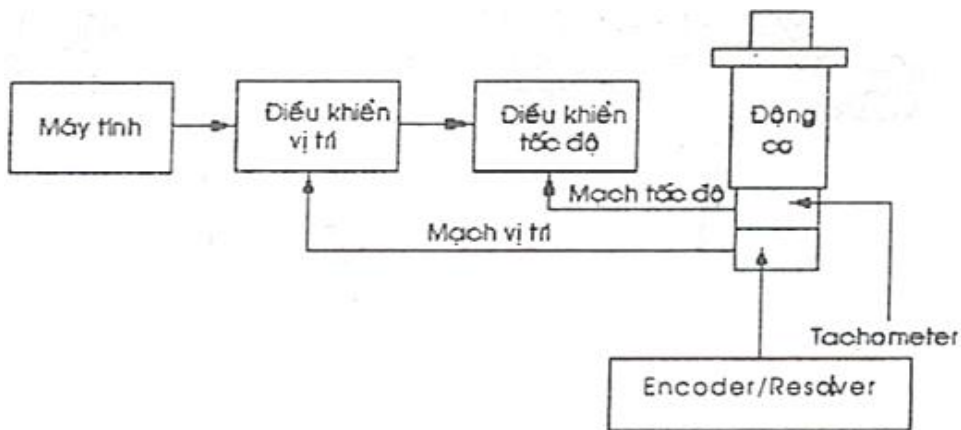
Tiristor T1, Transitor Q5 và Diode zener D7 hình thành mạch bảo vệ động cơ Servo và chống quá áp cho mạch điều khiển. Để không chế quá áp người ta nối điểm H trong hình 1.7 với điểm trong hình 1.6. Tiristor T1, transitor Q5 và diode zener D7, điện trở R_3 và R_4 được lắp như chỉ ra trên hình 1.7. Trong mạch điện trở R_3 và điện trở R_4 chọn đủ lớn để với điện áp bình thường Q5 luôn bị khóa do đó Tiristor T1 cũng luôn bị khóa. Khi điện áp tại D vượt quá điện áp định mức đủ lớn Transitor Q5 mở, Transitor T1 mở nên điện áp tại điểm E và F gần bằng không và mạch Darlington khóa. Chú ý rằng trong quá trình điện áp tại D vượt quá điện áp cho phép, Transitor Q2 đang ở trạng thái mở.



Hình 1.8: Kết cấu động cơ DC không chổi than

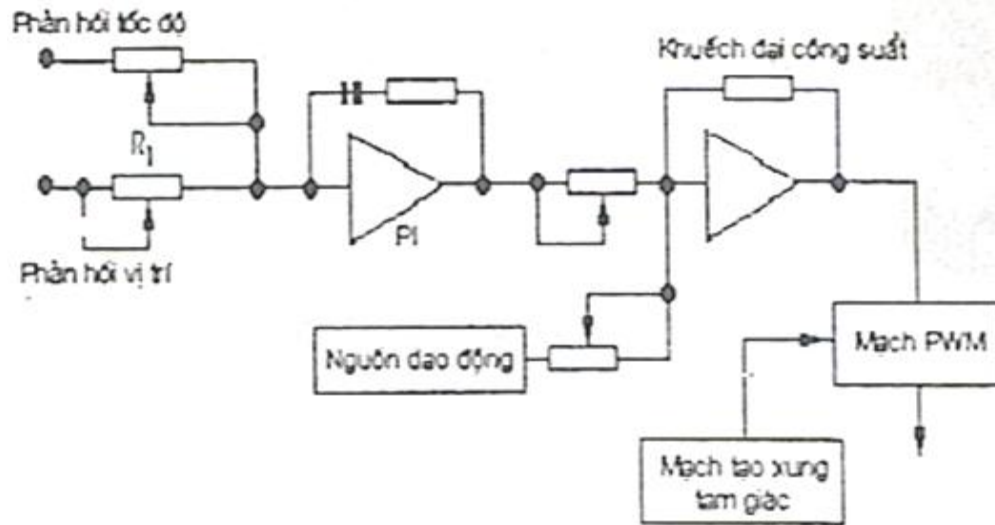
Hình 1.8 là kết cấu của động cơ DC không chổi than. Trên động cơ bố trí hệ thống phanh, sensor đo tốc độ, chuyển mạch hiệu ứng Hall, sensor kiểm tra nhiệt độ động cơ. Trong than đòi hỏi hệ điều khiển động cơ cung cấp tín hiệu điều khiển cả vị trí và cả tốc độ. Có 2 kiểu cơ bản của hệ điều khiển động cơ Servo: tương tự và số.

Hệ điều khiển Servo kiểu tương tự là sử dụng mạch điện để thực hiện bù sai số vị trí và tốc độ. Hệ gồm 4 cụm điều khiển cơ bản: máy tính điều khiển vị trí, điều khiển tốc độ và động cơ một chiều không chổi than. Mối quan hệ giữa các cụm điều khiển chỉ rõ trong hình với tín hiệu phản hồi vị trí từ bộ biến đổi encoder hoặc Resolver qua mạch phản hồi để hồi sinh ra sai số tốc độ và sai số được đưa đến hệ điều khiển tốc độ để xử lý cho phù hợp với vị trí. Hệ điều khiển tốc độ chứa mạch phản hồi tốc độ sinh ra từ Tachometer. Tín hiệu được so sánh với tín hiệu được đưa ra từ hệ điều khiển vị trí và sinh ra điện áp và dòng phù hợp bù cho sai số vị trí và tốc độ.



Hình 1.9: Sơ đồ khối của hệ điều khiển động cơ DC kiểu tương tự CNC

Hình 1.10 là một kiểu mạch điều khiển động cơ Servo DC dùng trong máy công cụ điều khiển số CNC. Điện áp lỗi tương tự CNC và tín hiệu phản hồi của Tachometer gửi tới mạch điều chỉnh (PI) để sinh ra tín hiệu điều khiển vị trí. Tín hiệu sinh ra từ bộ điều chỉnh PI và tín hiệu từ mạch dao động đưa tới mạch khuếch đại công suất trước khi tới mạch điều chế chiều rộng xung (PWM). Xung tam giác là xung chuẩn được sinh ra từ mạch phát xung. Xung này được gửi bộ điều chế chiều rộng xung. Trên hình 1.10 điện trở R1 là điện trở khuếch đại của mạch điều khiển vị trí.

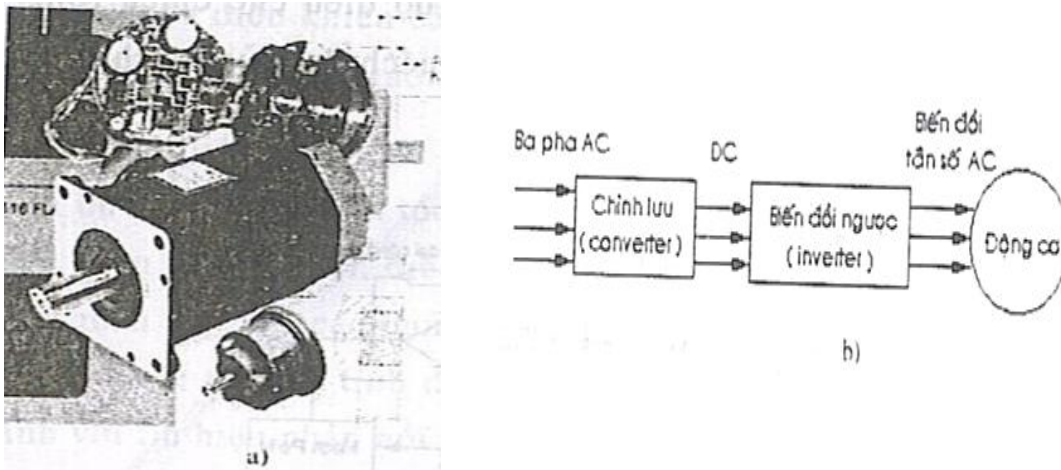


Hình 1.10: Mạch điều khiển động cơ Servo DC

1.1.2. Động cơ AC Servo

Nhờ sự phát triển vượt bậc của công nghệ điều khiển điện, hiện nay chuyển động chạy dao trong máy công cụ điều khiển số dùng khá phổ biến động cơ AC Servo. Hình -11 chỉ ra hình dạng ngoài của động cơ AC Servo.

Nhưng nhược điểm của động cơ AC Servo là hệ điều chỉnh tốc độ động cơ phức tạp và đắt tiền so với động cơ DC. Hệ điều khiển tốc độ động cơ AC Servo dựa trên cơ sở biến đổi tần số. Tốc độ động cơ được xác định theo tần số nguồn. Một trong những phương pháp điều khiển tốc độ động cơ AC Servo là biến đổi dòng xoay chiều thành dòng một chiều nhờ bộ chỉnh lưu 3 pha, sau đó biến đổi dòng 1 chiều thành dòng xoay chiều nhưng ở tần số đã được lựa chọn. Hình 1.11 là sơ đồ khối đơn giản hệ điều khiển tốc độ động cơ AC Servo.

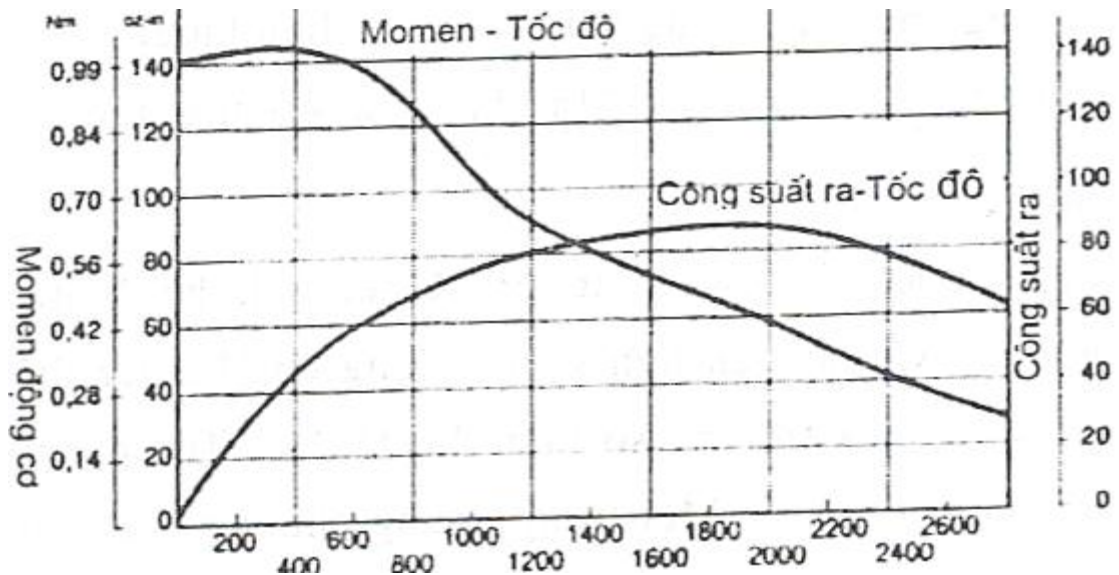


Hình 1.11: a) Dạng ngoài động cơ AC

b) Sơ đồ điều khiển tốc độ động cơ AC

1.1.3. Lựa chọn động cơ

Khi lựa chọn động cơ người thiết kế phải xem xét nhiều yếu tố và các đặc trưng về dải tốc độ, sự biến đổi momen tốc độ, tính thuận nghịch, chu kỳ làm việc, momen khởi động và công suất yêu cầu.



Hình 1.12: Đường cong momen tốc độ động cơ bước

Đặc biệt lưu ý tới đường cong momen tốc độ động cơ bởi vì các đường cong này cho ta những thông tin quan trọng. Hình 1.12 chỉ ra đường cong

momen tốc độ khác nhau với điện áp tiêu thụ tương ứng. Để lựa chọn lựa công suất chúng ta cần chọn lựa các vấn đề sau:

1.1.3.1. Momen khởi động động cơ.

Momen ở tốc độ quay bằng 0 được gọi là momen khởi động cơ. Để động cơ tự khởi động được, động cơ phải sinh ra momen lớn hơn momen ma sát và momen tải đặt lên trục của nó. Nếu gọi α là gia tốc góc của động cơ và được đo bằng Rad/s^2 , T_m là momen động cơ, $T_{\text{tải}}$ là momen tải đặt lên trục động cơ và J là momen quán tính của Rôto và tải ta có quan hệ:

$$A=(T_m-T_{\text{tải}})/J \quad (8)$$

1.1.3.2. Tốc độ cực đại của động cơ.

Nhìn vào đồ thị quan hệ momen tốc độ, tại điểm momen bằng 0 xác định tốc độ cực đại của động cơ. Cần phải nhớ rằng tại tốc độ này động cơ không qua momen và tốc độ này gọi là tốc độ không tải.

1.1.3.3. Công suất yêu cầu tải.

Công suất yêu cầu đặt biệt quan trọng đối với động cơ, vì vậy người thiết kế phải lựa chọn động cơ có công suất tương ứng với công suất yêu cầu trong chu kỳ làm việc.

1.1.3.4. Nếu hệ dẫn động yêu cầu điều chỉnh tốc độ:

Tốt nhất là lựa chọn động cơ đồng bộ hoặc động cơ một chiều.

1.1.3.5. Nếu hệ yêu cầu điều khiển cả vị trí và tốc độ.

Trong trường hợp vị trí góc thực hiện theo vị trí rời rạc hoặc gia số, tốt nhất là động cơ bước. Động cơ bước có thể điều khiển tốc độ bằng cách thay đổi tần số cấp xung và chỉ dùng trong các mạch điều khiển nhỏ có nghĩa là không có mạch phản hồi. Động cơ bước chỉ dùng trong trường hợp tải trọng tải nhỏ và không thể dùng trong trường hợp đòi hỏi tốc độ quá cao. Trong trường hợp yêu cầu điều khiển cả vị trí và tốc độ, ví dụ trong các thiết bị chuyển động theo

chương trình số, người ta thường sử dụng động cơ Servo. Động cơ Servo là động cơ AC, DC hoặc động cơ một chiều không có chổi than có mạch phản hồi vị trí. Động cơ Servo đắt hơn động cơ bước.

1.1.3.6. Hệ thống cần hay không cần giảm tốc.

Thông thường tải được điều khiển ở dải tốc độ thấp và momen lớn. Đặc tính của động cơ ở tốc độ cao momen thấp vì vậy cần hợp tốc độ để giảm tốc độ đầu ra. Khi dùng hợp tốc độ quán tính tải cũng thay đổi theo và sự thay đổi này thể hiện trong công thức:

$$J_c = J_{t\grave{a}i} (\omega_{t\grave{a}i} / \omega_d)^2 \quad (9)$$

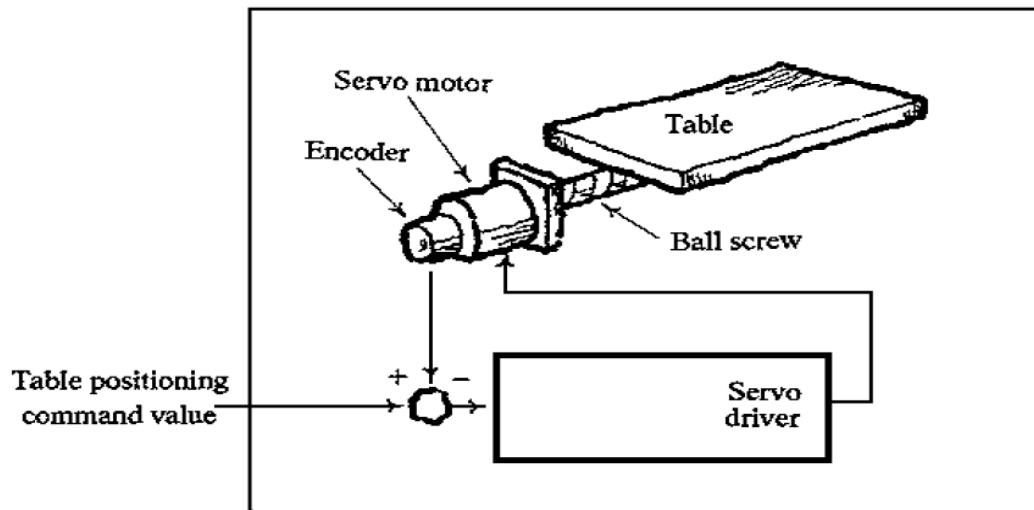
Trong đó: $\omega_{t\grave{a}i}$ - Tốc độ góc của tải, Rad/s

ω_d - Tốc độ góc của động cơ, rad/s

1.2. HỆ THỐNG SERVO

1.2.1. Hệ thống Servo là gì ?

“Servo” bắt nguồn từ tiếng Hy Lạp **Secvus** (vervant). Hệ thống được gọi là : “Hệ thống Servo” chấp hành trung thành với một lệnh



Hình 1.13: Mô hình một hệ thống Servo

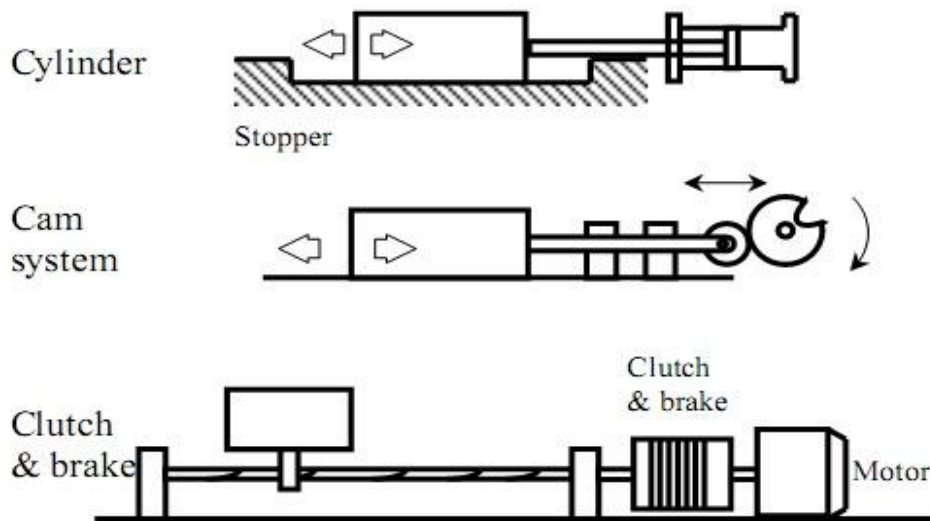
***Cơ cấu định vị:**

Hệ thống servo không đơn giản chỉ là một phương pháp thay thế điều khiển vị trí và tốc độ của các cơ cấu cơ học, ngoài những thiết bị cơ khí đơn giản, hệ thống servo bây giờ đã trở thành một hệ thống điều khiển chính trong phương pháp điều khiển vị trí và tốc độ. Sau đây là một số ví dụ về các cơ cấu định vị:

Cơ cấu định vị đơn giản :

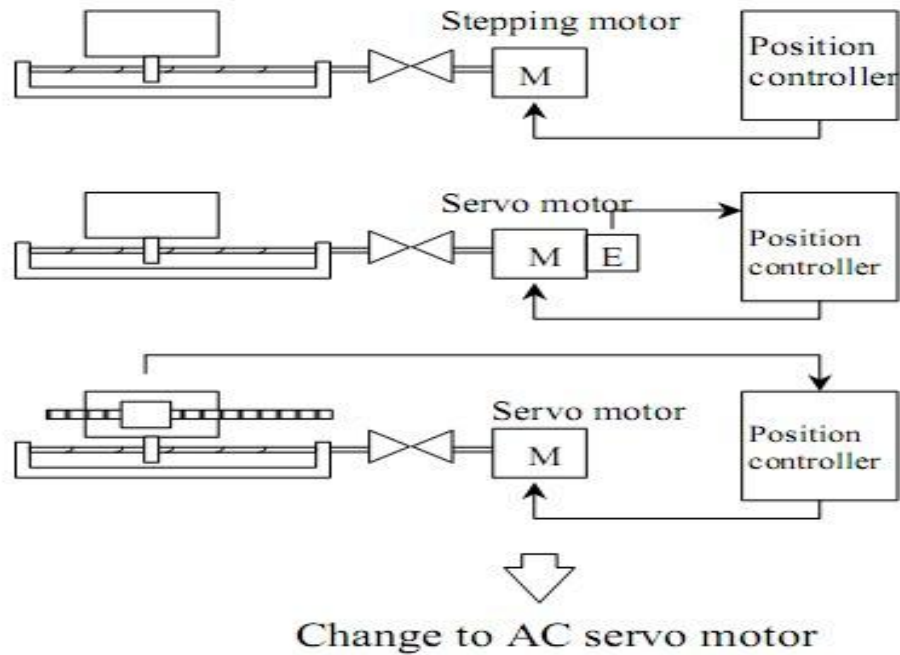
*Các ví dụ về cơ cấu này đó là xy lanh hay trục cam hay bộ ly hợp và phanh

hãm



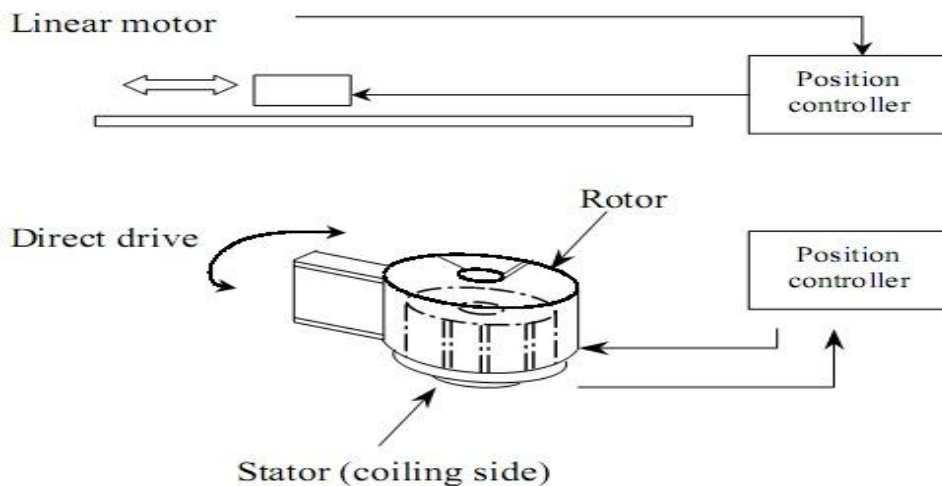
Hình 1.14: Xy lanh hay trục cam hay bộ ly hợp và phanh hãm

Ưu điểm của cơ cấu này đó là đơn giản, rẻ tiền, và có thể hoạt động ở tốc độ cao. Cơ cấu định vị linh hoạt điều khiển bởi servo motor. Cơ cấu này có thể được điều khiển vòng hở, nửa kín hay vòng kín



Hình 1.15: Điều khiển vị trí linh hoạt bởi động cơ servo

Ưu điểm của cơ cấu này đó là độ chính xác và đáp ứng tốc độ cao, có thể dễ dàng thay đổi vị trí đích và tốc độ của cơ cấu chấp hành. Cơ cấu chuyển động định hướng Cơ cấu này chuyển động theo hướng nhất định được chỉ định từ bộ điều khiển. Chuyển động có thể là chuyển động tịnh tiến hay quay.

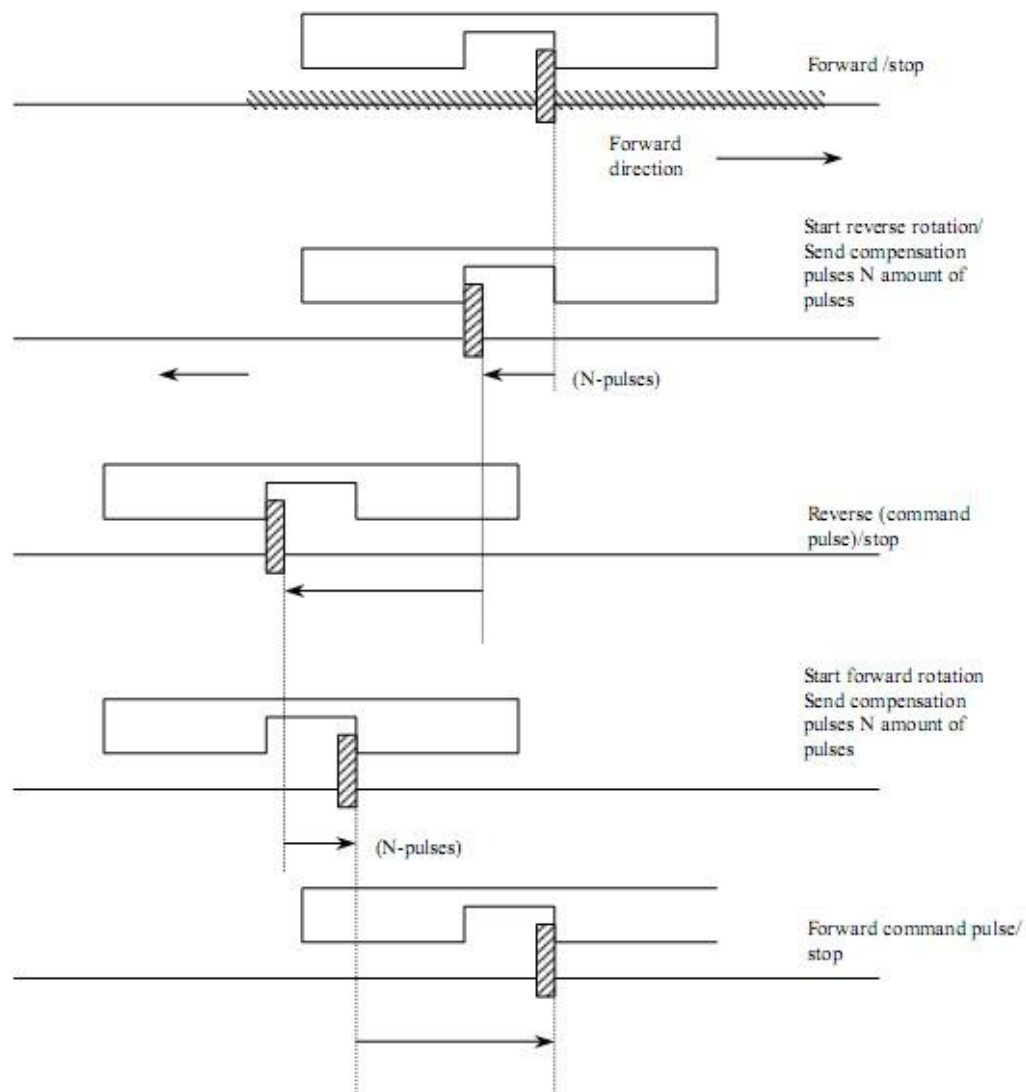


Hình 1.16: Điều khiển chạy trực tiếp

- Ưu điểm là cơ cấu chấp hành đơn giản và nâng cao tuổi thọ hộp số truyền động (do truyền động khá êm).

***Backlash và hiệu chỉnh:**

- Backlash hiệu đó là giới hạn chuyển động của một hệ thống servo. Tất cả các thiết bị cơ khí đều có một điểm trung tính giữa chuyển động hoặc quay theo chiều dương và âm (cũng giống như động cơ trước khi đảo chiều thì vận tốc phải giảm về 0). Xét một chuyển động tịnh tiến lui và tới như trong hình sau:



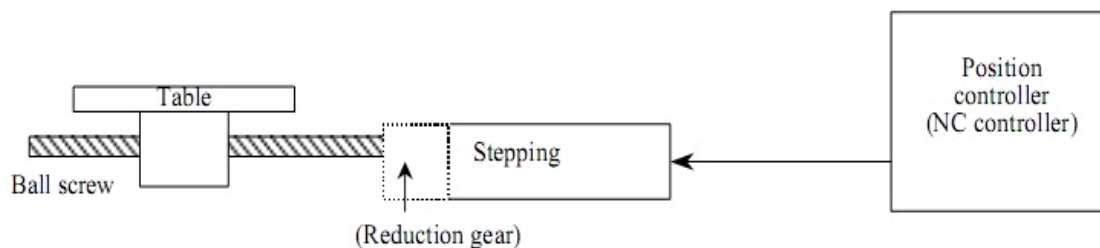
Hình 1.17: Sự giạt lùi của cơ khí

Chuyển động tịnh tiến này được điều khiển bởi một động cơ servo. Chuyển động tới và lui được giới hạn bởi một khoản trống như trong hình. Như vậy động cơ sẽ quay theo chiều dương hoặc chiều âm theo một số vòng nhất định để chuyển động của thanh quét lên toàn bộ khoản trống đó nhưng không được vượt quá khoản trống (đây là một trong những điều kiện cốt lõi của việc điều khiển động cơ servo). Giới hạn này được gọi là backlash. Tuy nhiên trong thực tế độ động cơ quay những vòng chính xác để con trượt trượt chính xác và quét lên toàn bộ khoản trống trên là rất khó thực hiện nếu không có một sự bù trừ cho nó. Và trong hệ thống servo nhất thiết có những hàm lệnh thực hiện việc bù trừ, hiệu chỉnh này. Như trong hình vẽ trên, hệ thống servo gửi xung lệnh hiệu chỉnh cộng/trừ số lượng xung lệnh điều khiển và các xung lệnh hiệu chỉnh này sẽ không được tính đến trong bộ đếm xung.

1.2.2. Hệ thống điều khiển

Có ba dạng :

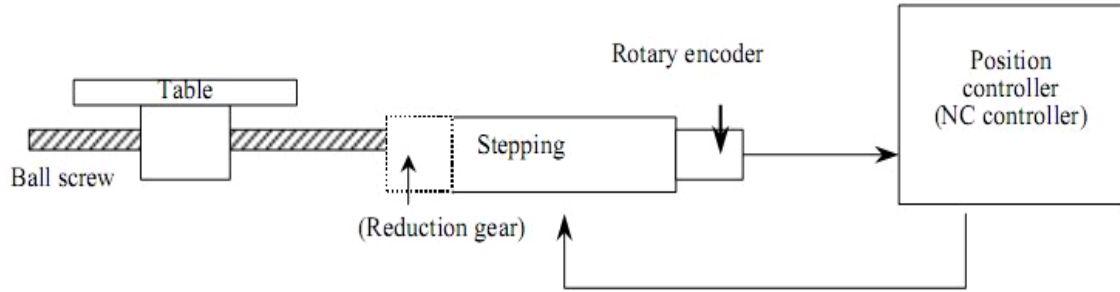
Điều khiển vòng hở:



Hình 1.18: Điều khiển vòng hở

Nghĩa là bộ điều khiển vị trí chỉ đặt lệnh cho động cơ quay mà thôi.

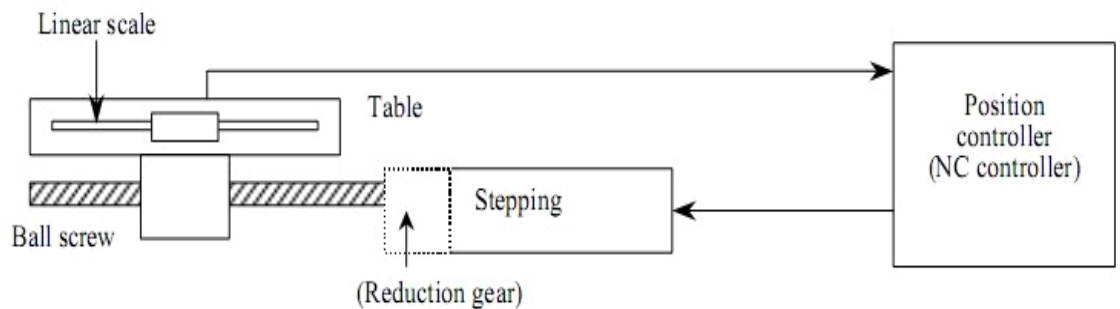
Điều khiển nửa kín:



Hình 1.19: Điều khiển nửa kín

Ở đây số vòng quay của step motor được mã hóa và hồi tiếp về bộ điều khiển vị trí. Nghĩa là đến đây thì động cơ step chỉ quay một số vòng nhất định tùy thuộc vào “lệnh” của bộ điều khiển vị trí, nói cách khác bộ điều khiển vị trí có thể ra lệnh cho chạy hoặc dừng động cơ theo một lập trình sẵn có tùy thuộc vào ý đồ của người thiết kế.

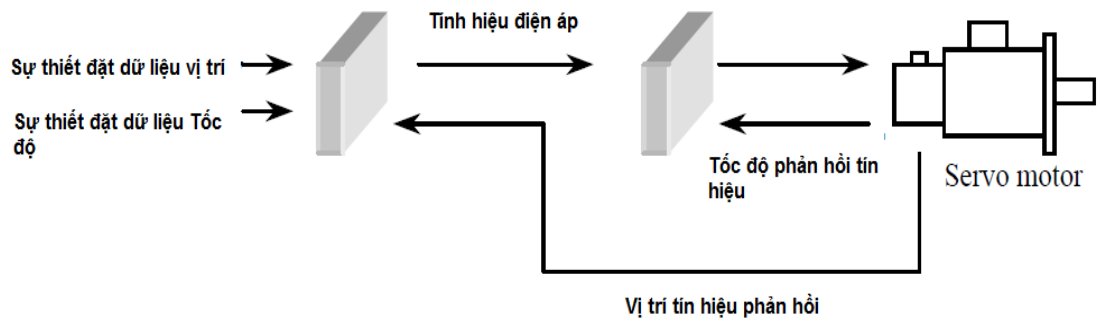
Điều khiển vòng kín



Hình 1.20: Điều khiển vòng kín

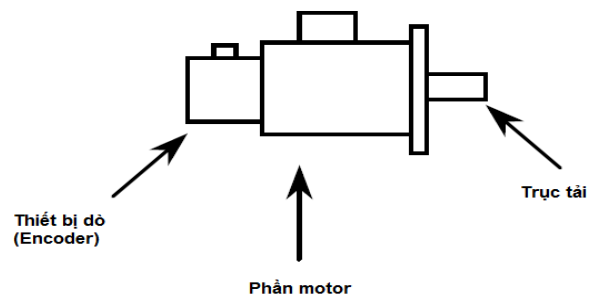
Vòng hồi tiếp lúc này không phải hồi tiếp từ trực động cơ về mà vòng hồi tiếp lúc này là hồi tiếp vị trí của bàn chạy thông qua một thước tuyến tính. Lúc này bộ điều khiển vị trí không điều khiển số vòng quay của motor nữa mà nó điều khiển trực tiếp vị trí của bàn chạy. Nghĩa là các sai số tĩnh do sai khác trong các bánh răng hay hệ thống truyền động được loại bỏ.

1.2.3. Cấu hình của hệ thống servo:



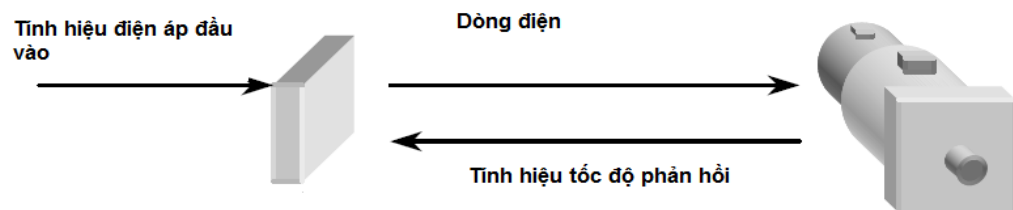
Hình 1.21: Cấu tạo của hệ thống servo

Sự khác biệt của động cơ servo so với những động cơ sử dụng cảm ứng từ nói chung là nó có một máy dò để phát hiện tốc độ quay và vị trí.

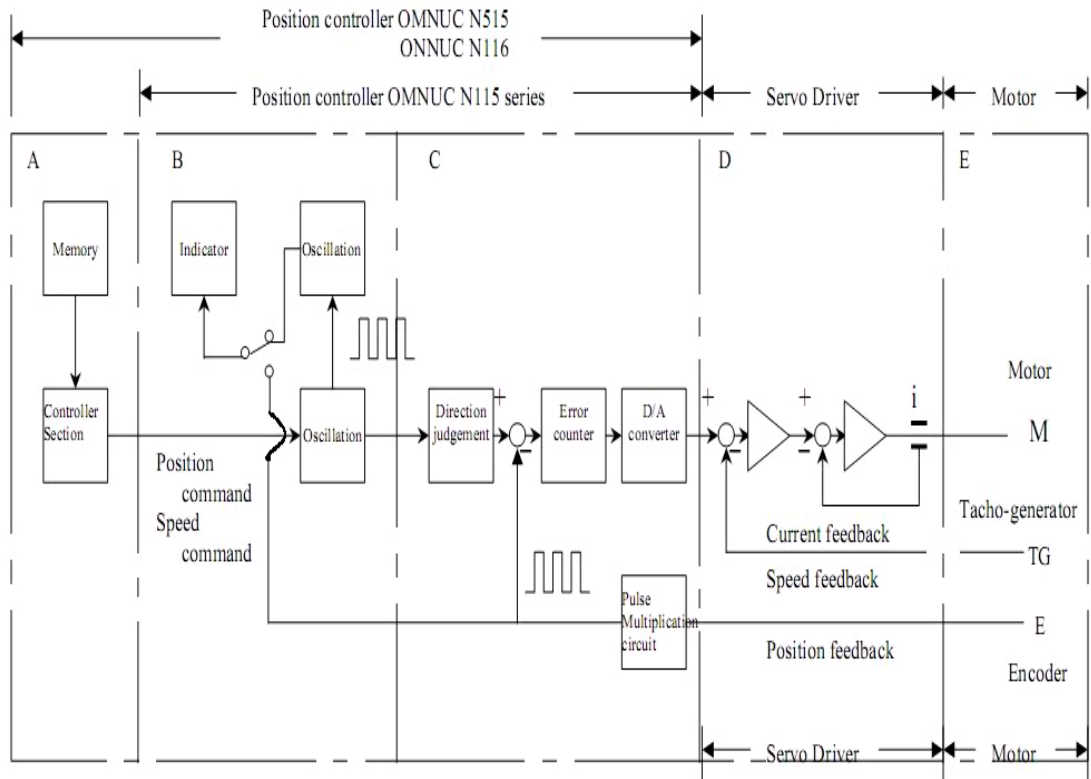


Bộ điều khiển (Tín hiệu đầu vào)

Điều khiển tốc độ động cơ servo quay với một tốc độ tương ứng với tín hiệu điện áp đầu vào. Vì vậy nó giám sát tốc độ quay của động cơ trong mọi thời điểm.



Sơ đồ khối điều khiển động cơ servo với 2 vòng hồi tiếp vị trí và tốc độ:

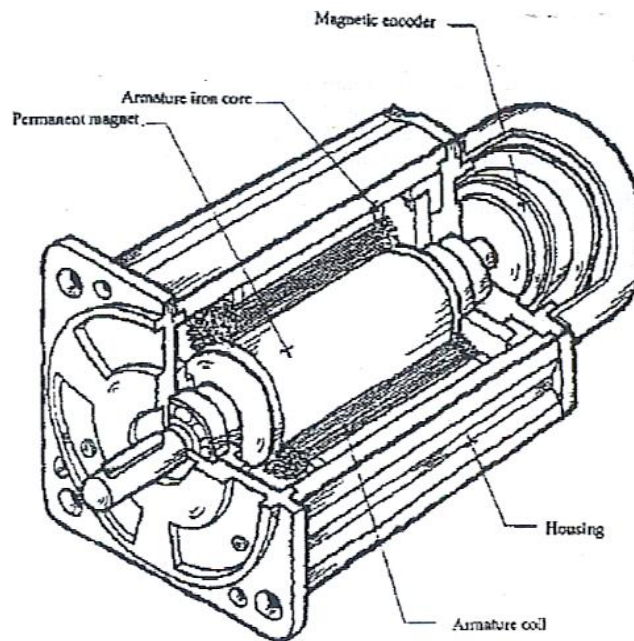


Hình 1.22: Sơ đồ khối động cơ servo

Trong đó phần A B C là phần so sánh xử lý tín hiệu hồi tiếp và hiệu chỉnh lệnh. Phần D E là cơ cấu thực thi và hồi tiếp. Các phần A B C thì khá phổ dụng trong các sơ đồ khối điều khiển, phần D E thì tùy các thiết bị sử dụng mà chúng có khác nhau đôi chút nhưng về bản chất chúng hoàn toàn giống nhau.

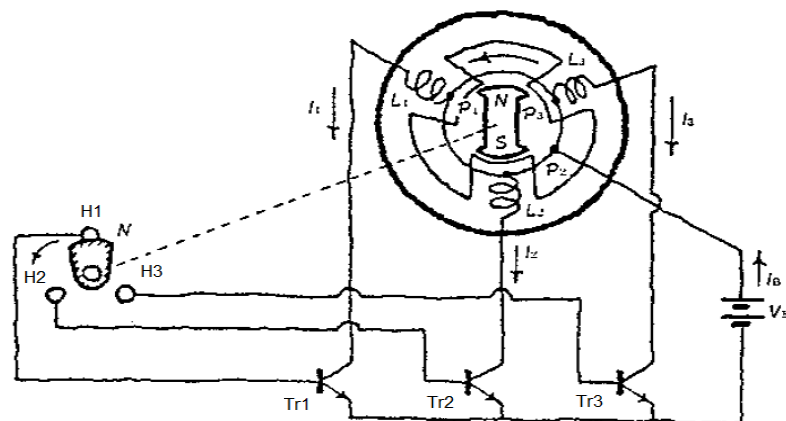
1.2.4. Cấu tạo và nguyên lý làm việc của động cơ Servo

1.2.4.1. Cấu tạo của động cơ servo

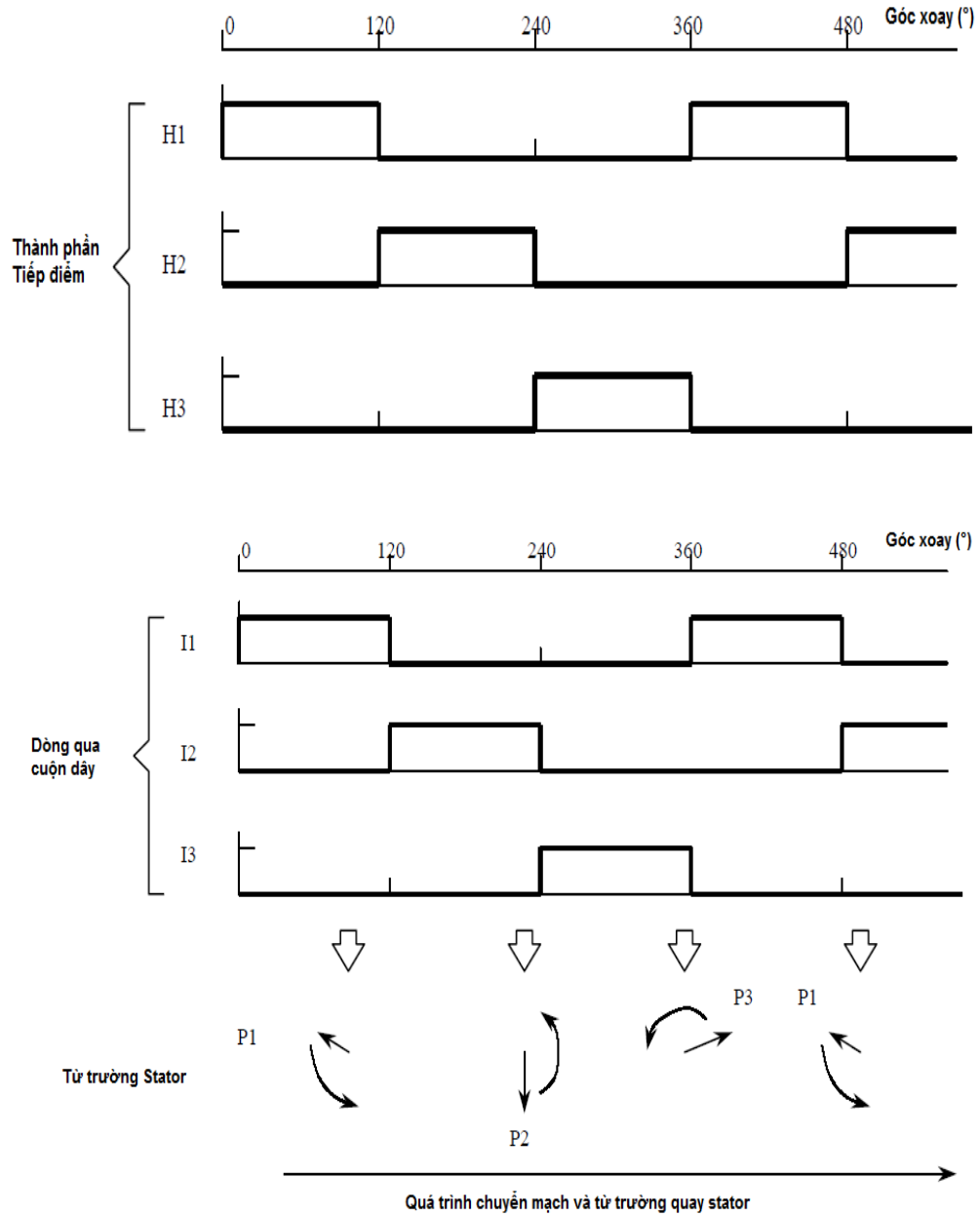


Hình 1.23: Sơ đồ cấu tạo của động cơ servo

1.2.4.2. Nguyên lý làm việc của động cơ servo



Nguyên tắc của động cơ có 3 cực



Hình 1.24: Sơ đồ nguyên lý làm việc của động cơ servo

Qua phần trình bày ở chương một đã giới thiệu được một số loại động cơ servo mà hiện nay đang được sử dụng rộng rãi trên thị trường nước ta.

CHƯƠNG 2: VI ĐIỀU KHIỂN ATMEGA8

2.1. GIỚI THIỆU CHUNG

Atmega8 là một con vi điều khiển thuộc dòng Mega AVR của hãng Atmel.
Dòng vi điều khiển này có nhiều tính năng hoạt động nổi trội như:

Rất tiết kiệm năng lượng, hiệu suất cao

CPU có kiến trúc RISC, có 130 lệnh hầu hết chúng chỉ thực hiện chỉ trong một chu kỳ xung clock

Có 23 thanh ghi đa dụng

Tốc độ tối đa lên 16MIPS với thạch anh là 16MHZ

Bộ nhớ phân đoạn có độ bền cao không dễ bay hơi

Có 8KB bộ nhớ flash lập trình ISP

512 bytes EEPROM

1KB SRAM

Chu kỳ ghi và xóa 10.000 lần cho bộ nhớ Flash ROM và 100.000 cho EEPROM

Dữ liệu không bị mất sau nhiều năm

Có tính năng bảo mật

Tính năng ngoại vi:

2 bộ timer/counter 8 bit và 1 bộ so sánh

1 bộ Timer/counter 16 bit

Bộ đếm thời gian thực với dao động riêng

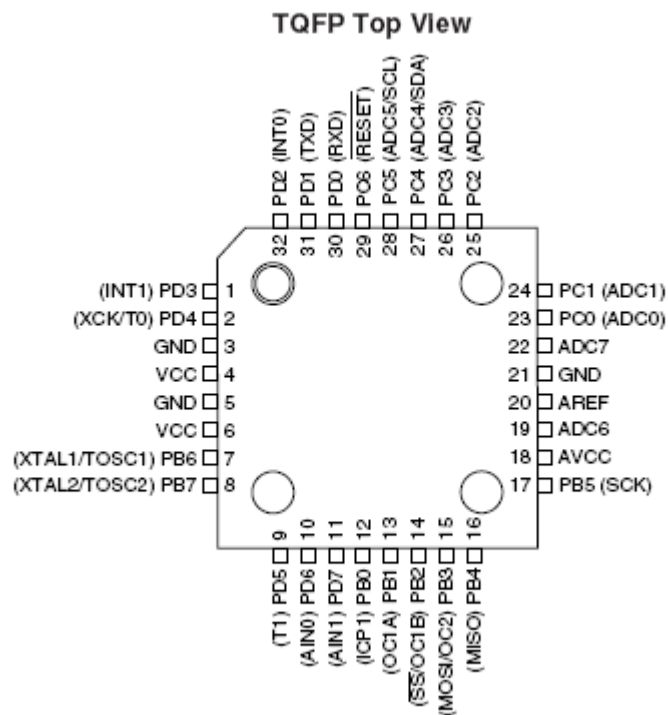
3 kênh PWM

6 kênh ADC 10 bit cho kiểu vỏ PDIP và 8 kênh ADC 10 bit cho kiểu vỏ

TQFP

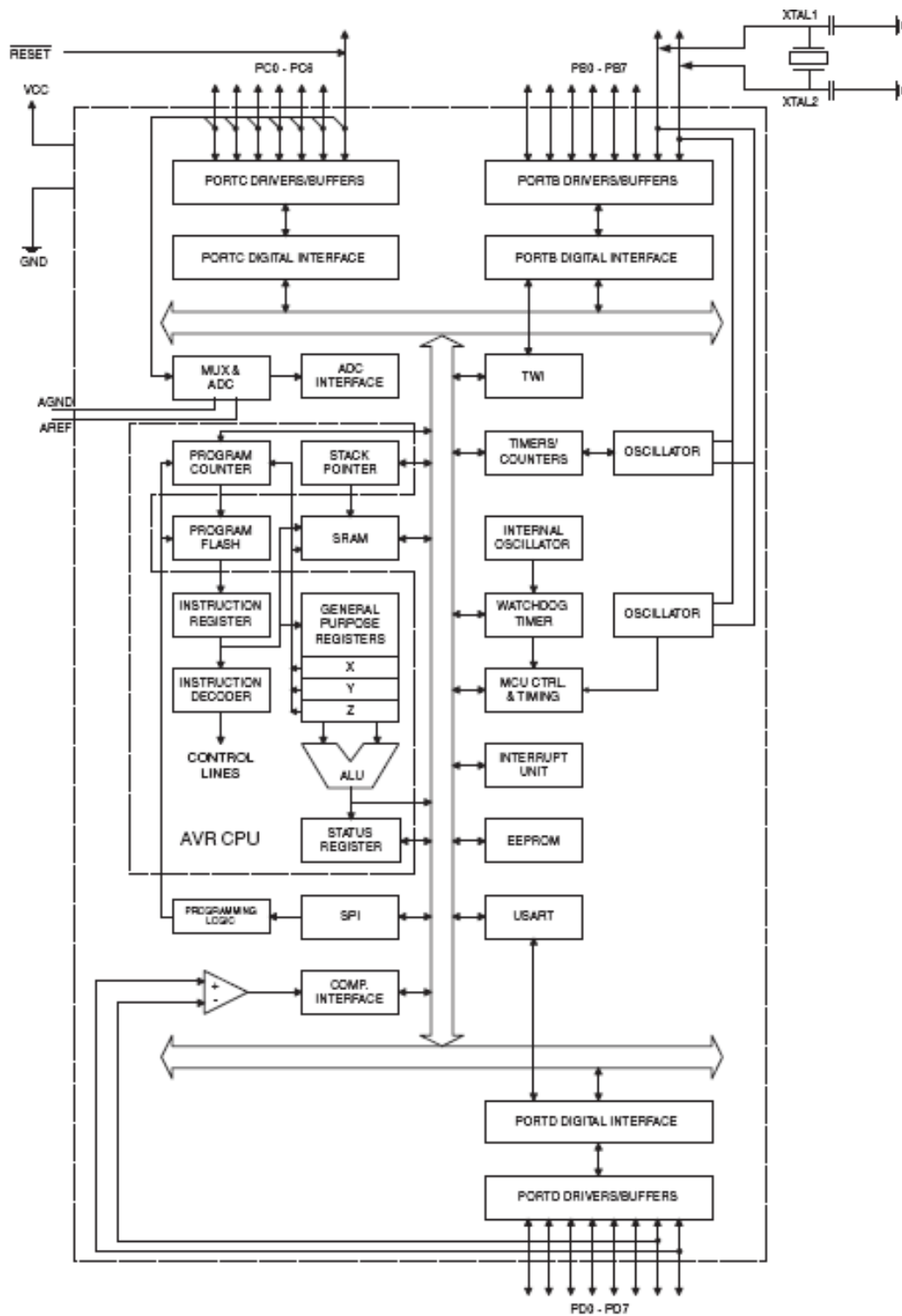
Giao tiếp nối tiếp TWI

Lập trình nối tiếp USART giao tiếp nối tiếp SPI master/slave
 Bộ so sánh Analog on chip
 Tính năng đặc biệt của Atmega8
 Hiệu chuẩn bộ dao động RC nội
 Bộ nguồn ngắt bên trong và bên ngoài
 Năm chế độ: Idle, giảm nhiễu ADC, tiết kiệm năng lượng, Power down, chế độ chờ.
 Có 23 ngõ vào ra
 Được đóng gói trong 32 chân kiểu TQFP
 Điện áp hoạt động 4,5 – 5,5v
 Tần số hoạt động 0 – 16 mhz
 Sơ đồ chân Atmega8



Hình 2.1: Sơ đồ chân atmega8

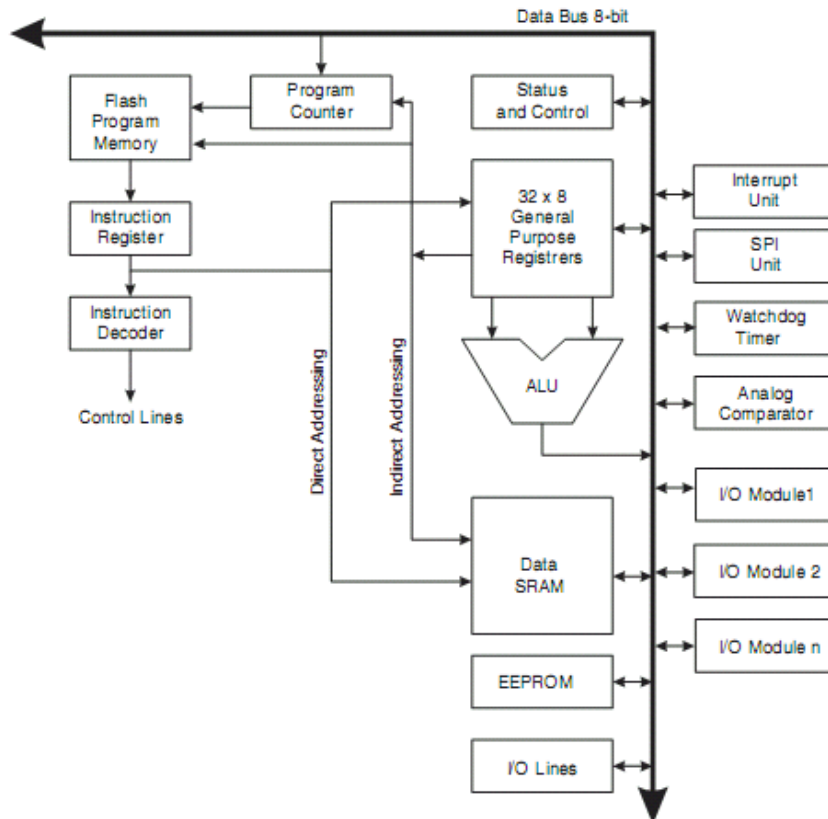
Sơ đồ khối



Hình 2.2: Sơ đồ khối atmega8

2.2. CẤU TRÚC NHÂN AVR

CPU của AVR có chức năng bảo đảm sự hoạt động chính xác của các chương trình. Do đó nó phải có khả năng truy cập bộ nhớ, thực hiện các quá trình tính toán, điều khiển các thiết bị ngoại vi và quản lý ngắt.



Hình 2.3: Cấu trúc nhân AVR

AVR sử dụng cấu trúc Harvard, tách riêng bộ nhớ và các bus cho chương trình và dữ liệu. Các lệnh được thực hiện chỉ trong một chu kỳ xung clock. Bộ nhớ chương trình được lưu trong bộ nhớ Flash.

2.2.1. ALU

ALU làm việc trực tiếp với các thanh ghi chức năng chung. Các phép toán được thực hiện trong một chu kỳ xung clock. Hoạt động của ALU được chia làm 3 loại: đại số, logic và theo bit.

2.2.1.1. Thanh ghi trạng thái

Đây là thanh ghi trạng thái có 8 bit lưu trữ trạng thái của ALU sau các phép tính số học và logic

Bit	7	6	5	4	3	2	1	0	
\$3F (\$5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

C: Carry Flag ;cờ nhớ (Nếu phép toán có nhớ cờ sẽ được thiết lập)

Z: Zero Flag ;Cờ zero (Nếu kết quả phép toán bằng 0)

N: Negative Flag (Nếu kết quả của phép toán là âm)

V: Two's complement overflow indicator (Cờ này được thiết lập khi tràn số bù 2)

V, For signed tests ($S=N \text{ XOR } V$) S: N

H: Half Carry Flag

T: Transfer bit used by BLD and BST instructions(Được sử dụng làm nơi chung gian trong các lệnh BLD,BST).

I: Global Interrupt Enable/Disable Flag (Đây là bit cho phép toàn cục ngắt. Nếu bit này ở trạng thái logic 0 thì không có một ngắt nào được phục vụ.)

2.2.1.2. Con trỏ ngăn xếp (SP)

Là một thanh ghi 16 bit nhưng cũng có thể được xem như hai thanh ghi chức năng đặc biệt 8 bit. Có địa chỉ trong các thanh ghi chức năng đặc biệt là \$3E (Trong bộ nhớ RAM là \$5E). Có nhiệm vụ trỏ tới vùng nhớ trong RAM chứa ngăn xếp.

Bit	15	14	13	12	11	10	9	8	
\$3E (\$5E)	-	-	-	-	-	-	SP9	SP8	SPH
\$3D (\$5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

2.2.1.3. Quản lý ngắt

Ngắt là một cơ chế cho phép thiết bị ngoại vi báo cho CPU biết về tình trạng sẵn sàng cho đổi dữ liệu của mình. Ví dụ: Khi bộ truyền nhận UART nhận được một byte nó sẽ báo cho CPU biết thông qua cờ RXC hoặc khi nó đã truyền được một byte thì cờ TX được thiết lập...

Khi có tín hiệu báo ngắt CPU sẽ tạm dừng công việc đang thực hiện lại và lưu vị trí đang thực hiện chương trình (con trỏ PC) vào ngăn xếp sau đó trở tới vector phục vụ ngắt và thực hiện chương trình phục vụ ngắt đó chờ tới khi gặp lệnh RETI (return from interrupt) thì CPU lại lấy PC từ ngăn xếp ra và tiếp tục thực hiện chương trình mà trước khi có ngắt nó đang thực hiện. Trong trường hợp mà có nhiều ngắt yêu cầu cùng một lúc thì CPU sẽ lưu các cờ báo ngắt đó lại và thực hiện lần lượt các ngắt theo mức ưu tiên. Trong khi đang thực hiện ngắt mà xuất hiện ngắt mới thì sẽ xảy ra hai trường hợp. Trường hợp ngắt này có mức ưu tiên cao hơn thì nó sẽ được phục vụ. Trường hợp nó có mức ưu tiên thấp hơn thì nó sẽ bị bỏ qua.

Bộ nhớ ngăn xếp là vùng bất kì trong SRAM từ địa chỉ 0x60 trở lên. Để truy nhập vào SRAM thông thường thì ta dùng con trỏ X,Y,Z và để truy nhập vào SRAM theo kiểu ngăn xếp thì ta dùng con trỏ SP. Con trỏ này là một thanh ghi 16 bit và được truy nhập như hai thanh ghi 8 bit chung có địa chỉ :SPL :0x3D/0x5D(IO/SRAM) và SPH:0x3E/0x5E.

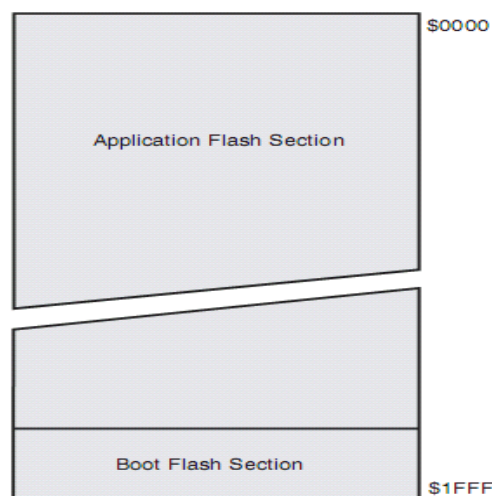
Khi chương trình phục vụ ngắt hoặc chương trình con thì con trỏ PC được lưu vào ngăn xếp trong khi con trỏ ngăn xếp giảm hai vị trí. Và con trỏ ngăn xếp sẽ giảm 1 khi thực hiện lệnh push. Ngược lại khi thực hiện lệnh POP thì con trỏ ngăn xếp sẽ tăng 1 và khi thực hiện lệnh RET hoặc RETI thì con trỏ ngăn xếp sẽ tăng 2. Như vậy con trỏ ngăn xếp cần được chương trình đặt trước giá trị khởi tạo ngăn xếp trước khi một chương trình con được gọi hoặc các ngắt được cho phép phục vụ. Và giá trị ngăn xếp ít nhất cũng phải lớn hơn 60H (0x60) vì 5FH trở lại là vùng các thanh ghi.

2.3. CẤU TRÚC BỘ NHỚ

AVR có 2 không gian bộ nhớ chính là bộ nhớ dữ liệu vào bộ nhớ chương trình. Ngoài ra Atmega8 còn có thêm bộ nhớ EEPROM để lưu trữ dữ liệu

2.3.1. Bộ nhớ chương trình (Bộ nhớ Flash)

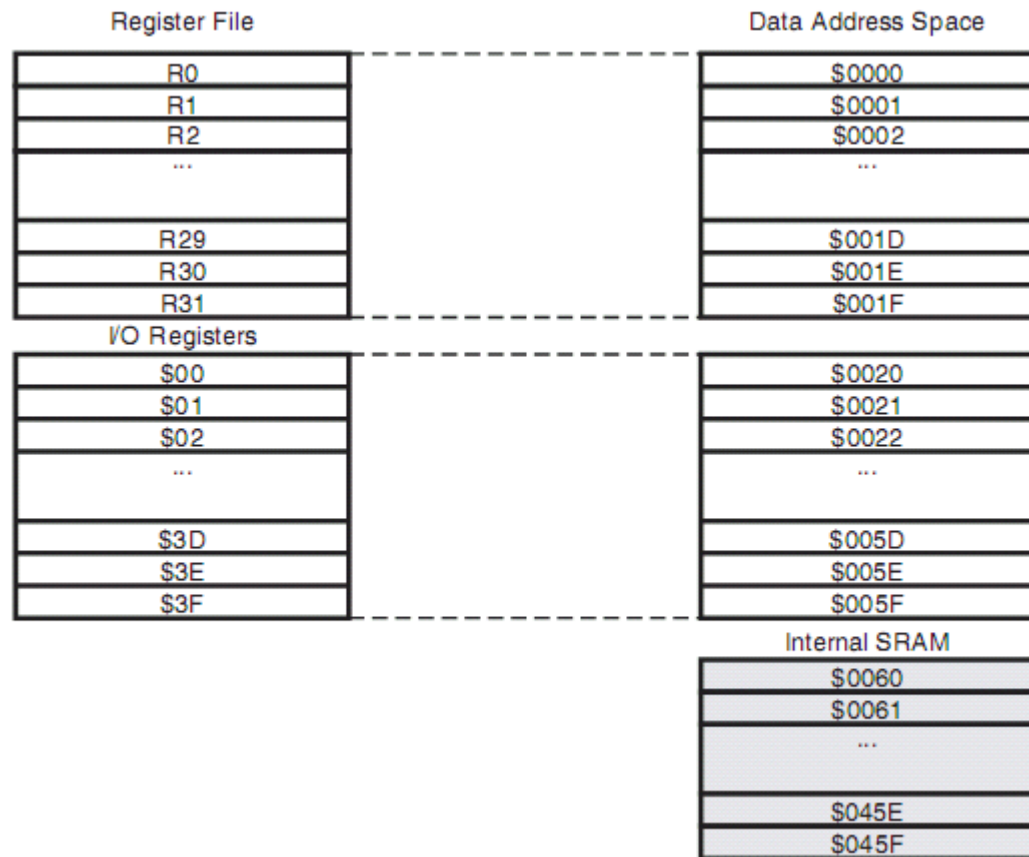
Bộ nhớ Flash 8KB của Atmega8 dùng để lưu trữ chương trình. Do các lệnh của AVR có độ dài 16 hoặc 32 bit nên bộ nhớ Flash được sắp xếp theo kiểu 4KX16. Bộ nhớ Flash được chia làm 2 phần, phần dành cho chương trình boot và phần dành cho chương trình ứng dụng.



Hình 2.4: Bộ nhớ chương trình

2.3.2. Bộ nhớ dữ liệu SRAM

1120 ô nhớ của bộ nhớ dữ liệu định địa chỉ cho file thanh ghi, bộ nhớ I/O và bộ nhớ dữ liệu SRAM nội. Trong đó 96 ô nhớ đầu tiên định địa chỉ cho file thanh ghi và bộ nhớ I/O, và 1024 ô nhớ tiếp theo định địa chỉ cho bộ nhớ SRAM nội.



Hình 2.5: Bộ nhớ dữ liệu

2.3.3. Bộ nhớ dữ liệu EEPROM

Atmega8 chứa bộ nhớ dữ liệu EEPROM dung lượng 512 byte, và được sắp xếp theo từng byte, cho phép các thao tác đọc/ghi từng byte một.

2.4. CÁC CÔNG VÀO RA (I/O)

Vi điều khiển ATmega8 có 23 đường vào ra chia làm bốn nhóm 8 bit một. Các đường vào ra này có rất nhiều tính năng và có thể lập trình được. Ở đây ta sẽ xét

chúng là các cổng vào ra số. Nếu xét trên mặt này thì các cổng vào ra này là cổng vào ra hai chiều có thể định hướng theo từng bit. Và chứa cả điện trở pull-up (có thể lập trình được). Mặc dù mỗi port có các đặc điểm riêng nhưng khi xét chúng là các cổng vào ra số thì dường như điều khiển vào ra dữ liệu thì hoàn toàn như nhau. Chúng ta có thanh ghi và một địa chỉ cổng đối với mỗi cổng, đó là : thanh ghi dữ liệu cổng (PORTA, PORTB, PORTC, PORTD), thanh ghi dữ liệu điều khiển cổng (DDRA, DDRB, DDRC, DDRD) và cuối cùng là địa chỉ chân vào của cổng (PINA, PINB, PINC, PIND).

2.4.1. Thanh ghi DDRx

Đây là thanh ghi 8 bit (ta có thể đọc và ghi các bit ở thanh ghi này) và có tác dụng điều khiển hướng cổng PORTx (tức là cổng ra hay cổng vào). Nếu như một bit trong thanh ghi này được set thì bit tương ứng đó trên PORTx được định nghĩa như một cổng ra. Ngược lại nếu như bit đó không được set thì bit tương ứng trên PORTx được định nghĩa là cổng vào.

2.4.2. Thanh ghi PORTx

Đây cũng là thanh ghi 8 bit (các bit có thể đọc và ghi được) nó là thanh ghi dữ liệu của cổng Px và trong trường hợp nếu cổng được định nghĩa là cổng ra thì khi ta ghi một bit lên thanh ghi này thì chân tương ứng trên port đó cũng có cùng mức logic. Trong trường hợp mà cổng được định nghĩa là cổng vào thì thanh ghi này lại mang dữ liệu điều khiển cổng. Cụ thể nếu bit nào đó của thanh ghi này được set (đưa lên mức 1) thì điện trở kéo lên (pull-up) của chân tương ứng của port đó sẽ được kích hoạt. Ngược lại nó sẽ ở trạng thái hi-Z. Thanh ghi này sau khi khởi động Vi điều khiển sẽ có giá trị là 0x00.

2.4.3. Thanh ghi PINx

Đây là thanh ghi 8 bit chứa dữ liệu vào của PORTx (trong trường hợp PORTx được thiết lập là cổng vào) và nó chỉ có thể đọc mà không thể ghi vào

được.

Để đọc dữ liệu từ ngoài thì ta phải thực hiện các bước sau:

- Đưa dữ liệu ra thanh ghi điều khiển DDR_{xn} để đặt cho PORT_x (hoặc bit n trong port) đó là đầu vào (xóa thanh ghi DDR_x hoặc bit).
- Sau đó kích hoạt điện trở pull-up bằng cách set thanh ghi PORT_x (bit).
- Cuối cùng đọc dữ liệu từ địa chỉ PIN_{xn} (trong đó x: là cổng và n là bit).

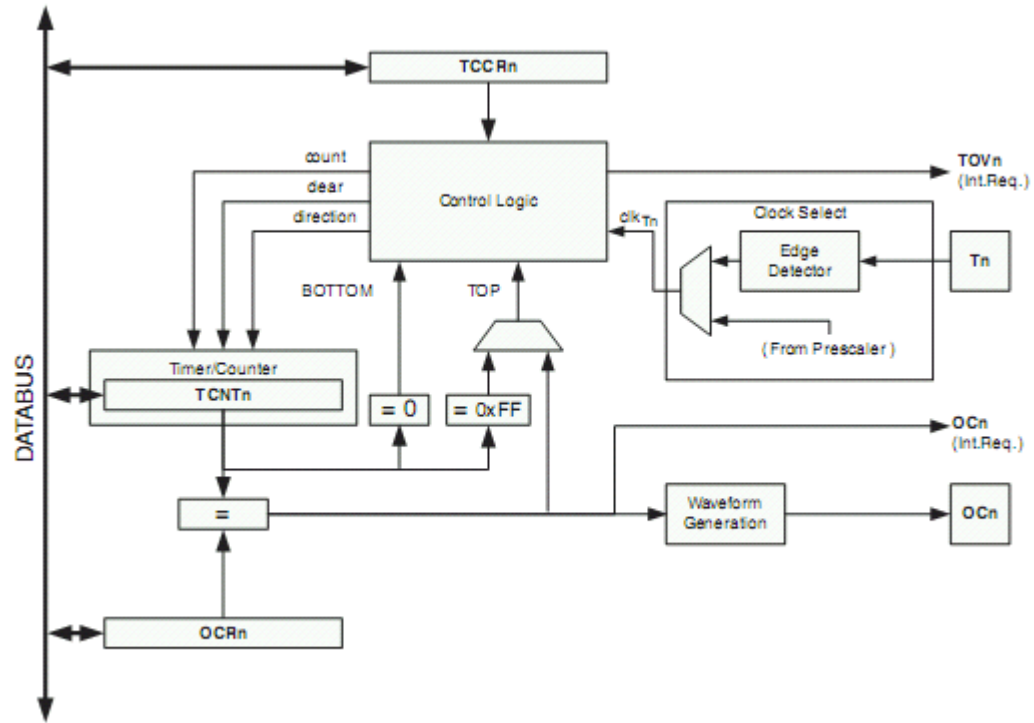
Để đưa dữ liệu từ vi điều khiển ra các cổng cũng có các bước hoàn toàn tương tự. Ban đầu ta cũng phải định nghĩa đó là cổng ra bằng cách set bit tương ứng của cổng đó...và sau đó là ghi dữ liệu ra bit tương ứng của thanh ghi PORT_x.

2.5. BỘ ĐỊNH THỜI

Bộ định thời (timer/counter0) là một module định thời/đếm 8 bit, có các đặc điểm sau:

- Bộ đếm một kênh
- Xóa bộ định thời khi trong mode so sánh (tự động nạp)
- PWM
- Tạo tần số
- Bộ đếm sự kiện ngoài
- Bộ chia tần 10 bit
- Nguồn ngắt tràn bộ đếm và so sánh

Sơ đồ cấu trúc của bộ định thời:



Hình 2.6: Sơ đồ cấu trúc bộ định thời

2.5.1. Các thanh ghi

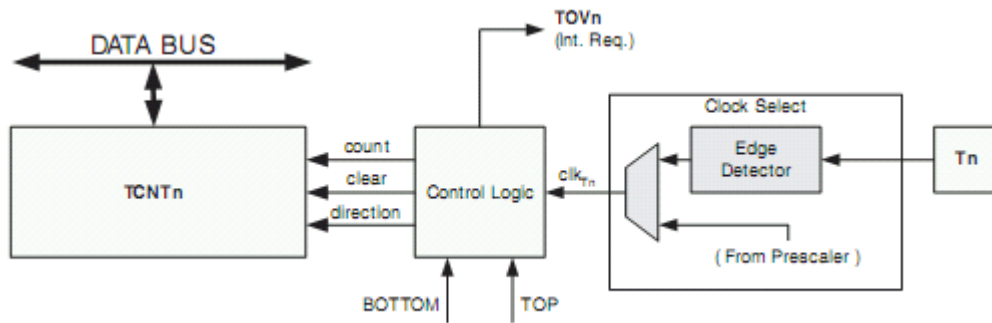
TCNT0 và OCR0 là các thanh ghi 8 bit. Các tín hiệu yêu cầu ngắt đều nằm trong thanh ghi TIFR. Các ngắt có thể được che bởi thanh ghi TIMSK.

Bộ định thời có thể sử dụng xung clock nội thông qua bộ chia hoặc xung clock ngoài trên chân T0. Khối chọn xung clock điều khiển việc bộ định thời bộ đếm sẽ dùng nguồn xung nào để tăng giá trị của nó. Ngõ ra của khối chọn xung clock được xem là xung clock của bộ định thời (clk_{T0}).

Thanh ghi OCR0 luôn được so sánh với giá trị của bộ định thời/bộ đếm. Kết quả so sánh có thể được sử dụng để tạo ra PWM hoặc biến đổi tần số ngõ ra tại chân OC0.

2.5.2. Đơn vị đếm

Phần chính của bộ định thời 8 bit là một đơn vị đếm song hướng có thể lập trình được. Cấu trúc của nó như hình dưới đây:



Hình 2.7: Cấu trúc bộ đếm

Count: tăng hay giảm TCNT0 1

Direction: lựa chọn giữa đếm lên và đếm xuống

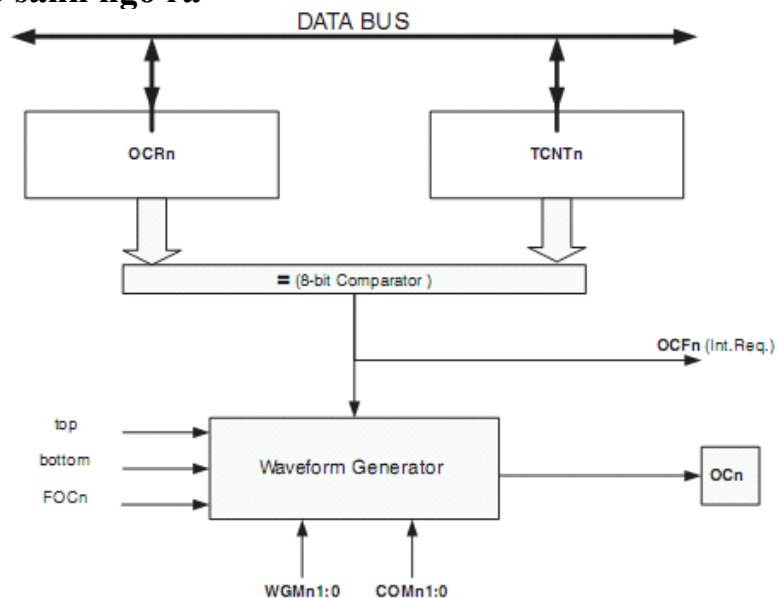
Clear: xóa thanh ghi TCNT0

Clk_{T0}: xung clock của bộ định thời

TOP: báo hiệu bộ định thời đã tăng đến giá trị lớn nhất

BOTTOM: báo hiệu bộ định thời đã giảm đến giá trị nhỏ nhất

2.5.3. Đơn vị so sánh ngõ ra



Hình 2.8: Sơ đồ đơn vị so sánh ngõ ra

Bộ so sánh 8 bit liên tục so sánh giá trị TCNT0 với giá trị trong thanh ghi so sánh ngõ ra (OCR0). Khi giá trị TCNT0 bằng với OCR0, bộ so sánh sẽ tạo một báo hiệu. Báo hiệu này sẽ đặt giá trị cờ so sánh ngõ ra (OCF0) lên 1 vào chu kỳ xung clock tiếp theo. Nếu được kích hoạt (OCIE0=1), cờ OCF0 sẽ tạo ra một ngắt so sánh ngõ ra và sẽ tự động được xóa khi ngắt được thực thi. Cờ OCF0 cũng có thể được xóa bằng phần mềm.

2.5.4. Mô tả các thanh ghi

2.5.4.1. Thanh ghi điều khiển bộ định thời/bộ đếm TCCR0

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Hình 2.11 Thanh ghi điều khiển bộ định thời

Bit 7-FOC0: So sánh ngõ ra bắt buộc

Bit này chỉ tích cực khi bit WGM00 chỉ định chế độ làm việc không có PWM. Khi đặt bit này lên 1, một báo hiệu so sánh bắt buộc xuất hiện tại đơn vị tạo dạng sóng.

Bit 6, 3-WGM01:0: Chế độ tạo dạng sóng

Các bit này điều khiển đếm thứ tự của bộ đếm, nguồn cho giá trị lớn nhất của bộ đếm (TOP) và kiểu tạo dạng sóng sẽ được sử dụng.

Bit 5:4-COM01:0: Chế độ báo hiệu so sánh ngõ ra

Các bit này điều khiển hoạt động của chân OC0. Nếu một hoặc cả hai bit COM01:0 được đặt lên 1, ngõ ra OC0 sẽ hoạt động.

Bit 2:0: CS02:0: Chọn xung đồng hồ

Ba bit này dùng để lựa chọn nguồn xung cho bộ định thời/bộ đếm

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$ /(No prescaling)
0	1	0	$clk_{I/O}/8$ (From prescaler)
0	1	1	$clk_{I/O}/64$ (From prescaler)
1	0	0	$clk_{I/O}/256$ (From prescaler)
1	0	1	$clk_{I/O}/1024$ (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

2.5.4.2. Thanh ghi bộ định thời/bộ đếm

Bit	7	6	5	4	3	2	1	0	
	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Thanh ghi bộ định thời/bộ đếm cho phép truy cập trực tiếp (cả đọc và ghi) vào bộ đếm 8 bit.

2.5.4.3. Thanh ghi so sánh ngõ ra-OCR0

Bit	7	6	5	4	3	2	1	0	
	OCR0[7:0]								OCR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Thanh ghi này chứa một giá trị 8 bit và liên tục được so sánh với giá trị của bộ đếm

2.5.4.4. Thanh ghi mặt nạ ngắt

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 1-OCIE0: Cho phép ngắt báo hiệu so sánh

Bit 0-TOIE0: Cho phép ngắt tràn bộ đếm

2.5.4.5. Thanh ghi cờ ngắt bộ định thời

Bit	7	6	5	4	3	2	1	0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 1-OCF0: Cờ so sánh ngõ ra 0

Bit 0-TOV0: Cờ tràn bộ đếm

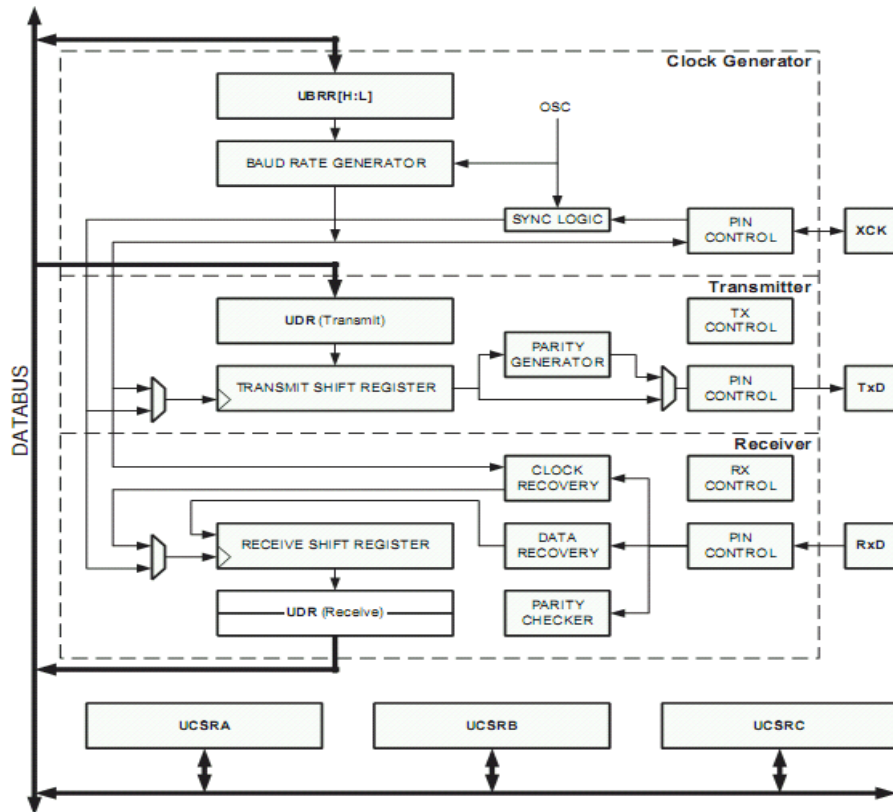
Bit TOV0 được đặt lên 1 khi bộ đếm bị tràn và được xóa bởi phần cứng khi vector ngắt tương ứng được thực hiện. Bit này cũng có thể được xóa bằng phần mềm.

2.6. USART

Bộ truyền nhận nối tiếp đồng bộ và bất đồng bộ là một thiết truyền thông nối tiếp có các chức năng chính như sau:

- Hoạt động song công (các thanh ghi truyền và nhận nối tiếp độc lập với nhau).
- Hoạt động đồng bộ hoặc bất đồng bộ
- Bộ tạo tốc độ baud có độ chính xác cao
- Hỗ trợ khung truyền nối tiếp với 5, 6, 7, 8, hoặc 9 bit dữ liệu và 1 hoặc 2 bit stop
- Kiểm tra chẵn lẻ
- Phát hiện tràn dữ liệu
- Phát hiện lỗi khung
- Lọc nhiễu, bao gồm phát hiện bit start lỗi và bộ lọc thông thấp số
- Ngắt khi kết thúc truyền, thanh ghi truyền hết dữ liệu và kết thúc nhận
- Chế độ truyền thông đa vi xử lý
- Chế độ truyền đồng bộ tốc độ cao

Sơ đồ khối của bộ USART như sau:

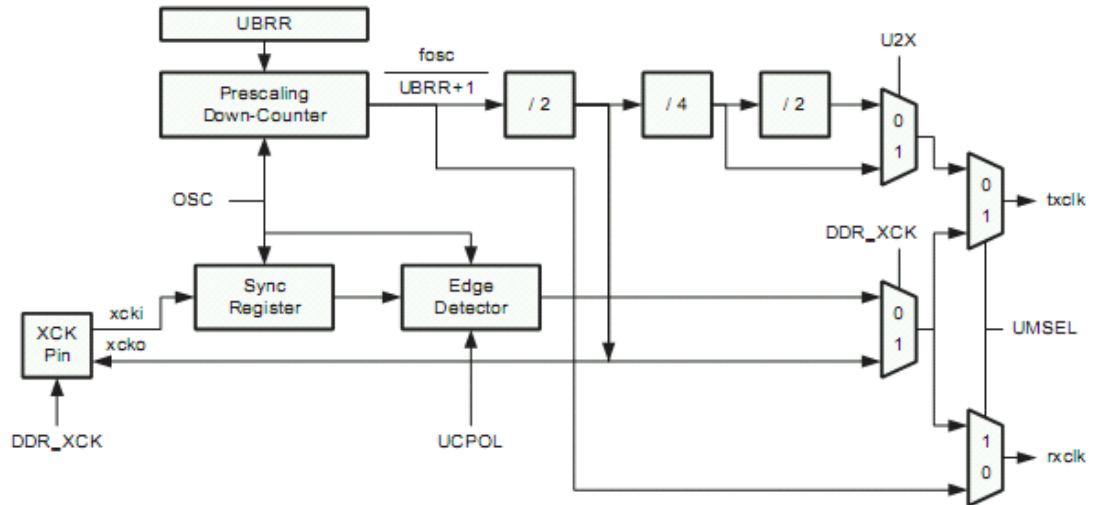


Hình 2.9: Sơ đồ khối của bộ USART

USART bao gồm 3 phần chính: bộ tạo xung clock, bộ truyền và bộ nhận. Các thanh ghi điều khiển được sử dụng chung giữa các phần này.

2.6.1. Tạo xung clock

Bộ tạo xung clock tạo ra xung đồng hồ căn bản cho bộ truyền và bộ nhận. USART hỗ trợ 4 chế độ hoạt động xung clock: bất đồng bộ, bất đồng bộ tốc độ cao, truyền đồng bộ master và truyền đồng bộ slave. Sơ đồ khối của bộ tạo xung clock như sau:



Hình 2.10: Đơn vị tạo xung clock

TXCLK: xung đồng hồ bộ truyền

RXCLK: xung đồng hồ bộ nhận

XCKI: tín hiệu vào từ chân XCK, sử dụng cho hoạt động truyền đồng bộ master

XCKO: tín hiệu xung clock ngõ ra tới chân XCK, sử dụng cho hoạt động truyền đồng bộ slave

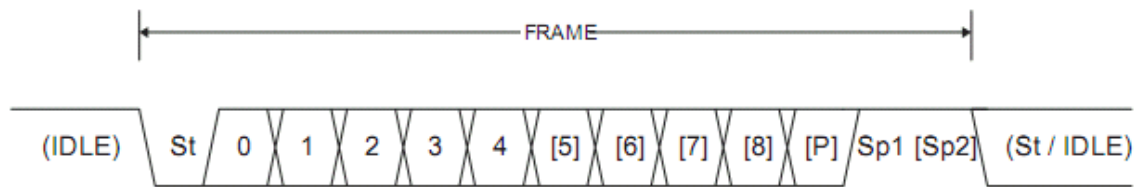
FOSC: tần số từ chân XTAL

2.6.2. Định dạng khung truyền

USART chấp nhận tất cả 30 tổ hợp của các định dạng khung truyền sau đây:

- 1 bit start
- 5, 6, 7, 8, hoặc 9 bit dữ liệu
- Có hoặc không có bit chẵn lẻ
- 1 hoặc 2 bit stop

Một khung truyền bắt đầu với một bit start, theo sau đó là bit có trọng số thấp nhất (LSB) của dữ liệu (có thể lên tới 9 bit), kết thúc bằng bit có trọng số lớn nhất (MSB) và bit stop.



Hình 2.18: Định dạng khung truyền

St: bit start (mức thấp)

(n): bit dữ liệu (0 đến 8)

P: bit chẵn lẻ

Sp: bit stop (mức cao)

IDLE: không có dữ liệu truyền (mức cao trong suốt thời gian idle)

2.6.3. Khởi tạo USART

Quá trình khởi tạo USART bao gồm việc thiết lập tốc độ baud, thiết lập định dạng khung và kích hoạt bộ truyền và bộ nhận.

2.6.4. Truyền thông dữ liệu-bộ truyền USART

Bộ truyền USART được kích hoạt bằng cách thiết lập bit TXEN trong thanh ghi UCSRB. Khi bộ truyền được kích hoạt, chân TxD hoạt động như ngõ ra của bộ truyền nối tiếp. Tốc độ baud, chế độ hoạt động và định dạng khung truyền phải được thiết lập trước khi thực hiện truyền dữ liệu.

2.6.4.1. Truyền khung 5 đến 8 bit dữ liệu

Việc truyền dữ liệu được thiết lập bằng cách nạp dữ liệu truyền vào bộ đệm truyền. Dữ liệu trong bộ đệm sẽ được đưa vào thanh ghi dịch khi thanh ghi dịch đã sẵn sàng gửi một khung mới.

2.6.4.2. Truyền khung 9 bit dữ liệu

Nếu sử dụng 9 bit dữ liệu, bit thứ 9 phải được ghi vào bit TXB8 trong thanh ghi UCSRB trước khi byte còn lại được ghi vào UDR.

2.6.5. Nhận dữ liệu-bộ nhận USART

Bộ nhận USART được kích hoạt bằng cách đặt bit RXEN trong thanh ghi UCSRB lên 1. Khi bộ nhận được kích hoạt, chân RxD hoạt động như ngõ vào của bộ nhận nối tiếp. Tốc độ baud, chế độ hoạt động và định dạng khung truyền phải được thiết lập trước khi thực hiện truyền dữ liệu.

2.6.5.1. Nhận khung với 5 đến 8 bit dữ liệu

Bộ nhận bắt đầu nhận dữ liệu khi nó phát hiện một bit start hợp lệ. Mỗi bit theo sau bit start sẽ được lấy mẫu tại tốc độ baud hoặc tốc độ đồng hồ XCK, và được dịch vào trong thanh ghi dịch của bộ nhận cho đến khi phát hiện một bit stop đầu tiên. Nội dung của thanh ghi dịch sau đó được đưa vào bộ đệm. Bộ đệm của bộ nhận có thể được đọc bằng cách đọc UDR.

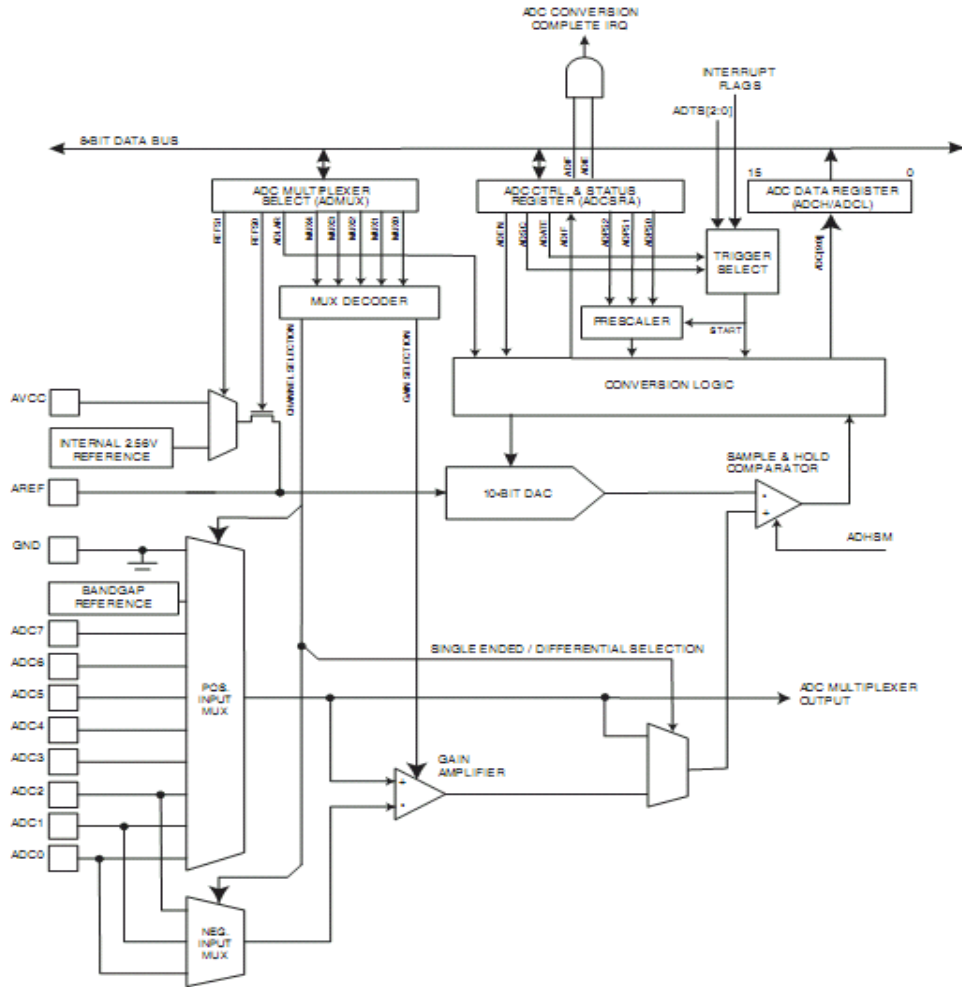
2.6.5.2. Nhận khung với 9 bit dữ liệu

Nếu nhận dữ liệu 9 bit, bit thứ 9 phải được đọc từ bit RXB8 trong thanh ghi UCSRB trước khi đọc các bit thấp trong UDR

2.7. BỘ BIẾN ĐỔI A/D

Vi điều khiển ATmega8 có một bộ biến đổi ADC tích hợp trong chip với các đặc điểm:

- Độ phân giải 10 bit
- Sai số tuyến tính: 0.5LSB
- Độ chính xác +/-2LSB
- Thời gian chuyển đổi: 65-260 μ s
- 8 Kênh đầu vào có thể được lựa chọn
- Có hai chế độ chuyển đổi free running và single conversion
- Có nguồn báo ngắt khi hoàn thành chuyển đổi
- Loại bỏ nhiễu trong chế độ ngủ



Hình 2.19: Sơ đồ bộ biến đổi A/D

Tám đầu vào của ADC là tám chân của PORTA và chúng được chọn thông qua một MUX.

Để điều khiển hoạt động vào ra dữ liệu của ADC và CPU chúng ta có 3 thanh ghi: ADMUX là thanh ghi điều khiển lựa chọn kênh đầu vào cho ADC, ADCSRA là thanh ghi điều khiển và thanh ghi trạng thái của ADC, ADCH và ADCL là 2 thanh ghi dữ liệu.

2.7.1. ADMUX: Multiplexer select register

Đây là thanh ghi điều khiển 8 bit.

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Với 4 bit được định nghĩa là MUX3, MUX2, MUX1, và MUX0, ứng với các tổ hợp logic ta có thể chọn kênh đầu vào. Cụ thể:

MUX4..0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
00000	ADC0	N/A		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			

Các bit REFS1 và REFS0 dùng để chọn giá trị điện áp tham khảo cho ADC, như sau:

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal Vref turned off
0	1	AVCC with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

Chú ý: Nếu như ta thay đổi kênh trong thời điểm mà ADC đang chuyển đổi thì khi quá trình chuyển đổi đã hoàn thành thì kênh vào mới được thay đổi.

2.7.2. ADCSR-ADC control and status register

Đây là thanh ghi điều khiển và lưu trạng thái của ADC.

Bit	7	6	5	4	3	2	1	0	
	ADCSRA								
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7-ADEN:ADC enable

Đây là bit điều khiển hoạt động của ADC. Khi bit này được set 1 thì ADC có thể hoạt động và ngược lại. Nếu như ta ngừng hoạt động của ADC trong khi nó đang chuyển đổi thì nó sẽ kết thúc quá trình chuyển đổi. Mặc dù chưa chuyển đổi xong.

Bit 6-ADSC: ADC start conversion

Trong chế độ chuyển đổi đơn thì bit này phải được set lên 1 để bắt đầu chuyển đổi. Trong chế độ chuyển đổi tự do thì bit này cần được set lên 1 để bắt đầu lần chuyển đổi đầu tiên. Bit này được giữ sót trong quá trình chuyển đổi và được xóa khi mà chuyển đổi xong.

Bit 5-ADATE :ADC Auto Trigger enable

Khi bit này được set thì ADC sẽ bắt đầu chuyển đổi mỗi khi có một nguồn kích hoạt xuất hiện. Việc lựa chọn nguồn kích hoạt được thực hiện bằng cách set các bit trong thanh ghi SFIOR.

Bit 4-ADIF: ADC interrupt Flag

Bit này được set lên 1 bởi phần cứng khi quá trình chuyển đổi đã hoàn thành và thanh ghi dữ liệu đã được cập nhật. Bit này được xóa bằng phần cứng nếu như ngắt này được phép và được phục vụ. Hoặc nó có thể được xóa bằng cách ghi giá trị logic “0” vào cờ này. Cụ thể khi ngắt bị cấm ta có thể sử dụng các lệnh sbi và cbi để tác dụng lên bit này

Bit 3-ADIE:ACD interrupt Enable

Nếu bit này set 1 và ngắt toàn cục được cho phép thì ngắt này được phép phục vụ (khi chuyển đổi xong dữ liệu) và nếu bị xóa thì ngược lại.

Bit 2.1.0-ADPS2...ADPS0: Bit lựa chọn xung nhịp(Tốc độ)

Nguồn xung được lấy từ nguồn xung của Vi điều khiển(XTAL) và được chia tần thông qua bộ chia tần.

Các bit ADPS có nhiệm vụ chọn số chia cho bộ chia tần theo bảng sau:

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

2.7.3. Thanh ghi dữ liệu ACDH và ADCL

Thanh ghi này chứa dữ liệu chuyển đổi từ tương tự sang số, được sắp xếp như hình dưới đây.

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

2.7.4. Nguyên tắc hoạt động và lập trình điều khiển

ADC có nhiệm vụ chuyển đổi tín hiệu điện áp tương tự thành tín hiệu số có độ phân giải 10 bit. Với giá trị nhỏ nhất của điện áp đặt ở chân AGND và giá trị cực đại của điện áp tương tự được mắc vào chân AREF. Tám kênh tương tự đầu vào được chọn lựa thông qua ADMUX và ADMUX này được điều khiển bởi thanh ghi ADMUX.

ADC này có thể hoạt động được ở hai chế độ. Đó là chuyển đổi đơn: chỉ chuyển đổi một lần khi có lệnh chuyển đổi và chế độ tự chuyển đổi (Free running mode) đây là chế độ mà ADC tự động chuyển đổi khi được hoạt động và công việc chuyển đổi có tính tuần hoàn (chỉ cần khởi động một lần).

ADC được phép hoạt động nhờ thiết lập bit ADEN. Quá trình chuyển đổi được bắt đầu bằng việc ghi vào bit ADSC mức logic 1 và trong suốt quá trình chuyển đổi bit này luôn được giữ ở mức cao. Khi quá trình chuyển đổi hoàn thành thì bit này được xóa bằng phần cứng và cờ AIDF được bật lên.

Dữ liệu sau khi chuyển đổi được đưa ra thanh ghi dữ liệu ADCL và ADCH, nhưng chú ý khi đọc dữ liệu từ hai thanh ghi này thì đọc ADCL trước rồi mới đọc ADCH. Nếu đọc ADCH trước thì dữ liệu cập nhật có thể ghi đè lên ADCL (Vi điều khiển nghĩ rằng đã đọc xong dữ liệu).

Để điều khiển vào ra dữ liệu với ADC, các bước thực hiện như sau:

Bước 1: Định nghĩa các cổng vào cho tín hiệu tương tự

Xóa bit tương ứng với chân đó trong thanh ghi DDRA. Sau đó loại bỏ điện trở treo bằng cách xóa bit tương ứng ở thanh ghi PORTA.

Bước 2: Chọn kênh tương tự vào (chọn chân vào cho ADC) thông qua thanh ghi ADMUX (có thể thay đổi trong quá trình hoạt động).

Bước 3: Thiết lập các thông số cho ADC

Tốc độ chuyển đổi thông qua xung nhịp chuyển đổi

Chế độ chuyển đổi: đơn hoặc tự động

Sử dụng ngắt hoặc không

Bước 4: Bắt đầu chuyển đổi và đọc dữ liệu.

2.8. GIAO TIẾP VI ĐIỀU KHIỂN VỚI MÁY TÍNH

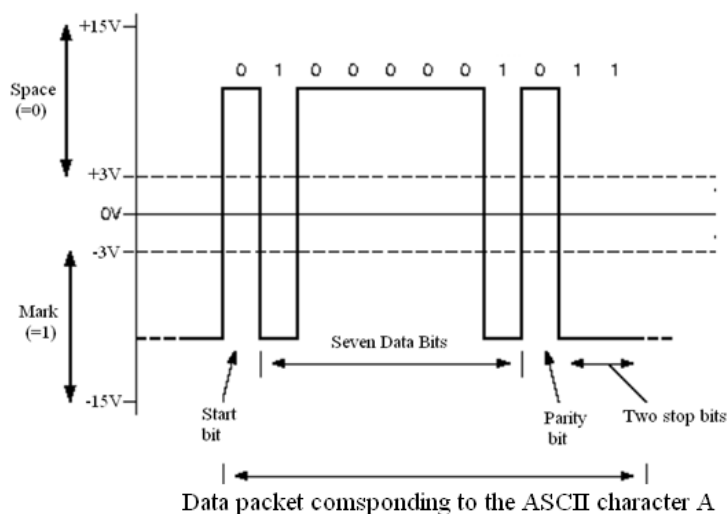
2.8.1. Chuẩn RS232

Chuẩn RS-232 từ năm 1969 được chấp nhận chuyên dùng cho truyền số liệu và các đường kiểm tra giữa terminal và modem, tốc độ cực đại là 20Kbps, với khoảng cách tối đa không quá 15m. Chuẩn RS-232 quy định mức logic 1 ứng với điện áp từ -3V đến -15V(mark), mức logic 0 ứng với điện áp từ 3V đến 15V(space) và có khả năng cung cấp dòng từ 10 mA đến 20 mA. Ngoài ra, tất cả các ngõ ra đều có đặc tính chống chập mạch.

Định dạng của khung truyền dữ liệu theo chuẩn RS-232 như sau:

Star	D0	D1	D2	D3	D4	D5	D6	D7	P	Stop
------	----	----	----	----	----	----	----	----	---	------

Khi không truyền dữ liệu, đường truyền sẽ ở trạng thái mark (điện áp -10V). Khi bắt đầu truyền, DTE (Data Terminal Equipment) sẽ đưa ra xung Start (space: 10V) và sau đó lần lượt truyền từ D0 đến D7 và Parity, cuối cùng là xung Stop (mark: -10V) để khôi phục trạng thái đường truyền. Dạng tín hiệu truyền mô tả như sau (truyền ký tự A):



Hình 2.20: Truyền ký tự A theo chuẩn RS232

Bảng 2.1. Các đặc tính kỹ thuật của chuẩn RS-232 như sau:

Chiều dài cable cực đại	15m
Tốc độ dữ liệu cực đại	20 Kbps
Điện áp ngõ ra cực đại	$\pm 25V$
Điện áp ngõ ra có tải	$\pm 5V$ đến $\pm 15V$
Trở kháng tải	3K đến 7K
Điện áp ngõ vào	$\pm 15V$
Độ nhạy ngõ vào	$\pm 3V$
Trở kháng ngõ ra	3K đến 7K

Các tốc độ truyền dữ liệu thông dụng trong cổng nối tiếp là: 1200 bps, 4800 bps, 9600 bps và 19200 bps.

2.8.2. Cổng COM

Cổng COM còn gọi là cổng nối tiếp theo chuẩn RS-232. Cổng COM được sử dụng khá phổ biến dữ liệu ở cổng COM thuộc dữ liệu dạng nối tiếp.

Cổng nối tiếp được sử dụng để truyền dữ liệu hai chiều giữa máy tính và ngoại vi, có các ưu điểm sau:

- Khoảng cách truyền xa hơn truyền song song.
- Số dây kết nối ít.
- Có thể truyền không dây dùng hồng ngoại.
- Có thể ghép nối với vi điều khiển hay PLC (Programmable Logic Device).
- Cho phép nối mạng.
- Có thể tháo lắp thiết bị trong lúc máy tính đang làm việc.
- Có thể cung cấp nguồn cho các mạch điện đơn giản

Các thiết bị ghép nối chia thành 2 loại: DTE (Data Terminal Equipment) và DCE (Data Communication Equipment). DCE là các thiết bị trung gian như MODEM còn DTE là các thiết bị tiếp nhận hay truyền dữ liệu như máy tính,

PLC, vi điều khiển, ... Việc trao đổi tín hiệu thông thường qua 2 chân RxD (nhận) và TxD (truyền). Các tín hiệu còn lại có chức năng hỗ trợ để thiết lập và điều khiển quá trình truyền, được gọi là các tín hiệu bắt tay (handshake). Ưu điểm của quá trình truyền dùng tín hiệu bắt tay là có thể kiểm soát đường truyền. Tín hiệu truyền theo chuẩn RS-232 của EIA (Electronics Industry Associations).

Các phương thức nối giữa DTE và DCE:

- Đơn công (simplex connection): dữ liệu chỉ được truyền theo 1 hướng.
- Bán song công (half-duplex): dữ liệu truyền theo 2 hướng, nhưng mỗi thời điểm
- Chỉ được truyền theo 1 hướng.
- Song công (full-duplex): số liệu được truyền đồng thời theo 2 hướng.
- Cổng COM có dạng đầu nối DB9 (9 chân).



Hình 2.21: Cổng DB9 trên máy tính

Bảng 2.2. Đầu nối DB9 mô tả như sau :

D9	Tín hiệu	Hướng truyền	Mô tả
1	DCD	DCE → DTE	Data carrier detect: DCE phát hiện sóng mang
2	RxD	DCE → DTE	Received data: dữ liệu nhận
3	TxD	DTE → DCE	Transmitted data: dữ liệu truyền
4	DTR	DTE → DCE	Data terminal ready: DTE sẵn sàng làm việc
5	GND	-	Ground: nối đất (0V)

6	DSR	DCE → DTE	Data set ready: DCE sẵn sàng làm việc
7	RTS	DTE → DCE	Request to send: DTE yêu cầu truyền dữ liệu
8	CTS	DCE → DTE	Clear to send: DCE sẵn sàng nhận dữ liệu
9	RI	RI DCE → DTE	Ring indicator: báo chuông

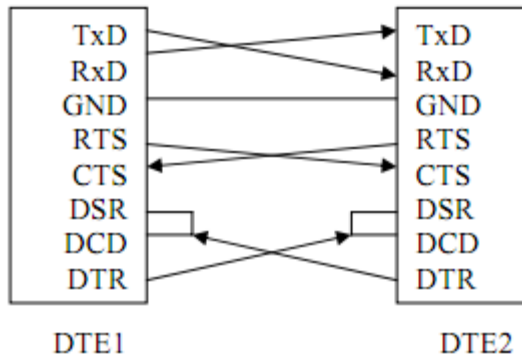
2.8.3. Truyền thông giữa hai nút

Các sơ đồ khi kết nối dùng cổng nối tiếp:



Hình 2.22: Kết nối đơn giản trong truyền thông nối tiếp

Khi thực hiện kết nối như trên, quá trình truyền phải bảo đảm tốc độ ở đầu phát và thu giống nhau. Khi có dữ liệu đến DTE, dữ liệu này sẽ được đưa vào bộ đệm và tạo ngắt. Ngoài ra, khi thực hiện kết nối giữa hai DTE, ta còn dùng sơ đồ sau



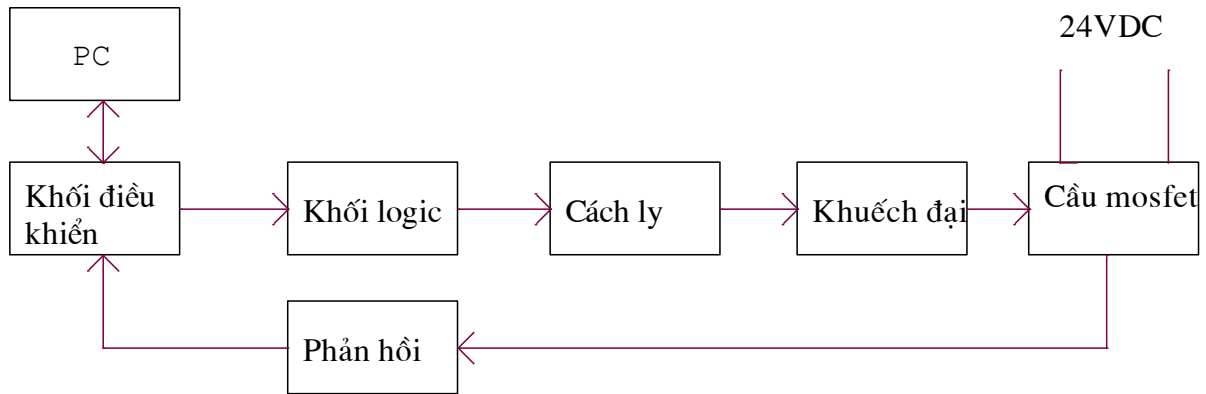
Hình 2.23: Kết nối truyền thông nối tiếp dùng tín hiệu bắt tay

Khi DTE1 cần truyền dữ liệu thì cho DTR tích cực, tác động lên DSR của DTE2 cho biết sẵn sàng nhận dữ liệu và cho biết đã nhận được sóng mang của MODEM (ảo). Sau đó, DTE1 tích cực chân RTS để tác động đến chân CTS của DTE2 cho biết DTE1 có thể nhận dữ liệu. Khi thực hiện kết nối giữa DTE và DCE, do tốc độ truyền khác nhau nên phải thực hiện điều khiển lưu lượng. Quá trình điều khiển này có thể thực hiện bằng phần mềm hay phần cứng. Quá trình điều khiển bằng phần mềm thực hiện bằng hai ký tự Xon và Xoff. Ký tự Xon được DCE gửi đi khi rảnh (có thể nhận dữ liệu). Nếu DCE bận thì sẽ gửi ký tự Xoff. Quá trình điều khiển bằng phần cứng dùng hai chân RTS và CTS. Nếu DTE muốn truyền dữ liệu thì sẽ gửi RTS để yêu cầu truyền, DCE nếu có khả năng nhận dữ liệu (đang rảnh) thì gửi lại CTS.

Chương 2 em đã giới thiệu chung về vi điều khiển atmega8 nó là tiền đề để làm chương tiếp theo.

CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG MẠCH ĐIỀU KHIỂN

3.1. SƠ ĐỒ KHỐI ĐIỀU KHIỂN CHUNG



Hình 3.1: Sơ đồ khối điều khiển

Khối điều khiển

Ta sử dụng vi điều khiển atmega8 đây là vi điều khiển mới của atmel có tích hợp sẵn kênh PWM chuyên dụng.

Giao tiếp với máy tính thông qua UART. Do máy tính của em không có cổng COM cho nên ta cần một mạch chuyển đổi từ cổng USB sang cổng COM và đầu ra được nối trực tiếp với atmega8.

Khối logic

Thực hiện điều khiển logic

Mạch logic có nhiệm vụ điều khiển mạch công suất. Mạch có chức năng phân kênh để đảo chiều động cơ DC servo

Khối cách ly

Mục đích là cách ly giữa phần điều khiển và phần công suất. Và được thực hiện bởi PC817

Khối khuếch đại

Khuếch đại điện áp để bão hòa mở mạch cầu cho Mosfet

Cầu Mosfet

Thực hiện cấp nguồn cho động cơ giúp động cơ đảo chiều dễ dàng tần số đảo chiều cao và tổn thất năng lượng ít.

Khởi phản hồi

Dùng encoder có độ phân giải 100 xung trên 1 vòng để phản hồi tốc độ động cơ.

Khởi nguồn

Có nhiệm vụ cung cấp nguồn 24VDC cho mạch hoạt động

3.2. MỘT SỐ LINH KIỆN ĐƯỢC DÙNG TRONG MẠCH

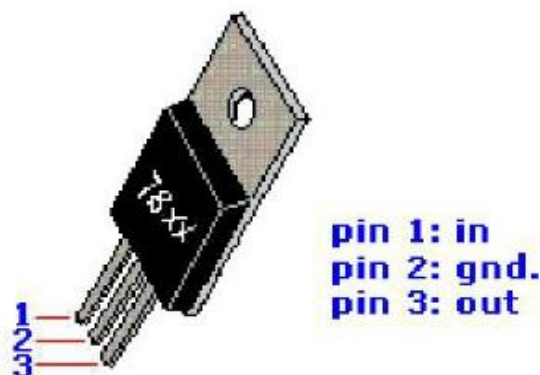
3.2.1. Ic nguồn

Linh kiện ổn áp là thành phần không thể thiếu trong các mạch ổn áp và mạch nguồn của kỹ thuật điện tử

Họ 78xx là họ cho ổn định điện áp đầu ra dương còn xx là giá trị điện áp đầu ra như +5V, +8V, +12V...

Trong các bộ nguồn thì họ 78 được sử dụng rất nhiều để tạo điện áp ra mong muốn đặc biệt những thiết bị này cần điện áp đầu vào ổn định không thay đổi lên xuống.

IC78xx gồm có 3 chân:



Hình 3.2: Sơ đồ chân họ 78xx

Chân 1.Vin là chân nguồn vào

Chân 2.GND là chân nối đất

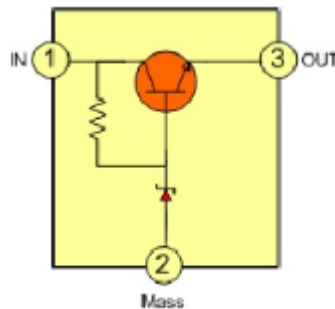
Chân 3.Vout là chân nguồn ra

Ở đây dùng :

IC7805 ổn áp 5V

IC7812 ổn áp 12V

Cấu tạo bên trong của 2 IC này:



Hình 3.3: Cấu trúc bên trong họ 78xx

Cấu tạo bên trong của họ 78xx gồm có một điện trở kết hợp với diode zener dùng để ghim mức điện áp và một transistor để khuếch đại dòng.

Họ 78xx chỉ cho dòng tiêu thụ khoảng 1A trở xuống khi lắp IC trong mạch thì Vin phải lớn hơn Vout từ 3V đến 5V thì mạch mới phát huy tác dụng

Công suất tiêu tán cực đại nếu không dùng tản nhiệt là 2W

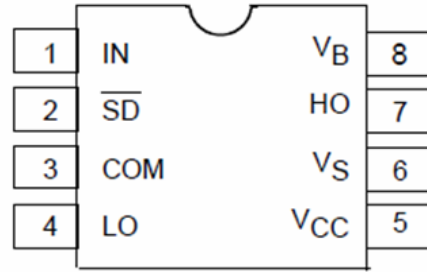
Công suất tiêu tán cực đại nếu dùng tản nhiệt là 15W

IC 78xx có bảo vệ chập tải nếu làm chập mạch chưa chắc 78xx đã hỏng

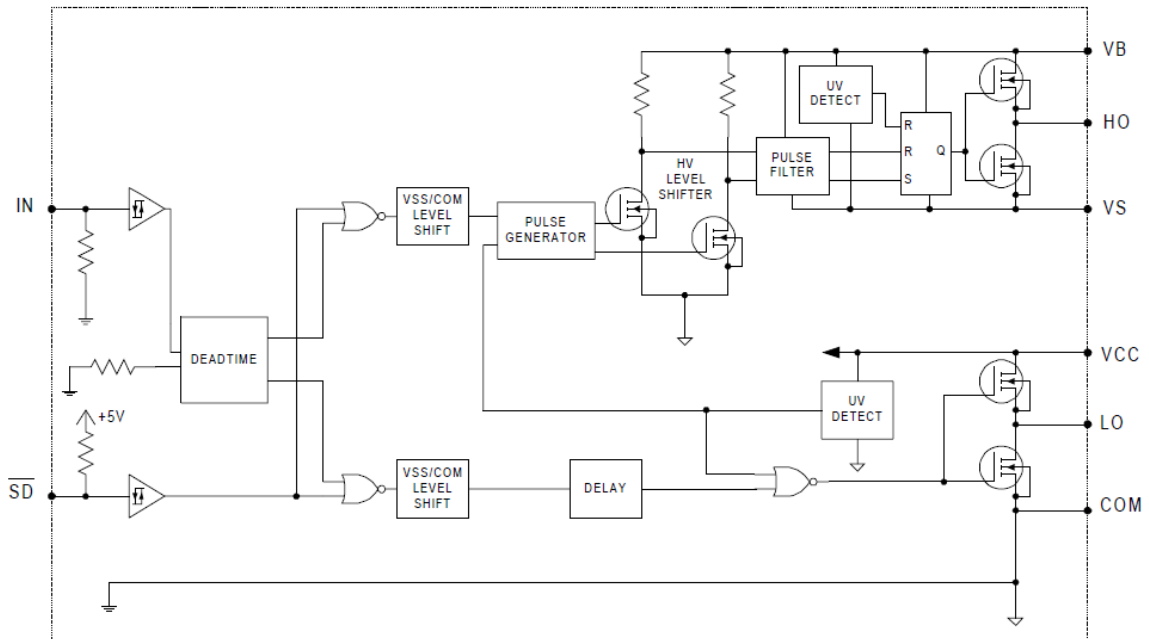
Mục đích sử dụng IC ổn áp 7805 trong đề tài này là hạ áp từ 24V xuống 5V để phục vụ cấp nguồn cho vi điều khiển atmega8 và opto PC817

Còn IC ổn áp 7812 là hạ áp từ 24V xuống 12V để phục vụ cấp nguồn cho IR2184.

3.2.2. IR2184



Hình 3.4: Sơ đồ chân của IC IR2184



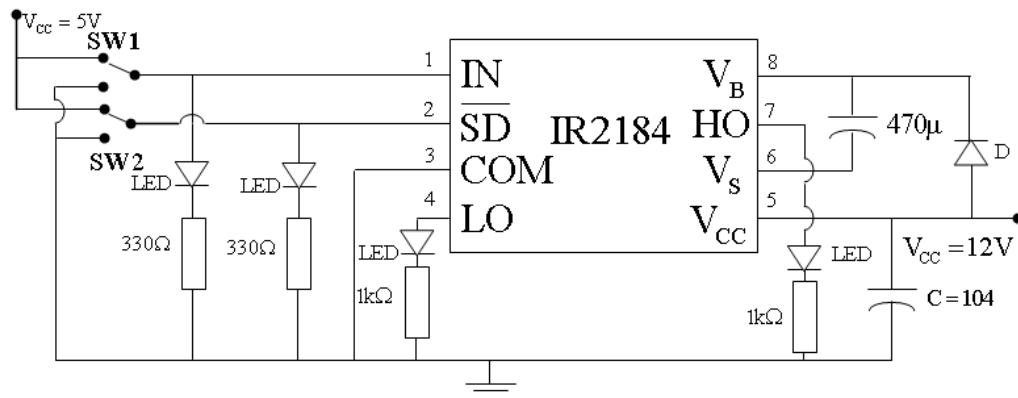
Hình 3.5 : Sơ đồ khối cấu tạo bên trong của IC IR2184

Bảng 3.1. Bảng giá trị về IC IR2184.

Đại lượng	Ý nghĩa	Điện thế nhỏ nhất	Điện thế lớn nhất	Đơn vị
V_B	Nguồn dự trữ thay đổi phía áp cao.	-0.3	625	V V
V_S	Phục hồi nguồn dự trữ thay đổi phía áp cao.	$V_B - 25$	$V_B + 0.3$	

V_{HO}	Điện thế ở ngõ ra điều khiển mức cao.	$V_S - 0.3$	$V_B + 0.3$	
V_{CC}	Nguồn nuôi (nguồn cố định mức logic).	-0.3	25	
V_{LO}	Điện thế ở ngõ ra điều khiển mức cao.	-0.3	$V_{CC} + 0.3$	
V_{IN}	Điện thế ngõ vào logic ở chân IN và chân .	$V_{SS} - 0.3$	$V_{SS} + 10$	

Dùng nguồn 12V cấp vào chân Vcc và nguồn 5V cấp cho chân IN



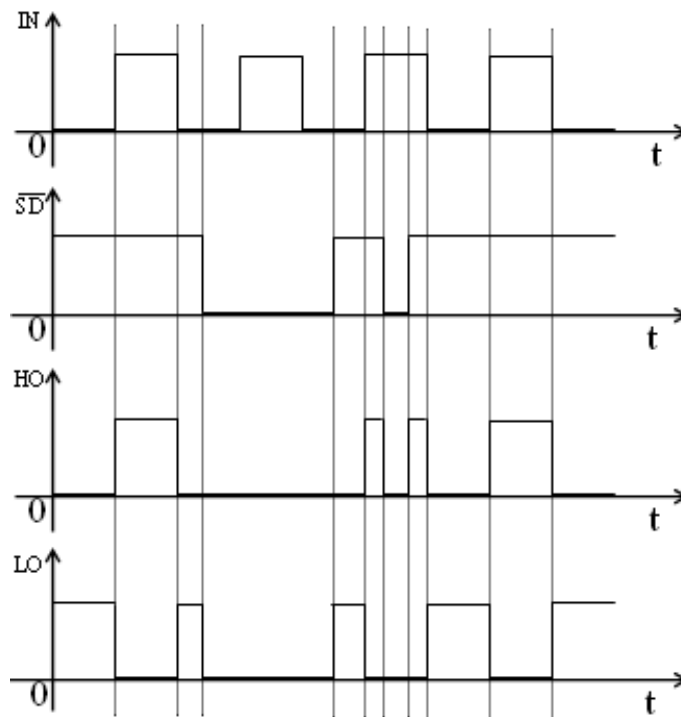
Hình 3.6: Sơ đồ mạch khảo sát IC IR2184.

Khảo sát mức logic ở các chân IN, , HO và LO ta được kết quả

Bảng 3.2. Bảng kết quả khảo sát mức logic.

5V		12V	
IN		HO	LO
1	1	1	0
0	1	0	1
1	0	0	0
0	0	0	0

Từ kết quả khảo sát ta giản đồ tín hiệu quan hệ giữa ngõ vào và ngõ ra của IC IR2184



Hình 3.7: Giản đồ tín hiệu quan hệ giữa ngõ vào và ngõ ra của IC IR2184

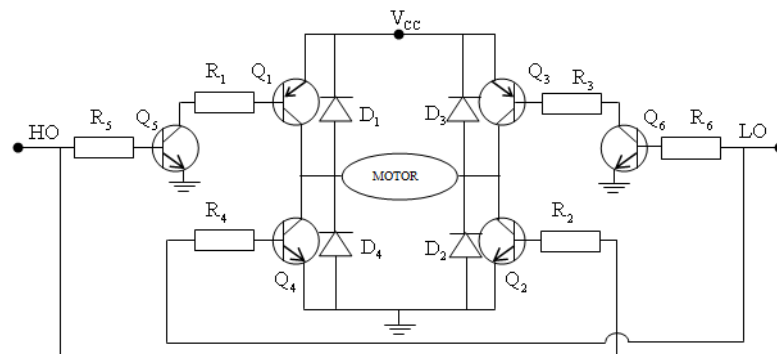
Nhận xét:

- Ngõ ra HO ở mức 1 và LO ở mức 0 khi cả hai ngõ vào IN và đều ở mức 1.
- Ngõ ra LO ở mức 1 và HO ở mức 0 khi ngõ vào IN ở mức 0 và ở mức 1.

Qua kết quả khảo sát quan hệ giữa ngõ vào và ngõ ra của IC IR2184 ta rút ra được kết luận để điều khiển motor DC như sau:

- Chỉ cần cố định chân =1 và điều khiển chân IN=0 hay IN=1 thì ta có thể điều khiển đảo chiều động cơ DC.
- Có thể dùng mạch tạo xung điều khiển ngõ vào của IC IR2184.

Ngõ ra HO và LO của IC IR2184 ta đưa qua cầu H để điều khiển motor.



Hình 3.8: Cầu H

3.2.3. PC817

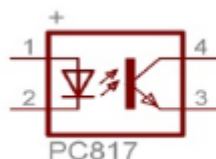
Nhiệm vụ của PC817 là cách ly hai tín hiệu

Tín hiệu điều khiển làm việc ở điện áp và dòng điện nhỏ nhưng rất ổn định

Tín hiệu của mạch động lực ở điện áp và dòng điện lớn nhưng không có tính ổn định

Khối điều khiển và khối công suất được cách ly với nhau bởi phân tử cách ly quang sử dụng PC817

Đây là một loại thiết bị hiện đại cho bộ quang điện tử



Hình 3.9: Sơ đồ chân và cấu tạo của PC817

Trong đó:

- 1: Anode
- 2: Kathode
- 3: Emitter
- 4: Colector

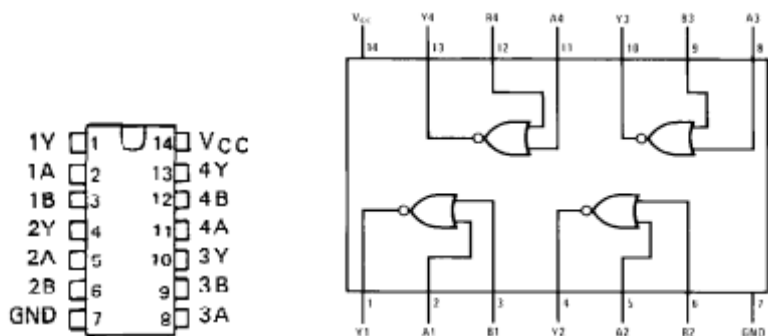
Bảng 3.3. Các thông số cơ bản của PC817

	Tham số	Ký hiệu	Mức	Đơn vị
Đầu vào	Dòng thuận	I_F	50	mA
	Dòng thuận cực đại	I_{FM}	1	A
	Điện áp ngược	V_R	60	V
	Công suất tiêu tán	P	70	mW
Đầu ra	Điện áp C-E	V_{CEO}	35	V
	Điện áp E-C	V_{ECO}	6	V
	Dòng điện cực C	I_C	50	mA
	Công suất tiêu tán cực C	P_C	150	mW
Tổng công suất tiêu thụ		P_{tot}	200	mW
Điện áp cách ly		V_{iso}	5	kVrms
Nhiệt độ làm việc		T_{opt}	-30:100	$^{\circ}C$

Khi có dòng điện nhỏ chạy qua chân 1 và chân 2 thì làm cho led phát sáng lúc này làm thông hai cực của photodiode bên trong mở cho dòng điện chạy qua.

3.2.4. IC SN7402N

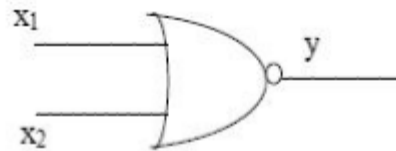
Sơ đồ chân IC SN7402N



Hình 3.10: Sơ đồ chân và cấu trúc bên trong

IC SN7402N gồm có 4 cổng NOR thực hiện phép cộng đảo logic các biến đầu vào

Ký hiệu



Phương trình logic

$$y = \overline{x_1 + x_2}$$

Bảng trạng thái

X ₁	X ₂	Y
0	0	1
0	1	0
1	0	0
1	1	0

3.2.5. Màn hình hiển thị LCD16x2



Hình 3.11: Màn hình LCD 16x2

Bảng 3.4. Chức năng của chân LCD

Chân số	Ký hiệu	Mức logic	I/O	Chức năng
1	V _{ss}			GND
2	V _{dd}			VCC 5V
3	V _{ee}		I	Chỉnh độ tương phản LCD
4	RS	0/1	I	Lựa chọn thanh ghi 0= thanh ghi lệnh 1=thanh ghi dữ liệu
5	R/W	0/1	I	0= ghi vào LCD 1= đọc từ LCD
6	E	1,1=>0	I	Tín hiệu cho phép
7	DB1	0/1	I/O	Data busline 0
8	DB2	0/1	I/O	Data busline 1
9	DB3	0/1	I/O	Data busline 2
10	DB4	0/1	I/O	Data busline 3
11	DB5	0/1	I/O	Data busline 4
12	DB6	0/1	I/O	Data busline 5
13	DB7	0/1	I/O	Data busline 6
14	DB8	0/1	I/O	Data busline 7(MSB)
15	V _{cc}			Nguồn cung cấp

16	GND			Nguồn cung cấp
----	-----	--	--	----------------

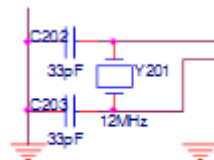
Một chương trình hiển thị ký tự trên LCD sẽ đi theo bốn bước sau:

- Xóa toàn bộ màn hình.
- Đặt chế độ hiển thị.
- Đặt vị trí con trỏ (nơi bắt đầu của ký tự hiển thị).
- Hiển thị ký tự.

Các bước 3, 4 có thể lặp lại nhiều lần nếu cần hiển thị nhiều ký tự.

Mỗi khi thực hiện ghi lệnh hoặc ghi dữ liệu hiển thị lên LCD cần phải kiểm tra cờ bận trước. Vì vậy, cần phải chủ động phân phối thời gian khi ra lệnh cho LCD(ví dụ sau khi xóa màn hình sau khoảng 2ms mới ra lệnh khác vì thời gian để LCD xóa màn hình là 1,64ms). Chế độ hiển thị mặc định sẽ là hiển thị dịch, vị trí con trỏ mặc định sẽ là đầu dòng thứ nhất.

3.2.6. Thạch anh

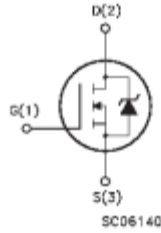


Hình 3.12: Sơ đồ chân của thạch anh

Ở đây ta dùng thạch anh có tần số 12.000MHZ để tạo bộ dao động

3.2.7. Mạch cầu H dùng MOSFET

MOSFET là viết tắt của cụm Meta Oxide Semiconductor Field-Effect Transistor tức Transistor hiệu ứng trường có dùng kim loại và oxit bán dẫn. Hình dưới mô tả cấu tạo của MOSFET kênh n

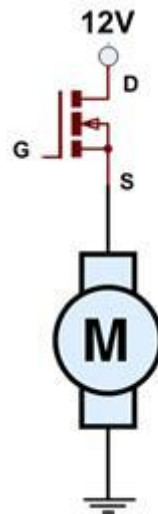


Hình 3.13: Sơ đồ chân MOSFET

MOSFET có 3 chân gọi là Gate (G), Drain (D) và Source (S) tương ứng với B, E và C của BJT. Đối với MOSFET kênh N, nếu điện áp chân G lớn hơn chân S khoảng từ 3V thì MOSFET bão hòa hay dẫn. Khi đó điện trở giữa 2 chân D và S rất nhỏ (gọi là điện trở dẫn DS), MOSFET tương đương với một khóa đóng. Ngược lại, với MOSFET kênh P, khi điện áp chân G nhỏ hơn điện áp chân S khoảng 3V thì MOSFET dẫn, điện trở dẫn cũng rất nhỏ. Vì tính dẫn của MOSFET phụ thuộc vào điện áp chân G (khác với BJT, tính dẫn phụ thuộc vào dòng IB), MOSFET được gọi là linh kiện điều khiển bằng điện áp, rất lý tưởng cho các mạch số nơi mà điện áp được dùng làm mức logic (ví dụ 0V là mức 0, 5V là mức 1).

MOSFET thường được dùng thay các BJT trong các mạch cầu H vì dòng mà linh kiện bán dẫn này có thể dẫn rất cao, thích hợp cho các mạch công suất lớn. Do cách thức hoạt động, có thể hình dung MOSFET kênh N tương đương một BJT loại npn và MOSFET kênh P tương đương BJT loại pnp. Thông thường các nhà sản xuất MOSFET thường tạo ra 1 cặp MOSFET gồm 1 linh kiện kênh N và 1 linh kiện kênh P, 2 MOSFET này có thông số tương đồng nhau và thường được dùng cùng nhau. Một ví dụ dùng 2 MOSFET tương đồng là các mạch số CMOS (Complementary MOS). Cũng giống như BJT, khi dùng MOSFET cho mạch cầu H, mỗi loại MOSFET chỉ thích hợp với 1 vị trí nhất định, MOSFET

kênh N được dùng cho các khóa phía dưới và MOSFET kênh P dùng cho các khóa phía trên.



Hình 3.14: Dùng MOSFET kênh N điều khiển motor DC

Ban đầu MOSFET không được kích, không có dòng điện trong mạch, điện áp chân S bằng 0. Khi MOSFET được kích và dẫn, điện trở dẫn DS rất nhỏ so với trở kháng của motor nên điện áp chân S gần bằng điện áp nguồn là 12V. Do yêu cầu của MOSFET, để kích dẫn MOSFET thì điện áp kích chân G phải lớn hơn chân S ít nhất 3V, nghĩa là ít nhất 15V trong khi chúng ta dùng vi điều khiển để kích MOSFET, rất khó tạo ra điện áp 15V. Như thế MOSFET kênh N không phù hợp để làm các khóa phía trên trong mạch cầu H. MOSFET loại P thường được dùng trong trường hợp này. Tuy nhiên, một nhược điểm của MOSFET kênh P là điện trở dẫn DS của nó lớn hơn MOSFET loại N. Vì thế, dù được thiết kế tốt, MOSFET kênh P trong các mạch cầu H dùng 2 loại MOSFET thường bị nóng và dễ hỏng hơn MOSFET loại N, công suất mạch cũng bị giảm phần nào.

3.2.8. Hiện thị bàn phím



Hình 3.15: Sơ đồ bàn phím

Chức năng bàn phím: dùng để thiết lập điều khiển chạy các chế độ của động cơ DC servo

Nút ấn từ 0 đến 9 là nút chọn tốc độ

Nút ấn C là nút xóa

Nút ấn D là nút đảo chiều

Nút ấn # là bắt đầu chương trình

Nút ấn * là nút dừng chương trình

3.2.9. Động cơ DC servo

Ở đây dùng động cơ DC servo:

Hãng kitashiba

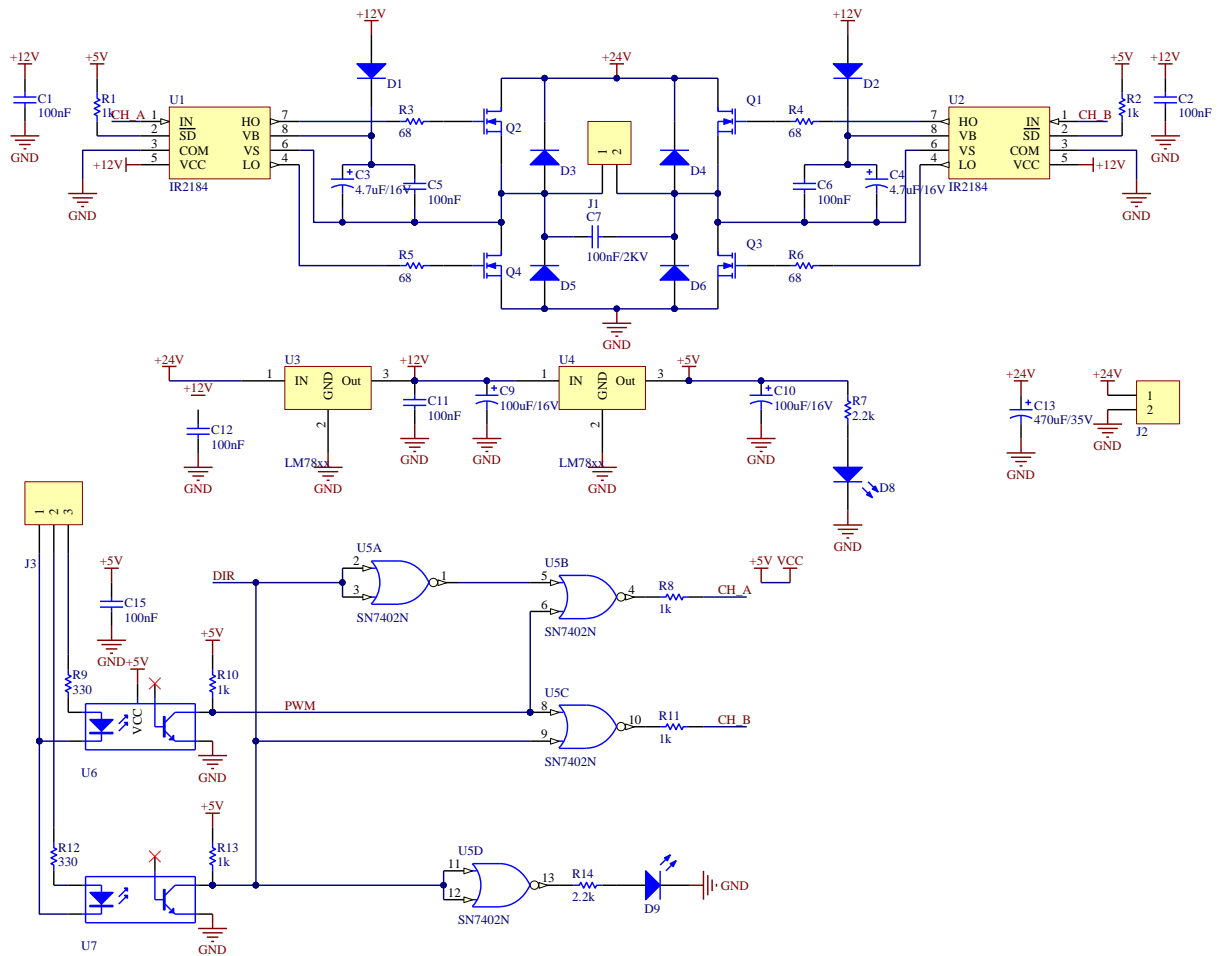
Điện áp vào là DC 24V

Có encoder gắn liền với trục động cơ

Độ phân giải của encoder là 100xung/vòng

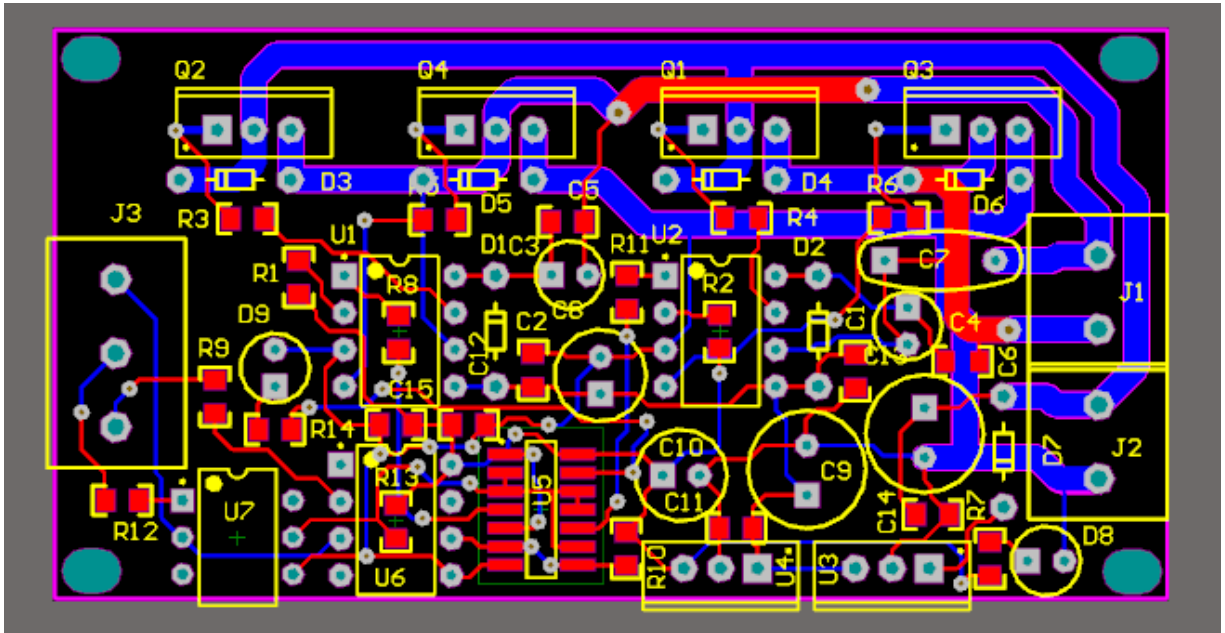
3.3. SƠ ĐỒ MẠCH ĐIỀU KHIỂN VÀ CHƯƠNG TRÌNH

3.3.1. Sơ đồ của mạch công suất

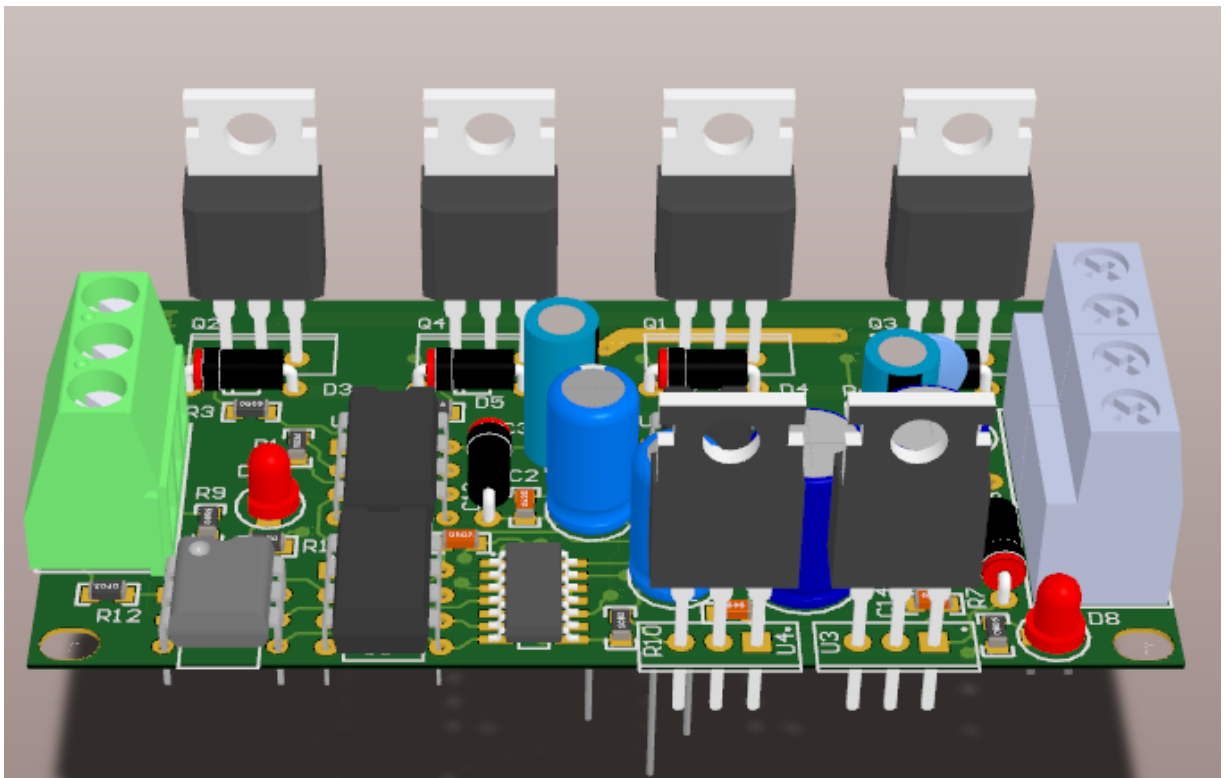


Hình 3.16: Sơ đồ nguyên lý mạch công suất

3.3.2. Sơ đồ mạch in của mạch công suất

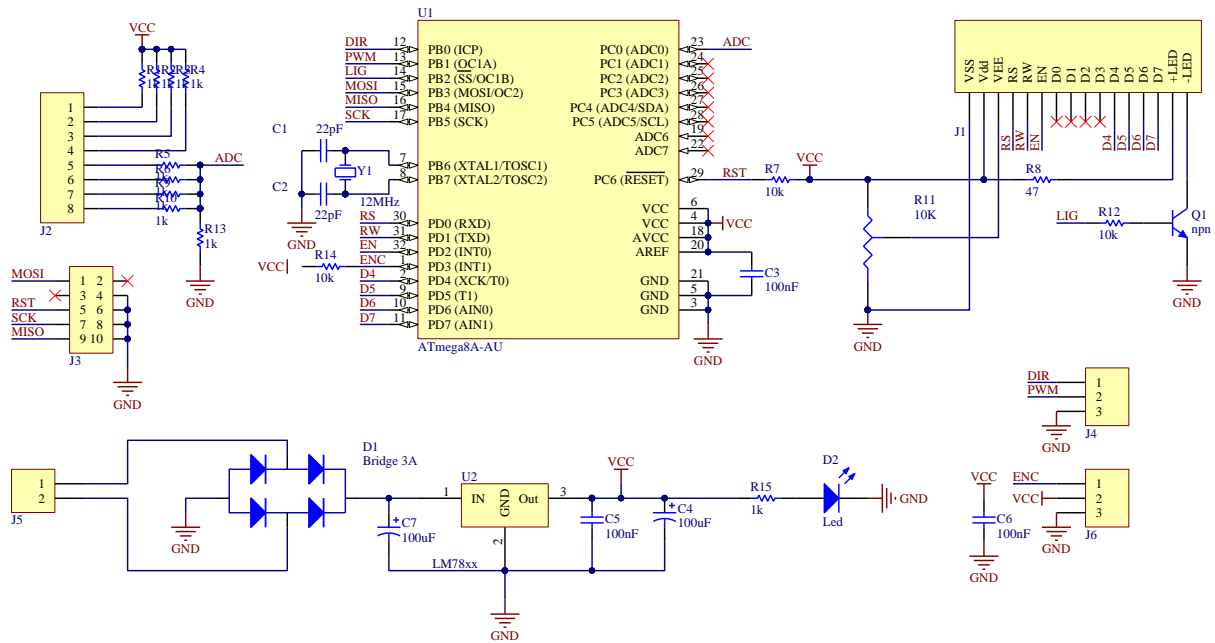


Hình 3.17: Sơ đồ mạch in 2D của mạch công suất



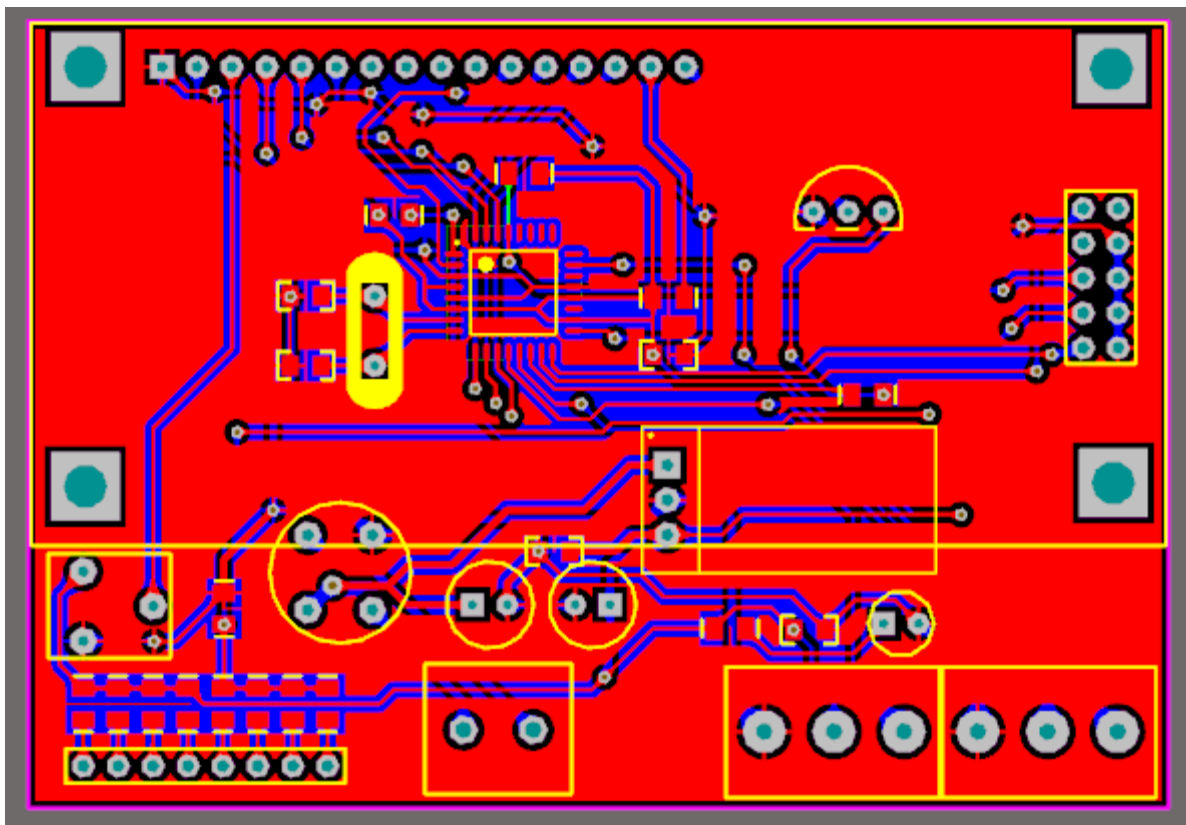
Hình 3.18: Sơ đồ mạch in 3D của mạch công suất

3.3.3. Sơ đồ nguyên lý của mạch điều khiển

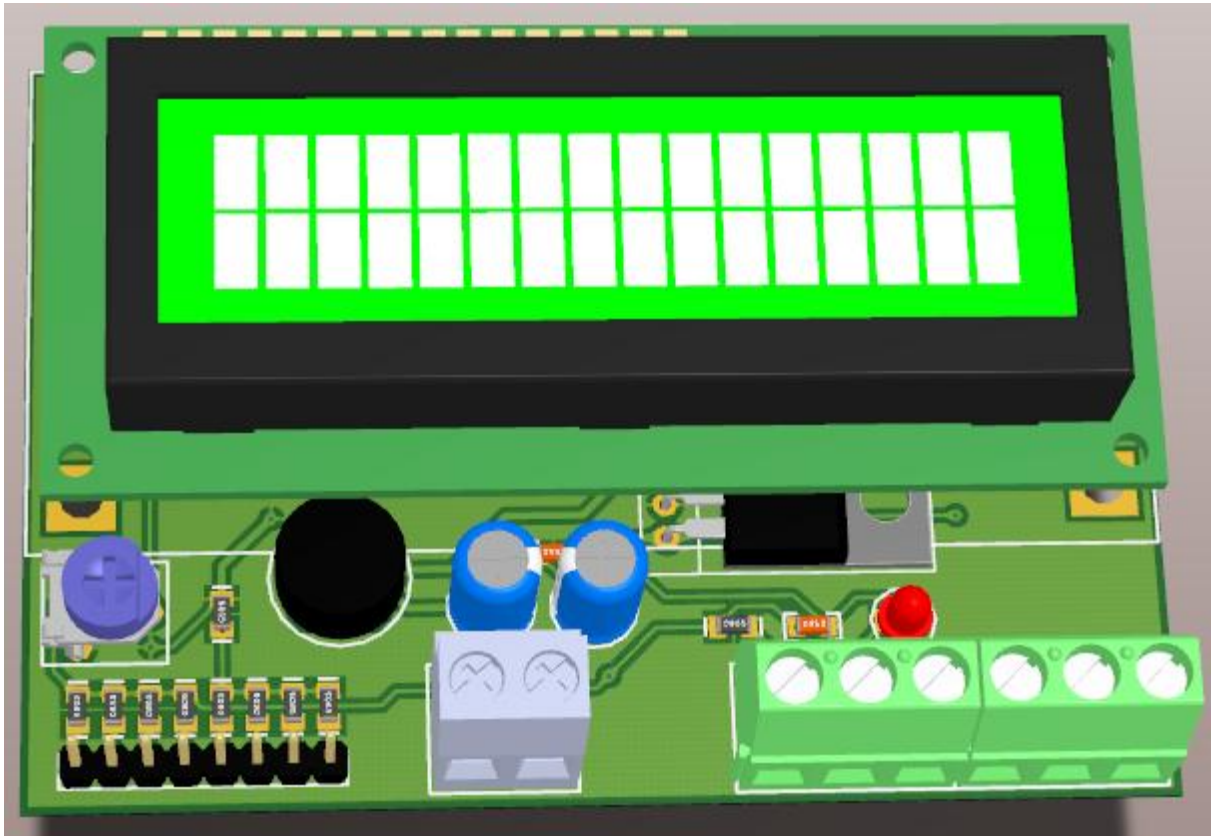


Hình 3.19: Sơ đồ mạch nguyên lý của mạch điều khiển

3.3.4. Sơ đồ mạch in của mạch điều khiển

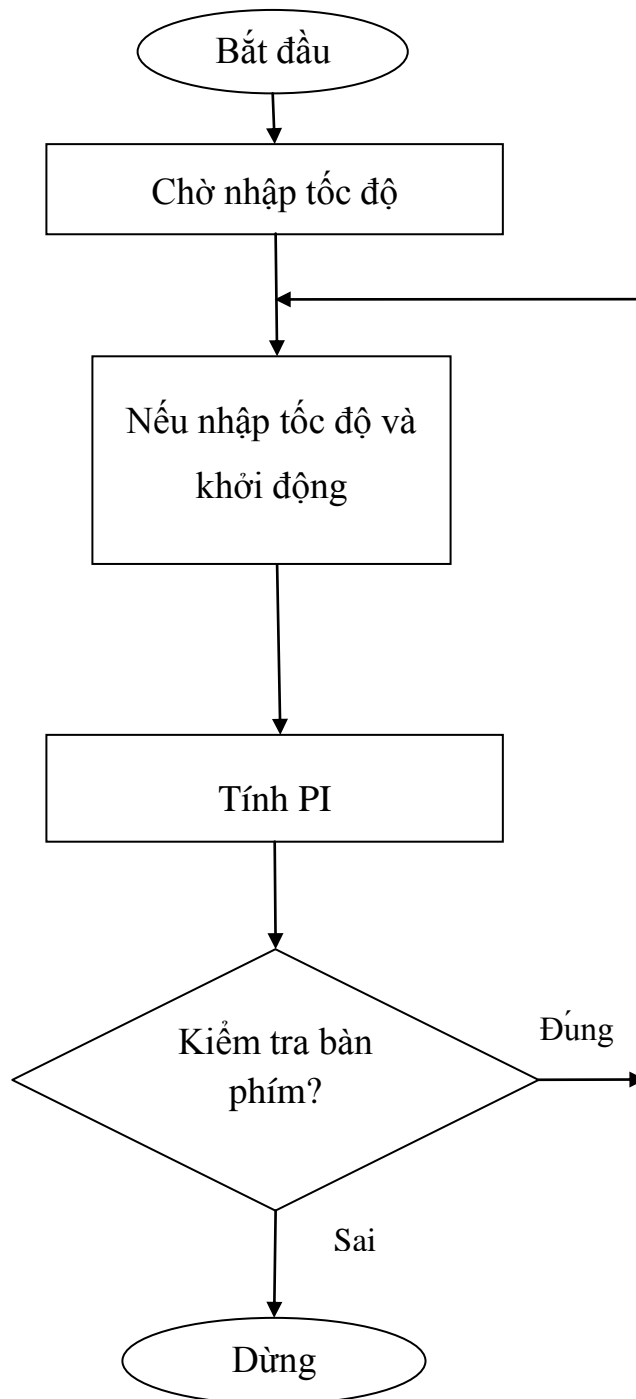


Hình 3.20: Sơ đồ mạch in 2D của mạch điều khiển



Hình 3.21: Sơ đồ mạch in 3D của mạch điều khiển

3.3.5. Lưu đồ thuật toán chương trình



Hình 3.22: Lưu đồ thuật toán

3.3.6 Chương trình điều khiển

Chương trình viết trên phần mềm Codevision 2.03.9

```
/*  
  
This program was produced by the  
Chip type      : ATmega8  
Program type   : Application  
Clock frequency : 12.000000 MHz  
Memory model   : Small  
External RAM size : 0  
Data Stack size : 256  
*****/  
  
#include <mega8.h>  
#define BackLight PORTB.2// Backlight  
unsigned char adc_in=0, Keypad=0;  
void lcd_put_int(int num);//Xuat 1 so nguyen ra LCD  
// Alphanumeric LCD Module functions  
#asm  
    .equ _lcd_port=0x12 ;PORTD  
#endasm  
#include <lcd.h>  
#include <delay.h>  
    #define FIRST_ADC_INPUT 0  
#define LAST_ADC_INPUT 0  
unsigned char adc_data[LAST_ADC_INPUT-FIRST_ADC_INPUT+1];  
#define ADC_VREF_TYPE 0x20
```

```

// ADC interrupt service routine
// with auto input scanning
interrupt [ADC_INT] void adc_isr(void)
{
static unsigned char input_index=0;
// Read the 8 most significant bits
// of the AD conversion result
adc_data[input_index]=ADCH;
// Select next ADC input
if (++input_index > (LAST_ADC_INPUT-FIRST_ADC_INPUT))
    input_index=0;
ADMUX=(FIRST_ADC_INPUT | (ADC_VREF_TYPE & 0xff))+input_index;
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
}
// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
// Place your code here
adc_in++;
}
// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{

```

```

// Reinitialize Timer 0 value
TCNT0=0x08;
// Place your code here
/*
  if(adc_data[0]>0) Keypad=10;// Neu an so 0
  else if(adc_data[0]>0) Keypad=11;// Neu an so 1
  else if(adc_data[0]>0) Keypad=12;// Neu an so 2
  else if(adc_data[0]>0) Keypad=13;// Neu an so 3
  else if(adc_data[0]>0) Keypad=14;// Neu an so 4
  else if(adc_data[0]>0) Keypad=15;// Neu an so 5
  else if(adc_data[0]>0) Keypad=16;// Neu an so 6
  else if(adc_data[0]>0) Keypad=17;// Neu an so 7
  else if(adc_data[0]>0) Keypad=18;// Neu an so 8
  else if(adc_data[0]>0) Keypad=19;// Neu an so 9
  else if(adc_data[0]>0) Keypad=20;// Neu an A
  else if(adc_data[0]>0) Keypad=21;// Neu an B
  else if(adc_data[0]>0) Keypad=22;// Neu an C
  else if(adc_data[0]>0) Keypad=23;// Neu an D
  else if(adc_data[0]>0) Keypad=24;// Neu an #
  else if(adc_data[0]>0) Keypad=25;// Neu an *
  */
}
// Declare your global variables here
void lcd_put_int(int num)//Xuat 1 so nguyen ra LCD
{
  int temp;

```

```

unsigned char i = 0, c[5];
temp = num;
if (temp != 0) {
    if (temp < 0){
        lcd_putchar('-');
        temp = - temp;
    }
    while(temp){
        c[i++] = temp%10;
        temp /= 10;
    }
    while(i) lcd_putchar(c[--i] + '0');
}
else lcd_putchar('0');
}
void main(void)
{
// Declare your local variables here
// Input/Output Ports initialization
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=Out
Func1=Out Func0=Out
// State7=T State6=T State5=T State4=T State3=T State2=0 State1=0
State0=0
PORTB=0x00;
DDRB=0x07;

```

```

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTD=0x00;
DDRD=0x00;
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 11.719 kHz
TCCR0=0x05;
TCNT0=0x08;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 46.875 kHz
// Mode: Ph. correct PWM top=00FFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge

```

```
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x01;
TCCR1B=0x04;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
// External Interrupt(s) initialization
// INT0: Off
// INT1: On
```



```

// INT1 Mode: Falling Edge
GICR|=0x80;
MCUCR=0x08;
GIFR=0x80;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x01;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
// ADC initialization
// ADC Clock frequency: 750.000 kHz
// ADC Voltage Reference: AREF pin
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=FIRST_ADC_INPUT | (ADC_VREF_TYPE & 0xff);
ADCSRA=0xCC;
// LCD module initialization
lcd_init(16);
// Global enable interrupts
#asm("sei")
    while (1)
    {
// Place your code here
        lcd_clear();

```

```
    adc_in=adc_data[0];  
    lcd_put_int(adc_in);  
    delay_ms(700);  
    //BackLight=~BackLight;  
};  
}
```

KẾT LUẬN

Sau một thời gian thực hiện đồ án đến nay đã kết thúc và đạt được những kết quả sau:

Tìm hiểu và tổng quát hóa các loại động cơ servo

Tìm hiểu được vi điều khiển atmega8

Thiết kế và xây dựng được mô hình điều khiển động cơ DC servo bằng vi điều khiển

Từ đề tài này có thể phát triển thêm để nâng cao công suất của mạch hơn nhằm áp dụng nhiều vào thực tiễn hơn.

Nhưng do kiến thức còn hạn hẹp nên em không thể tránh được những nhiều sót trong quá trình làm đồ án. Em rất mong nhận được sự chỉ bảo của những thầy cô trong hội đồng.

Hải phòng, ngày...tháng...năm 2012

Sinh viên thực hiện

Nguyễn Chí Vũ

TÀI LIỆU THAM KHẢO

1. PGS-TS Nguyễn Tiến Ban(2010), *Bài giảng Phần tử tự động*, Trường Đại học Hàng Hải Việt Nam
2. Đặng Văn Đào – Lê Văn Doanh (2006), *Giáo trình Kỹ thuật điện*, Nhà Xuất Bản khoa học và kỹ thuật Hà Nội
3. Ngô Diên Tập (2003), *Kỹ thuật vi điều khiển AVR*, NXB khoa học kỹ thuật
4. Lê Văn Doanh (2000), *Điện tử công suất*, NXB khoa học kỹ thuật
5. Phạm Quang Trí (2005), *Giáo trình thực hành vi xử lý*, Trường ĐHCN TP.HCM
6. Webside diễn đàn điện tử Việt Nam www.dientuvietnam.net
7. Webside www.alldatasheet.com
8. Webside www.tailieu.vn