BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA ĐIỆN - ĐIỆN TỬ

ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ - TRUYỀN THÔNG

ĐIỀU KHIỂN ĐỘNG CƠ KHÔNG ĐỒNG BỘ BA PHA SỬ DỤNG VI ĐIỀU KHIỂN DSPIC30F4011 THEO PHƯƠNG PHÁP SPWM

GVHD: ThS. HOÀNG NGỌC VĂN
SVTH: TÔ TIỂU VŨ
MSSV: 11141260
SVTH: PHẠM THANH TÂM
MSSV: 11141417

SKL004440

Tp. Hồ Chí Minh, tháng 12/2016

BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN ĐIỆN TỬ-CÔNG NGHIỆP

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG

ĐỀ TÀI:

ĐIỀU KHIỂN ĐỘNG CƠ KHÔNG ĐỒNG BỘ BA PHA SỬ DỤNG VI ĐIỀU KHIỂN DSPIC30F4011 THEO PHƯƠNG PHÁP SPWM

GVHD: GVC.Ths Hoàng Ngọc Văn
SVTH: Tô Tiểu Vũ
MSSV: 11141260
SVTH: Phạm Thanh Tâm
MSSV: 11141417

Tp. Hồ Chí Minh - 12/2016

BỘ GIÁO DỤC & ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN ĐIỆN TỬ-CÔNG NGHIỆP

ĐỒ ÁN TỐT NGHIỆP

NGÀNH KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG

ĐỀ TÀI:

ĐIỀU KHIỂN ĐỘNG CƠ KHÔNG ĐỒNG BỘ BA PHA SỬ DỤNG VI ĐIỀU KHIỂN DSPIC30F4011 THEO PHƯƠNG PHÁP SPWM

GVHD: GVC Ths Hoàng Ngọc Văn
SVTH: Tô Tiểu Vũ
MSSV: 11141260
SVTH: Phạm Thanh Tâm
MSSV: 11141417

Tp. Hồ Chí Minh – 12/2016

LỜI CAM ĐOAN

Kính thưa các thầy cô trong khoa công nghệ điện tử, sau bốn năm học tập tại trường, được các thầy cô tận tình hướng dẫn, em đã có được vốn kiến thức chuyên ngành đủ để có thể nhận đề tài đồ án tốt nghiệp. Em xin cam đoan sẽ cố gắng hoàn thành đề tài, tuân thủ đúng các hướng dẫn, yêu cầu của giáo viên hướng dẫn để có thể hoàn thành tốt đề tài cũng như khóa học của mình.Em xin đảm bảo đề tài này do em làm và hoàn thành. Trong quá trình làm em có tham khảo của một số đề tài đã làm trước đây, kèm theo phụ lục bên dưới.

Ngày 31 tháng 12 năm 2016

SINH VIÊN THỰC HIỆN

Phạm Thanh Tâm

Tô Tiểu Vũ

LỜI CẢM ƠN

Lời đầu tiên xin được gửi đến gia đình. Cảm ơn chân thành và biết ơn bố mẹ đã sinh ra và nuôi dạy em nên người, đã cho em ăn học để đến ngày hôm nay có thể cầm trên tay cuốn luận văn này. Gia đình luôn cho em động lực và mục tiêu sống.

Xin gửi lời cảm ơn chân thành đến Thầy Hoàng Ngọc Văn, người đã trực tiếp hướng dẫn em suốt thời gian qua. Thầy đã hết lòng hướng dẫn, truyền đạt kinh nghiệm cũng như kiến thức cho em, nhờ vậy em đã hoàn thành được luận văn này. Bên cạnh những kiến thức chuyên môn Thầy còn dạy cho em những kỹ năng sống: phương pháp giải quyết một vấn đề trong cuộc sống, tự tin vào bản thân, tính cẩn thận tỉ mỉ trong giải quyết một vấn đề, tinh thần trách nhiệm khi làm việc.

Xin cảm ơn chân thành đến quý thầy cô trường Đại học Sư Phạm Kỹ Thuật TP. Hồ Chí Minh, bộ môn Điện tử công nghiệp. Thầy cô đã tận tình chỉ bảo và truyền đạt kiến thức cho em trong suốt các năm học qua. Để em có thể tự tin là một người kỹ sư trong tương lai.

Em xin chân thành cảm ơn!

Em xin chân thành cám ơn!

Sinh viên thực hiện

Phạm Thanh Tâm

Tô Tiểu Vũ

MỤC LỤC

LỜI CAM ĐOAN

LỜI CẢM ƠN

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

MỤC LỤC

DANH SÁCH CÁC BẢNG VÀ HÌNH

NỘI DUNG **Trang**

CHƯƠNG 1 TỔNG QUAN..... 1

1.1 Đặt vấn đề..... 1

1.1 Mục tiêu..... 1

1.1 Nội dung..... 1

1.4 Giới hạn..... 1

1.5 Bố cục..... 1

CHƯƠNG 2 CƠ SỞ LÍ THUYẾT 3

2.1 PHƯƠNG PHÁP SINE PULSE WIDTH MODULATION 3

2.1.1 Phương Pháp Điều Rộng Xung SINPWM 3

2.1.2 Các công thức tính toán, Cách thức điều khiển 4

2.1.3 Quy trình tính toán 6

2.1.4 Hiệu Quả của phương pháp điều khiển 8

2.1.5 Kỹ thuật SinPWM cho VĐK Dspic30f4011..... 9

2.2 VI ĐIỀU KHIỂN DSPIC30F4011 12

2.2.1 Tổng quan về vi điều khiển dsPIC30F4011..... 12

2.2.2 Ngắt 15

2.2.3 Bộ định thời Timer 15

2.2.4 Bộ chuyển đổi tương tự sang số ADC 16

2.2.5 Khối PWM cho điều khiển động cơ 17

2.3 BỘ NGHỊCH LƯU 19
2.4 IC LÁI 20
2.5 ĐỘNG CƠ BA PHA 21
CHƯƠNG 3 TÍNH TOÁN VÀ THIẾT KẾ..... 22
3.1 THIẾT KẾ PHẦN CỨNG 22
3.1.1 Sơ đồ khối của hệ thống 22
3.1.2 Mạch vi điều khiển 23
3.1.3 Nguồn kích..... 25
3.1.4 Mạch công suất..... 27
3.1.5 Mạch driver..... 29
CHƯƠNG 4 THI CÔNG HỆ THỐNG 31
4.1 Giới thiệu 31
4.2 Thi công hệ thống và kết quả khác 31
4.2.1 Thi công bo mạch 31
4.2.2 Lắp ráp và kiểm tra 34
4.3 Lặp trình hệ thống 34
4.4 Phần mềm lặp trình cho chương trình 36
4.5 Viết tài liệu hướng dẫn sử dụng và thao tác..... 43

CHƯƠNG 5 KẾT LUẬN VÀ NHẬN XÉT ĐÁNH GIÁ 44
5.1 Bộ công suất..... 44
5.2 Bộ điều khiển..... 45
5.3 Mạch nguồn kích..... 46
5.4 Bộ vi xử lý 46
5.5 Kết quả mô phỏng 51
CHƯƠNG 6 KẾT LUẬN VÀ HƯỚNG GIẢ QUYẾT 54
TÀI LIỆU THAM KHẢO……..…..55
PHỤ LỤC ……..……56

LIỆT KÊ HÌNH VẼ

Hình **Trang**

Hình 2.1: Nguyên lý của phương pháp điều rộng xung sin 3
Hình 2.2: Nguyên lý của phương pháp điều rộng xung sin 4
Hình 2.3: Quá trình hoạt động của bộ điều khiển 6
Hình 2.4: Sơ đồ kết nối các khoá trong bộ nghịch lưu. 8
Hình 2.5: Sơ đồ dạng song Sine Pulse Width Modulation 10
Hình 2.6: Tín hiệu thu được 10
Hình 2.7 Giá trị đầu ra PWM 11
Hình 2.8: Sơ đồ chân dsPIC30f4011 13
Hình 2.9: Sơ đồ khối DSPIC30F4011 14
Hình 2.10: Sơ đồ khối Timer 1 16
Hình 2.11: Sơ đồ khối ADC 17
Hình 2.12: Sơ đồ khối của PWM điều khiển động cơ. 19
Hình 2.13: Sơ đồ điều khiển Mosfet 20
Hình 2.14: Động cơ 3 pha 21
Hình 3.1: Sơ đồ khối hệ thống 22
Hình 3.2: Sơ đồ nguyên lí mạch vi điều khiển 25
Hình 3.3: Sơ đồ nguyên lí mạch nguồn kích. 26
Hình 3.4: Sơ đồ nguyên lí kích và mạch công suất. 27
Hình 3.5: Sơ đồ nguyên lý mạch Driver 29
Hình 4.1: Mạch nguồn kích 30
Hình 4.2: Mạch công suất 31
Hình 4.3: Mạch điều khiển 32
Hình 4.4: Lưu đồ giải thuật 34
Hình 4.5: Mạch nạp Burn E 39
Hình 5.1: Mô hình mạch công suất 44
Hình 5.2: Mô hình mạch điều khiển 45
Hình 5.3: Mồ hình mạch nguồn kích 46
Hình 5.4: Màn hình khởi động khi mạch được cấp nguồn 47
Hình 5.5: Mô hình hoạt động ở chế độ 1 48

Hình 5.6: Mô hình hoạt động ở chế độ 2 49
Hình 5.7: Mô hình hoạt động ở chế độ 3 50
Hình 5.8: Dạng sóng ngõ ra PWM 51
Hình 5.9: Dạng sóng ngõ ra ở tải (1 pha) 52
Hình 6.0: Dạng sóng ngõ ra trên tải thực tế (động cơ) 53

CHƯƠNG 1
TỔNG QUAN

1.1 Đặt vấn đề:

Hiện nay, công nghiệp phát triển mạnh mẽ, máy móc thay thế cho sức người. Động cơ là nguồn truyền động phổ biến nhất. Trong các loại động cơ không đồng bộ ba pha được sử dụng nhiều nhất do cấu tạo đơn giản, giá thành rẻ, công suất lớn. Nhưng nhược điểm lớn nhất của loại động cơ này là khó điều khiển. Đứng trước vấn đề này, các kỹ sư luôn muốn tìm ra một phương pháp để có thể điều khiển tối ưu, chi phí chế tạo bộ điều khiển thấp để có thể đưa động cơ không đồng bộ ba pha ứng dụng rộng rãi hơn nữa, nhất là trong các ứng dụng mà động cơ này còn bị hạn chế.

Trong các phương pháp điều khiển hiện nay thì phương pháp SINE PULSE WIDTH MODULATION cho kết quả tốt. Nhưng phương pháp khó thực hiện hơn các phương pháp khác nên chưa được ứng dụng nhiều ngoài thực tế. Chính vì lý do này mà em chọn đề tài ĐIỀU KHIỂN ĐỘNG CƠ KHÔNG ĐỒNG BỘ BA PHA SỬ DỤNG VI ĐIỀU KHIỂN DSPIC30F4011 THEO PHƯƠNG PHÁP SPWM với mục đích tìm hiểu, nghiên cứu, thiết kế bộ điều khiển theo phương pháp này. Với sự tìm hiểu ban đồng này và việc sẽ nghiên cứu sâu hơn sau này có thể đưa phương pháp này ứng dụng nhiều trong thực tiễn.

1.2 Mục tiêu:

- Tìm hiểu và thiết kế bộ biến tần truyền thống (6 khóa) ba pha điều khiển ĐCKĐB theo phương pháp V/f và điều chế SINPWM.
- Khảo sát nguyên tắc đóng cắt các khóa bán dẫn trong bộ nghịch lưu.
- Nguyên cứu giải thuật và viết chương trình điều khiển.

1.3 Nội dung:

Nội dung 1: Thiết kế bộ biến tần sử dụng phương pháp SWM
Nội dung 2: Thiết kế khối công suất.
Nội dung 3: Thiết kế mạch điều khiển: nút nhấn, LCD.
Nội dung 4: Đánh giá kết quả thực hiện.

1.4 Giới hạn:

Mô hình điều khiển động cở 3 pha không đồng bộ ba pha sử dụng vi điều khiển DsPic30f4011 theo phương pháp SWPM là mô hình điều khiển động cơ công suất nhỏ P=120W. Sử dụng điện áp 220VAC 50Hz, sau khi qua bộ biến tần tạo điện áp mới Vout = 220VAC với tần số thay đổi 0 – 100Hz.

1.5 Bố cục:

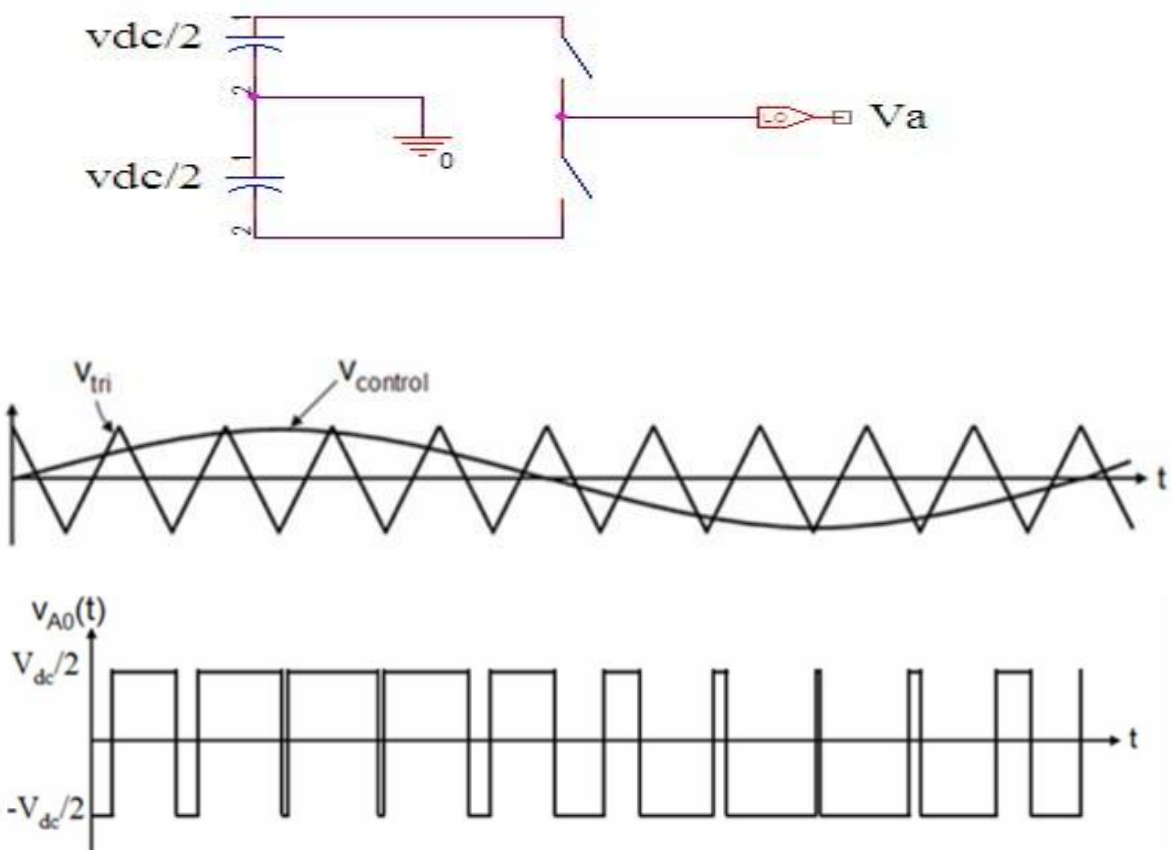
- Chương 1: Tổng quan chương này trình bày đặt vấn đề dẫn nhập lý do chọn đề tài, mục tiêu, nội dung đề tài thực hiện.
- Chương 2: Cơ sở lý thuyết trình bày cơ sở lý thuyết liên quan đến phương pháp SPWM, chíp vi xử lý, mosfet/IGBT.
- Chương 3: Tính toán và thiết kế hệ thống mô tả quá trình thiết kế hệ thống, các sơ đồ kết nối của hệ thống.
- Chương 4: Thi công hệ thống trình bày quá trình thi công để hoàn thiện hệ thống, qua trình lập trình các vấn đề liên quan đến hệ thống.
- Chương 5: Kết quả-nhận xét: tổng kết lại kết quả đã đạt được, nêu lên nhận xét đánh giá về toàn bộ quá trình làm việc.
- Chương 6: Kết Luận - hướng phát triển: kết luận những gì thu được trong quá trình thực hiện hệ thống, nêu hướng phát triển cho đề tài.

CHƯƠNG 2 CƠ SỞ LÍ THUYẾT

2.1 PHƯƠNG PHÁP SINE PULSE WIDTH MODULATION

2.1.1 Phương pháp điều rộng xung sinpwm:

Để tạo ra điện áp xoay chiều bằng phương pháp SinPWM, ta sử dụng một tín hiệu xung tam giác tần số cao đem so sánh với một điện áp sin chuẩn có tần số f. Nếu đem xung điều khiển này cấp cho một bộ biến tần một pha thì ngõ ra sẽ thu được một dạng điện áp dạng điều rộng xung có tần số bằng tần số nguồn sóng sin mẫu và biên độ hài bậc nhất phụ thuộc vào nguồn điện một chiều cung cấp và tỉ số giữa biên độ sóng sin mẫu và sóng mang. Tần số sóng mang phải lớn hơn tần số sóng sin mẫu. Sau đây là hình vẽ miêu tả nguyên lý của phương pháp điều rộng xung sin.

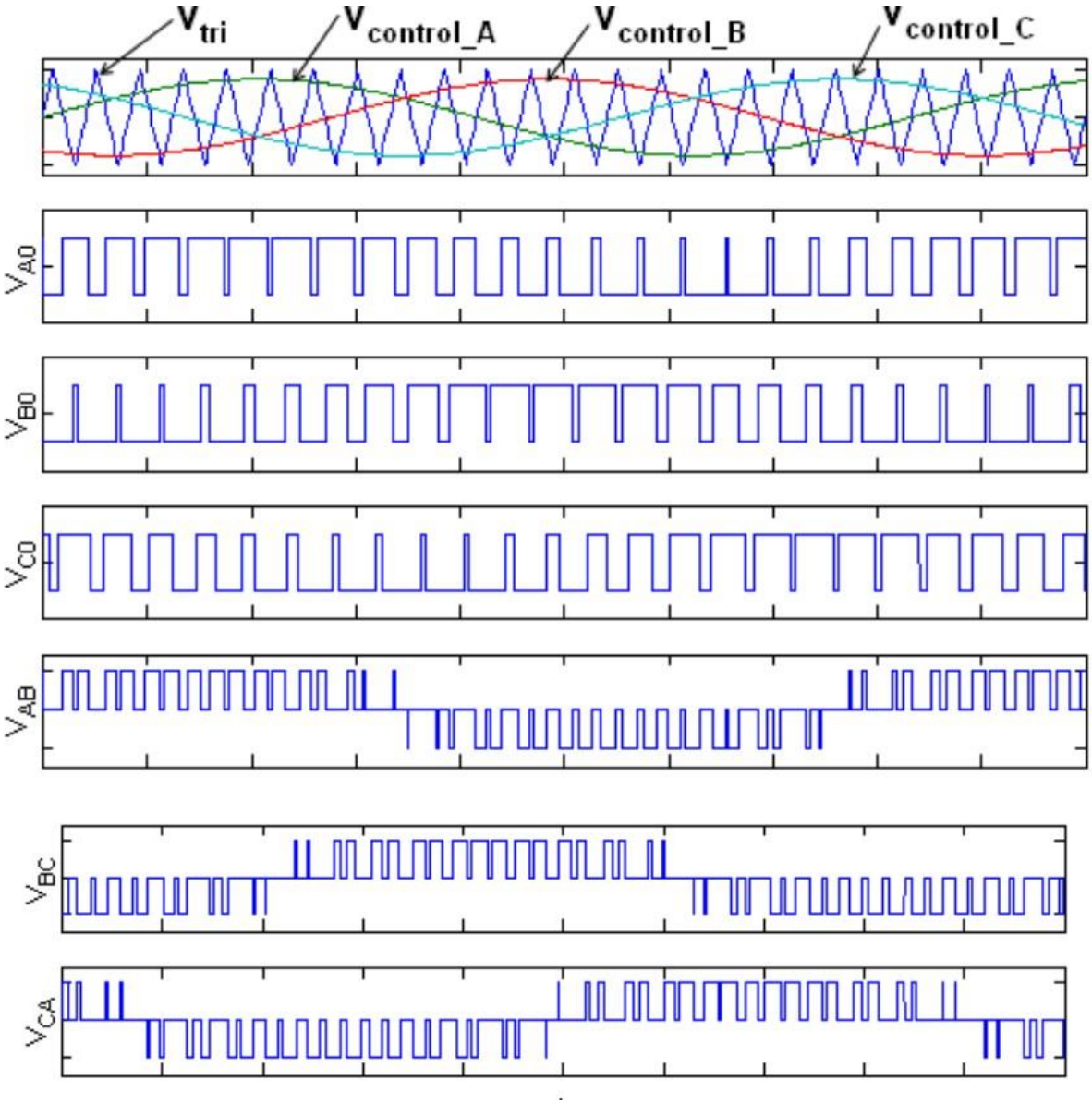


Hình 2.1 Nguyên lý của phương pháp điều rộng xung sin.

Khi: $V_{control} > V_{tri}$ thì $Vao = \frac{Vdc}{2}$

$V_{control} < V_{tri}$ thì $Vao = -\frac{Vdc}{2}$

Như vậy, để tạo ra nguồn điện ba pha dạng điều rộng xung, ta cần có nguồn sin ba pha mẫu và giản đồ kích đóng của ba pha sẽ được biểu diễn như hình dưới đây:



Hình 2.2 Sơ đồ dạng điện áp trên các pha.

2.1.2 Các công thức tính toán, cách thức điều khiển:

Ta cần tính được biên độ hài bậc nhất của điện áp ngõ ra từ biên độ giữa sóng mang và sóng tam giác.

Công thức tính biên độ hài bậc nhất:

$$Ut = ma\frac{Udc}{2}$$

Trong đó ma lả tỉ số giữa biên độ sóng sin mẫu mà biên độ sóng mang – còn gọi là tỉ số điều biên.

$$ma = \frac{Udk}{Ucarry}$$

Cách thức điều khiển:

Sau khi đã nói về phương pháp điều khiển V/f= const và phương pháp điều khiển bộ nghịch lưu áp theo phương pháp điều rộng xung SINPWM, ta có thể đưa ra một thuật toán điều khiển động cơ theo tần số đặt cho trước như sau.

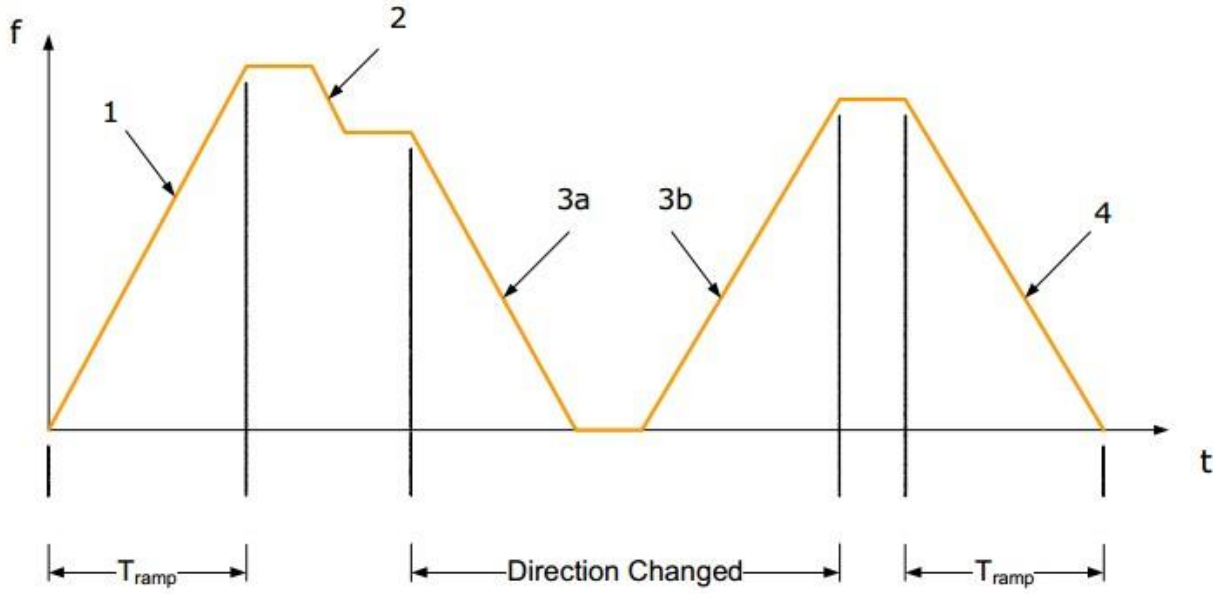
Trường hợp ta muốn cho động cơ ở trạng thái đứng yên chuyển sang chạy ở tần số đặt thì phải thông qua một quá trình khởi động mềm tránh cho động cơ khởi động lập tức đến tốc độ đặt, gây ra dòng điện khởi động lớn làm hỏng động cơ. Tần số nguồn cung cấp sẽ tăng từ giá trị 0(đứng yên). Thời gian khởi động này có thay đổi theo công suất của từng động cơ. Đối với động cơ có công suất lớn thời gian khởi động lâu hơn động cơ có công suất nhỏ. Thời gian khởi động của đông cơ thông thường được chọn từ 5 tới 10giây.

Sau khi tần số nguồn đã đạt tới giá trị yêu cầu lúc đầu sẽ giữ nguyên giá trị đó. Trong quá trình động cơ đang chạy ổn định mà có một nhu cầu thay đổi tần số thì cũng có một quá trình chuyển đổi tần số từng bước thay vì nhảy ngay lập tức đến giá trị tần số yêu cầu mới.

Khi muốn thay đổi chiều của động cơ cần phải đưa tần số động cơ về đủ nhỏ rồi sau đó mới thực hiện chuyển đổi ngay (thay đổi thứ tự nguồn pha cung cấp cho động cơ) tránh hiện tượng moment xoắn có thể làm gãy trục động cơ và tăng dòng đột ngột.

Khi muốn dừng động cơ thì phải hạ tần số từ giá trị hiện tại về giá trị 0. Thời gian hãm này phụ thuộc vào quán tính quay của động cơ. Khi muốn hãm phanh có thể dùng các phương pháp hãm như phương pháp hãm động năng – có dung điện trở thắng.

Như vậy ta có thể hình dung quá trính hoạt động của bộ điều khiển như sau:



Hình 2.3 Quá trình hoạt động của bộ điều khiển.

Đoạn 1 ứng với khởi động động cơ - tần số tăng từ 0 tới giá trị đặt sau khoảng thời gian khởi động (T_{ramp}).

Đoạn 2 ứng với việc thay đổi tần số khi động cơ đang chạy ổn định.

Đoạn 3 ứng với chiều chuyển đội động cơ - được chia làm 2 giai đoạn: đoạn 3a ứng với giảm tần số về 0, cuối đoạn 3a sẽ thay đổi thứ tự pha nguồn cung cấp cho động cơ, đoạn 3b ứng với tăng tần số lên đến giá trị mới.

Đoạn 4 ứng với ngừng động cơ. Tần số cho động cơ được giảm dần từ giá trị đặt về 0 sau khoảng thời gian dừng (T_{ramp}).

2.1.3 Quy trình tính toán

Tần số sóng mang trong MCU 4011 là trong MCU 4011 là

$$\text{PTPER} = \frac{Fcy}{(Fpwm*prescale*2)-1} \qquad (1)$$

Trong đó PTPER là giá trị cần nạp vào thanh ghi để có được tần số song mang mong muốn.

$$F_{cy} = \frac{fosc}{4} \qquad (2)$$

Với Fosc là tần số thạch anh đưa vào vi điều khiển.

Ứng với các giá trị tính toán trên, để tạo ra một song mang có tần số 5 kHz. Giá trị cần nạp vào thanh ghi PTPER là 1999.

Sóng điều khiển (U_{dk}) được tạo ra bằng cách lập một bảng sin có giá trị từ 0 tới 2π tượng trưng cho một chu kì sóng điều khiển dạng sin. Như đã biết, sóng điều khiển mang thông tin về độ lớn giá trị hiệu dụng và tần số sóng hài cơ bản của điện áp ngõ ra, vì vậy khi biên độ và tần số của sóng diều khiển thay đổi thì ta có điện áp và tần số ngõ ra cũng thay đổi theo.

Tần số của sóng điều khiển thay đổi tuỳ thuộc vào tần số di chuyển của con trỏ trong bảng sin. Nếu tần số của sóng điều khiển càng lớn thì số bước nhảy của sóng điều khiển di chuyển của con trỏ di chuyển trong bảng sin trong một chu kì sóng điều khiển càng ít và ngược lại. Quan hệ giữa số bước nhảy của con trỏ trong bảng sin và tần số của sóng điều khiển được xác định theo công thức sau:

$$K = \frac{\alpha}{\alpha min} \qquad (3)$$

Trong đó α_{min} là độ phân giải của bảng sin (với bảng sin gồm có 720 giá trị thỉ độ phân giải của bảng sin là 0.5 độ/giá trị).

$$\alpha = \frac{T_{pwm} * 360}{T_{Udk}} \qquad (4)$$

Là góc nhảy của con trỏ trong bảng sin sau một chu kì PWM.

Từ công thức 1 và 2 ta có: $Ut = \frac{Udk}{Ucarry} * \frac{Udc}{2}$ (3)

Trong đó U_{dc} = 310V ứng với trường hợp tỉ số điều chế ma=1 và động cơ hoạt động ở chế độ định mức.

Khi động cơ hoạt động ở chế độ định mức:

$$\frac{U_{dm}}{F_{dm}} = 3.667 \ (U_{dm} = 220, f_{dm} = 60Hz) \qquad (4)$$

từ 3 và 4 ta có:

$$f_{req} = 0.03 * U_{dk} \qquad (5)$$

Giá trị của tần số đặt vào động cơ từ biến trở được thông qua bộ chuyển đổi ADC 10bits theo công thức:

$$f_{req} = ADC_Result * 60/1024. \qquad (6)$$

Dựa vào tần số yêu cầu đầu vào ta có thể tính biên độ của sóng điều khiển để giữ cho tỉ số V/f không đổi.

Thời gian tăng tốc và giảm tốc của động cơ được tính toán dựa vào chu kì PWM, kể từ khi có sự thay đổi về tần số đặt, sau mỗi chu kì PWM, giá trị tần số hiện tại sẽ cộng thêm vào hoặc trừ ra một giá trị cho tới khi bằng với giá trị của tần số đặt mới.

Giá trị cộng vào hoặc trừ ra được tính theo công thức sau:

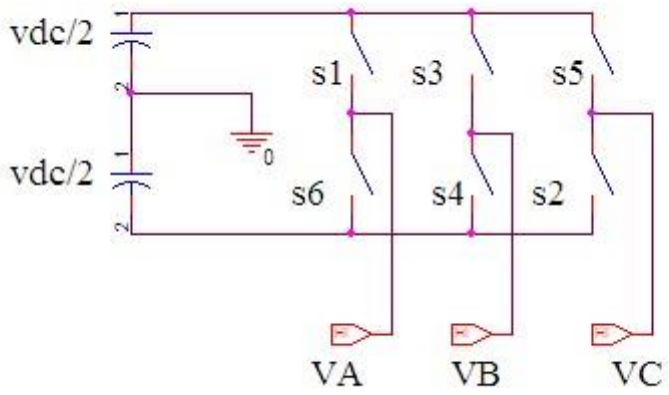
$$\Delta = (60*T_{PWM})/t \qquad (7)$$

Trong đó t(s) là thời gian tăng tốc hoặc giảm tốc của động cơ.

Để đảm bảo sự chuyển mạch diễn ra đúng, tại một thời điểm trên một nhánh chỉ có một khoá bán dẫn trong trạng thái dẫn, một khoảng thời gian nghỉ (dead time) cần được thêm vào khoảng giữa hai khoá, với tần số thạch anh đưa vào vi điều khiển lá 10Mhz, tần số sóng mang là 5Khz, khoảng thời gian nghỉ từ 1 tới 25ms, ở đây khoảng thời gian chọn là 2ms.

2.1.4 Hiệu quả phương pháp điều khiển:

Đối với phương pháp điều chế SINPWM, tại thời điểm mà một trong hai khoá trên cùng một nhánh ở trạng thái ON thì biểu thức điện áp giữa một pha và điểm trung tín ảo (O) có dạng như sau:



Hình 2.4 Sơ đồ kết nối các khoá trong bộ nghịch lưu.

$$V_{AO} = \frac{V_{DC}}{2} * (m * \sin(\alpha))$$
$$V_{BO} = \frac{V_{DC}}{2} * (m * \sin(\alpha + \frac{2\pi}{3})) \quad (7a)$$
$$V_{CO} = \frac{V_{DC}}{2} * (m * \sin(\alpha + \frac{4\pi}{3}))$$

Điện áp giữa hai pha đươc tính toán:

$$V_{AB} = V_{AO} - V_{BO} = \frac{\sqrt{3}}{2} Vdc.m.\sin(\alpha + \frac{\pi}{6})$$
$$V_{BO} = \frac{\sqrt{3}}{2} Vdc.m.\sin(\alpha + \frac{5\pi}{6}) \quad (7b)$$
$$V_{CO} = \frac{\sqrt{3}}{2} Vdc.m.\sin(\alpha + \frac{3\pi}{2})$$

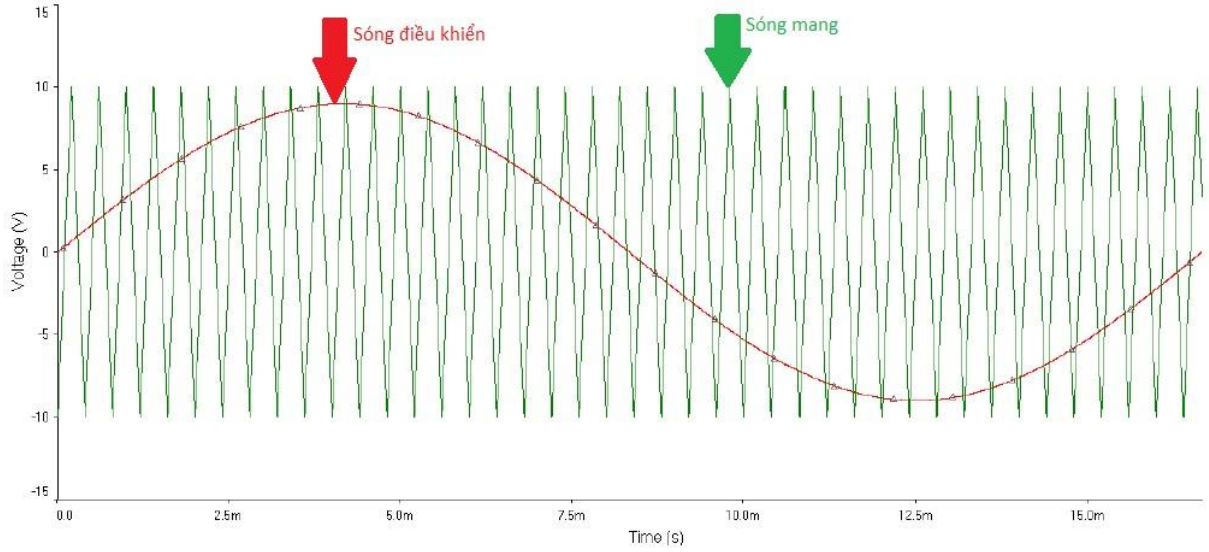
Từ công thức 7a với 7b trên ta thấy giá trị điện áp lớn nhất giữa hai pha đạt được giữa hai vùng tuyến tính khi m=1.

Giá trị điện áp lớn nhất là $V_{line_to_line_max} = \frac{\sqrt{3}}{2} Vdc$ (8)

Vậy với phương pháp này, điện áp bộ nghịch lưu cung cấp chỉ dược sử dụng tối đa 86,67% trong vùng diều khiển tuyến tính.

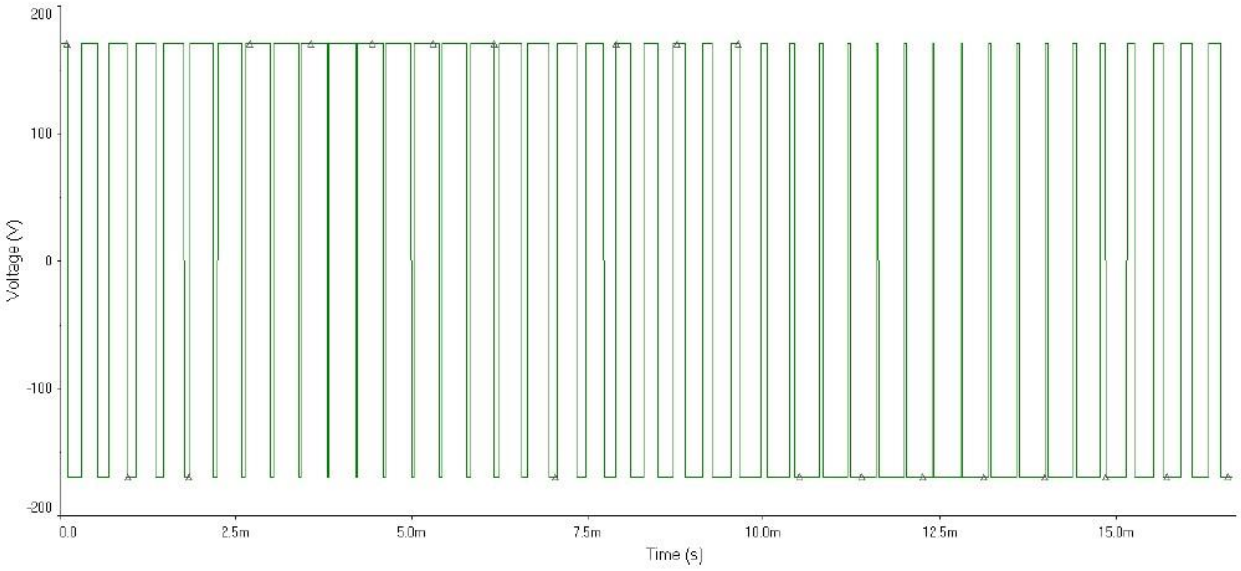
2.1.5 Kỹ thuật SinPWM cho VDK Dspic30f4011:

SinPWM là một kỹ thuật nghịch lưu bằng cách điều rộng xung PWM kích mạch công suất MOSFET/IGBT tạo ra tín hiệu AC gần sin (Pure Sine Inverters). Mức logic của xung PWM được quyết định bởi việc so sánh một sóng tam giác (sóng mang) và một sóng sin chuẩn (sóng điều khiển). Tần số sóng mang càng lớn thì tín hiệu AC dạng sine thu được càng "pure".



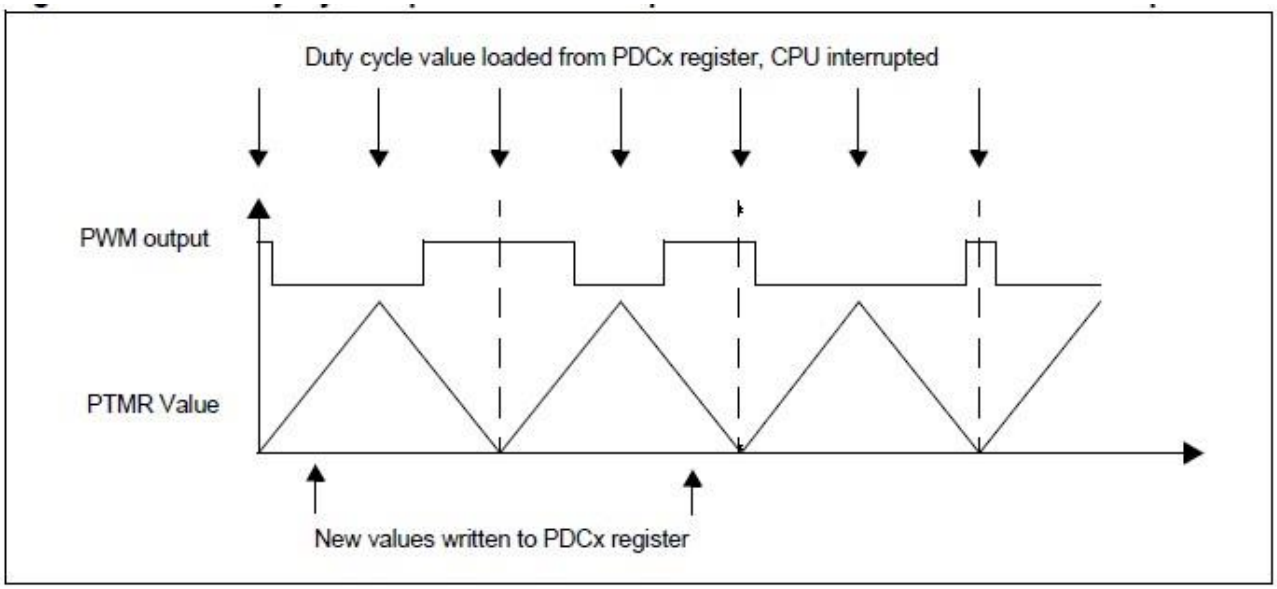
Hình 2.5 Sơ đồ dạng sóng Sine Pulse Width Modulation

Khi giá trị sóng mang nhỏ hơn sóng điều khiển, PWM ở mức thấp (inactive), khi giá trị sóng mang lớn hơn sóng điều khiển, PWM ở mức cao (active). Tín hiệu PWM thu được sẽ là SinPWM:



Hình 2.6 Tín hiệu thu được.

Áp dụng vào vi xử lý dsPIC30F4011 ta có module PWM khi hoạt động ở chế độ Up/Down Count Mode with Double Updates có giá trị bộ đếm (PTMR) tăng/giảm tuyến tính nên tạo thành một dạng sóng digital tam giác.



Hình 2.7 Giá trị đầu ra PWM.

Mức logic của ngõ ra phụ thuộc vào giá trị của thanh ghi PDCx, nếu PDCx nhỏ hơn PTMR thì ngõ ra tích cực, ngược lại thì ngõ ra thụ động. Căn cứ vào nguyên lý làm việc này, nếu giá trị PDCx có dạng sin digital thì ngõ ra PWM sẽ là SinPWM.

Với thạch anh 7.3728 MHz, tỷ lệ nhân x16 ta có:

Tốc số thực hiện lệnh: Fcy = (7.3728 x 16)/4 = 29.4912 MHz.

Chu kỳ lệnh: Tcy = 1/Fcy = 33.9 ns. $\frac{1}{(PTMR*4*T_{CY})*2} = \frac{1}{2000*4*33.9ns*2}$

Đặt bộ đếm PWM có chu kỳ xung clock = ¼ Tcy, giá trị cực đại của bộ đếm là 2000 (PTMR = 2000). Khi đó tần số sóng mang là:

$$Fm = 3686.4 \text{ Hz} \approx 3.7 \text{ kHz}$$

Tín hiệu điều khiển dạng digital sin được tạo ra bằng cách cập nhật giá trị (lấy mẫu) từ bảng sin tính sẵn (bảng sin 720 giá trị). Việc cập nhật được thực hiện thông qua một ngắt Timer với giá trị Timer (PRx) được tính là:

PRx = Fcy / (720/số bước nhảy trong bảng sin) / tần số đặt.

Việc lựa chọn bước nhảy phải làm sao cho tín hiệu sin càng nhuyễn càng tốt và việc ngắt Timer không được ảnh hưởng đến chương trình chính. Do đó, số bước nhảy được chọn như sau:

Tần số sóng sin	Số bước nhảy
1~25 Hz	1
26~50 Hz	2
51~75 Hz	3
76~100 Hz	4
101~125 Hz	5
126~150 Hz	6
151~200 Hz	8

Bảng 2.1: Mối liên hệ giữa tần số và bước nhảy.

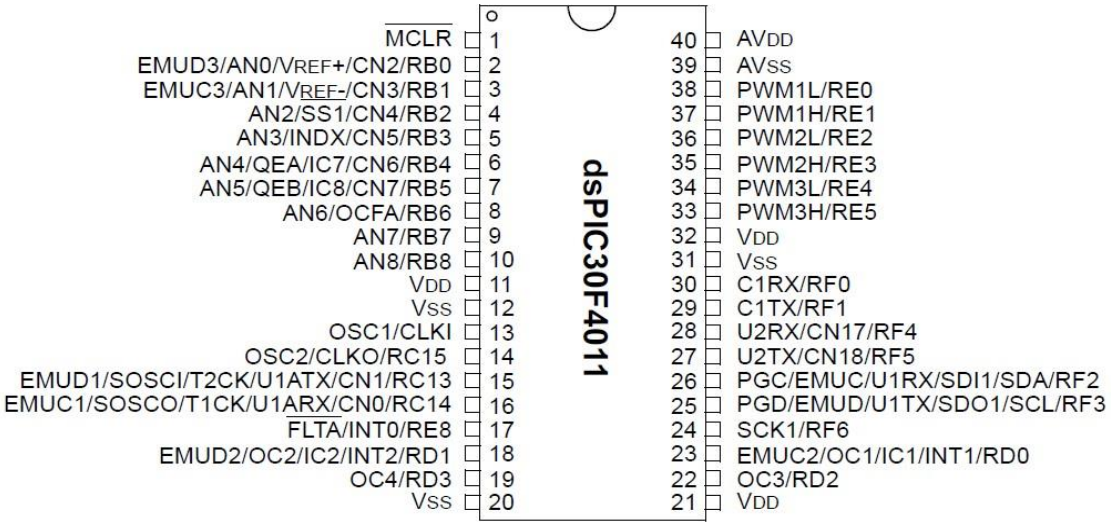
2.2 VI ĐIỀU KHIỂN DSPIC30F4011

2.2.1 Tổng quan về vi điều khiển dsPIC30F4011

DSPIC30F4011 là một chíp xử lí mạnh với bộ xử lí 16 bit. Tốc độ xử lí có thể lên tới 30 triệu lệnh một giây, có thể thực hiện chức năng của một bộ xử lí tín hiệu số. DSPIC còn được trang bị bộ nhớ flash, bộ nhớ EEPROM và các ngoại vi hiệu năng cao và rất đa dạng các thư viện phần mềm cho phép thực hiện các giải thuật nhúng với hiệu suất cao trong một khoảng thời gian ngắn. Vì vậy, dsPIC được ứng dụng rộng rãi trong các ứng dụng xử lí tín hiệu số, đo lường và điều khiển tự động.

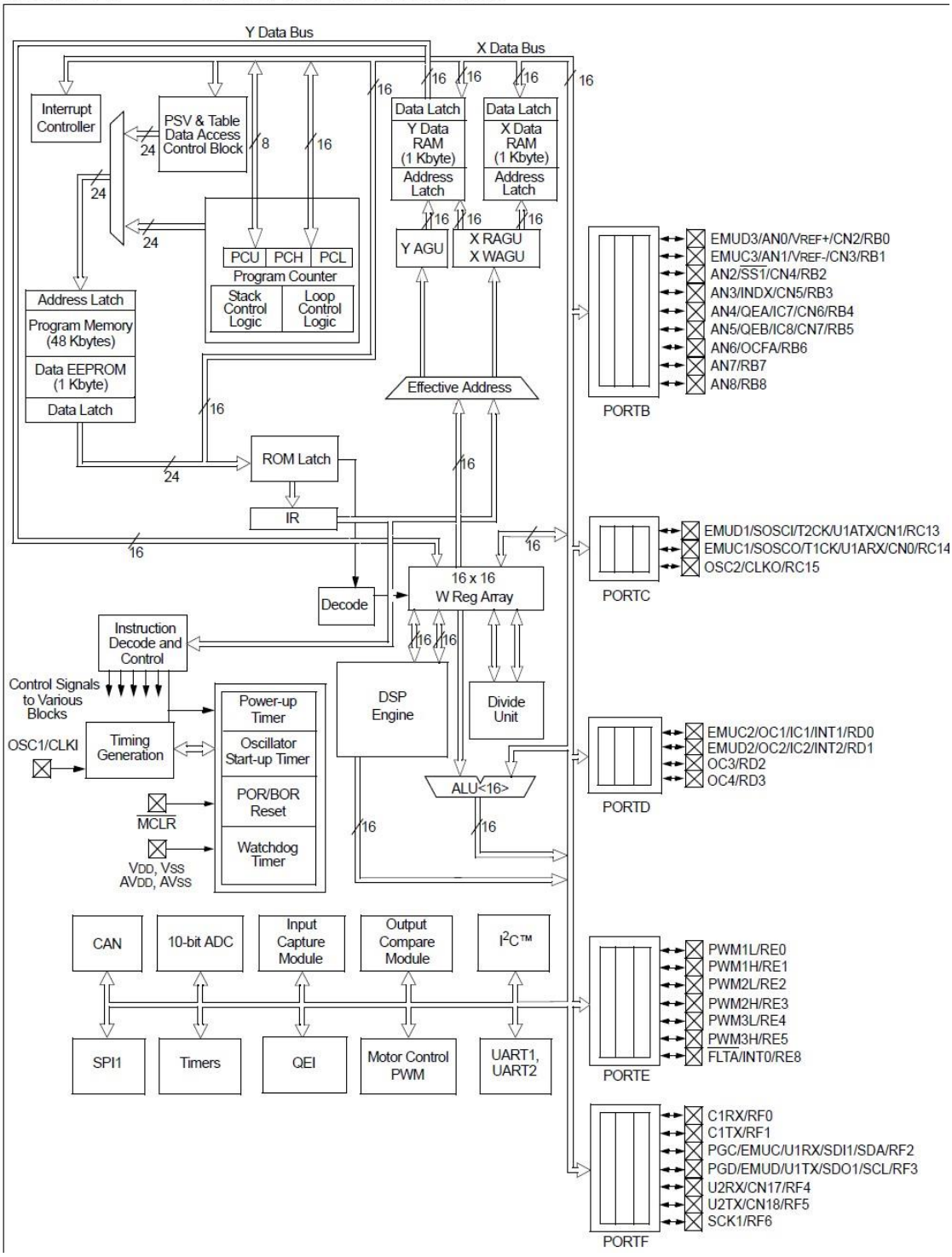
Một số đặt điểm của dsPIC30F4011:

- Bộ nhớ chương trình 48K.
- Bộ nhớ RAM 2 Kbytes
- Bộ nhớ EEPROM 1Kbyte
- Xung clock ngoài tối đa lên tới 40MHz.
- 30 nguồn ngắt.
- 5 timer 16 bit có thể ghép thành timer 32 bit
- Chế độ bắt giữ, so sánh, điều rộng xung 16 bit.
- Giao tiếp SPI, I^2C, UART, CAN.
- 6 kênh phát PWM chuyên dùng điều khiển động cơ.
- Module QEI chuyên đọc encoder.
- □ ADC 10 bit tốc độ cao.



Hình 2.8 Sơ đồ chân dsPIC30f4011.

FIGURE 1-1: dsPIC30F4011 BLOCK DIAGRAM



Hình 2.9 Sơ đồ khối DSPIC30F4011.

2.2.2 Ngắt:

Dspic30f4011 có 30 nguồn ngắt. Bảng vecto ngắt được đặt gần bộ nhớ chương trình tại địa chỉ 0x000004. Bộ điều khiển ngắt điều khiển các ngắt. Thiết lập các ngắt thông qua các thanh ghi đặc biệt sau:

- IFS0<15:0>, IFS1<15:0>, IFS2<15:0>: chứa tất cả các cờ ngắt. Cờ ngắt được bật lên bởi ngoại vi của vi điều khiển hoặc tín hiệu từ bên ngoài và được xóa bằng phần mềm.
- IEC0<15:0>, IEC1<15:0>, IEC2<15:0>: chứa tất cả các bit điều khiển cho phép ngắt. Các bit này được sử dụng để bật các ngắt riêng biệt từ các thiết bị ngoại vi hay tín hiệu từ bên ngoài.
- IPC0<15:0>... IPC11<7:0>: thiết lập các ngắt ưu tiên thông qua 12 thanh ghi.
- IPL<3:0>: các mức độ ưu tiên được lưu trữ trong các bit IPL. IPL<3> nằm trong thanh ghi CORCON, IPL<2:0> nằm trong thanh ghi trạng thái.
- INTCON1<15:0>, INTCON2<15:0>: điều khiển ngắt toàn cục. INTCON1 bao gồm các bit cờ điều khiển các ngắt khi có lỗi xử lí. INTCON2 điều khiển các tín hiệu yêu cầu ngắt ngoại vi và việc sử dụng bảng chuyển đổi các vecto ngắt.

Tất cả các nguồn ngắt có thể được người dùng gán cho một trong 7 cấp độ ưu tiên thông qua các thanh ghi IPCx. Nếu bit NSTDIS (INTCON1 <15>) được thiết lập, tất cả các ngắt đều bị cấm.

2.2.3 Bộ định thời Timer

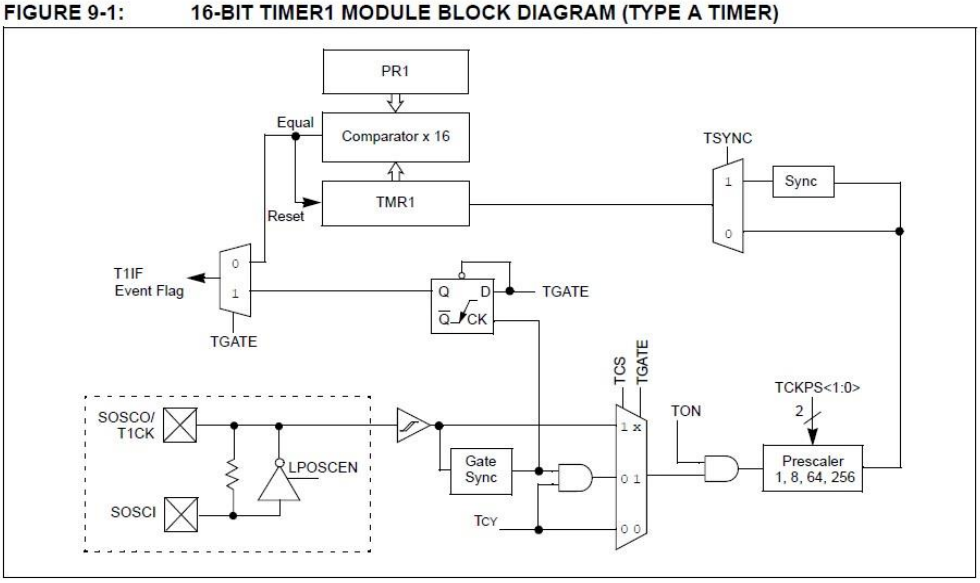
Trong vi điều khiển dsPIC40F4011 có tới năm bộ định thời (Timer) 16-bit. Trong đó các Timer có thể hoạt động riêng biệt, riêng hai Timer 2, 3 và hai Timer 4, 5 có thể kết hợp với nhau để trở thành một Timer 32 bit.

Timer 1 có thể hoạt động với nguồn tạo dao động tần số thấp 32KHz, và chế độ không đồng bộ với nguồn tạo dao động ngoài. Đặc điểm riêng biệt của Timer 1 đó là có thể dùng trong các ứng dụng thời gian thực. Timer 1 có thể hoạt động trong ba chế độ:

- Timer 16-bit: trong chế độ này, timer sẽ tăng sau mỗi chu kỳ lệnh đến khi giá trị của timer bằng giá trị của thanh ghi chu kỳ PR1 (Period Register) thì sẽ reset về ‘0’ và tiếp tục đếm.
- Counter đồng bộ 16-bit: trong chế độ này, timer sẽ tăng ở mỗi sườn lên của của xung nhịp ngoài mà được đồng bộ với pha của các xung nhịp trong. Timer tăng đến giá trị nằm trong thanh ghi PR1 thì dừng và reset timer về ‘0’ rồi tiếp tục đếm lên.

- Counter không đồng bộ 16-bit: khi hoạt động trong chế độ này, timer sẽ tăng dần sau mỗi sườn lên của xung nhịp bên ngoài tác động vào. Timer sẽ tăng dần đến khi giá trị của nó bằng thanh ghi PR1 thì bị reset về ‘0’ rồi lại tiếp tục đếm lên.

FIGURE 9-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM (TYPE A TIMER)



Hình 2.10 Sơ đồ khối Timer 1

2.2.4 Bộ chuyển đổi tương tự sang số ADC:

Vi điều khiển dsPIC30F4011 cung cấp bộ chuyển đổi tương tự sang số 10 bit với 10 kênh ngõ vào. Điện thế chuẩn có thể là nguồn cung cấp (Vdd/Vss) hoặc mức điện thế của các chân VREF+/VREF- . Bộ ADC có 6 thanh ghi

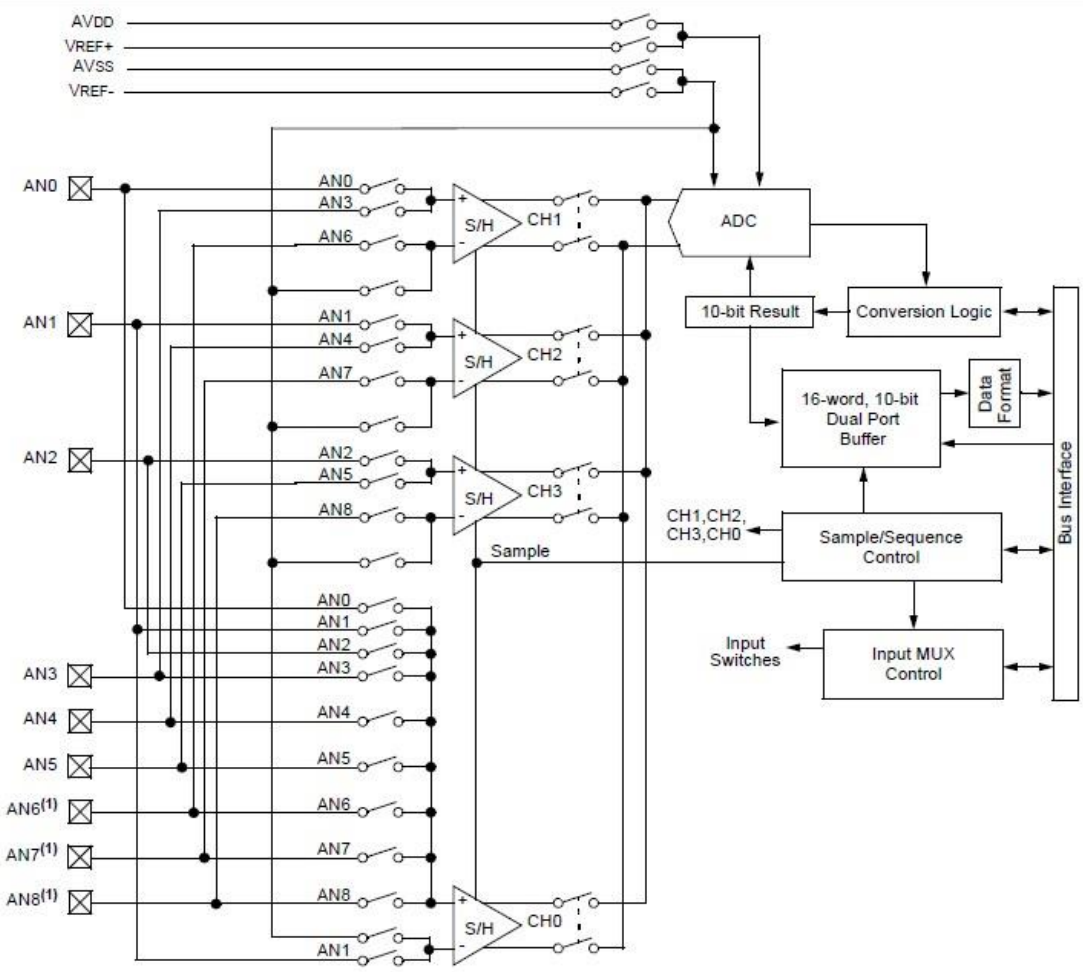
- Ba thanh ghi điều khiển: ADCON1, ADCON2, ADCON3 điều khiển hoạt động của ADC.
- Thanh ghi lựa chọn kênh vào để biến đổi ADCSH.
- Thanh ghi cấu hình ngõ vào là tương tự hay số ADPCFG.
- Thanh ghi lựa chọn quét ADCSSL.

DsPIC cũng có thể lấy mẫu đồng thời 2 kênh hoặc 4 kênh cùng lúc, lấy mẫu tuần tự, thời gian lấy mẫu và chuyển đổi có thể thiết lập trước, có thể kích chuyển đổi bằng một sự kiện do ngoại vi khác tạo ra. Kết quả chuyển đổi được lưu trong bộ đếm gồm 16 thanh ghi ADCBUF0 đến ADCBUFF.

Các bước tiến hành chuyển đổi ADC:

- Cấu hình chân là ngõ vào tương tự, chọn điện áp chuẩn.
- Chọn kênh cần chuyển đổi, chọn xung nhịp chuyển đổi
- Bật ADC hoạt động.
- Xóa cờ ngắt, chọn mức ưu tiên ngắt cho ADC.

- Bắt đầu lấy mẫu, đợi kết thúc lấy mẫu và bắt đầu chuyển đổi
- Đọc kết quả từ bộ đệm.



Hình 2.11 Sơ đồ khối ADC

2.2.5 Khối PWM điều khiển động cơ

DsPIC30F4011 có 6 chân phát PWM với 3 cặp tương ứng chuyên dùng cho điều khiển động cơ với các đặc điểm:

- 6 chân phát PWM với 3 bộ tạo chu kì.
- Độ phân giải 16 bit.
- Chế độ tạo xung tự do: thanh ghi đếm được đếm lên bằng giá trị trong PTPER thì PTMR bị reset về 0, sau đó tiếp tục đếm lên nếu PTEN còn bật.
- Chế tạo đơn xung: cũng giống như chế độ tạo xung tự do chỉ khác ở chỗ khi PTMR bằng PTPER thì bit PTEN sẽ bị xóa bằng phần cứng để ngăn bộ tao xung hoạt động tiếp.

- Chế độ tạo xung canh giữa (đếm lên/xuống): thanh ghi sẽ đếm lên khi bằng giá trị trong PTPER thì sẽ đảo hướng đếm bằng cách đảo bit PTDIR. Ngắt khi thanh ghi PTMR trở về 0.
- Chế độ tạo xung cạnh giữa cập nhật kép: ngắt xảy ra khi PTMR bằng 0 cũng như PTMR = PTPER. Chế độ này giúp cập nhật độ rộng xung trong chu kì kép cũng như tạo ra dạng sóng không đều khi cạnh giữa phù hợp với một số ứng dụng điều khiển.
- Các chân phát PWM có thể hoạt động độc lập hay giao hoán nhau.
- Có thanh ghi tạo thời gian chết DTCON, tránh trùng dẫn khi tạo xung kích bộ nghịch lưu.
- Có thể áp đặt giá trị logic ngõ ra thông qua thanh ghi OVDCON.
- Có thể tạo sự kiện kích chuyển đổi ADC.

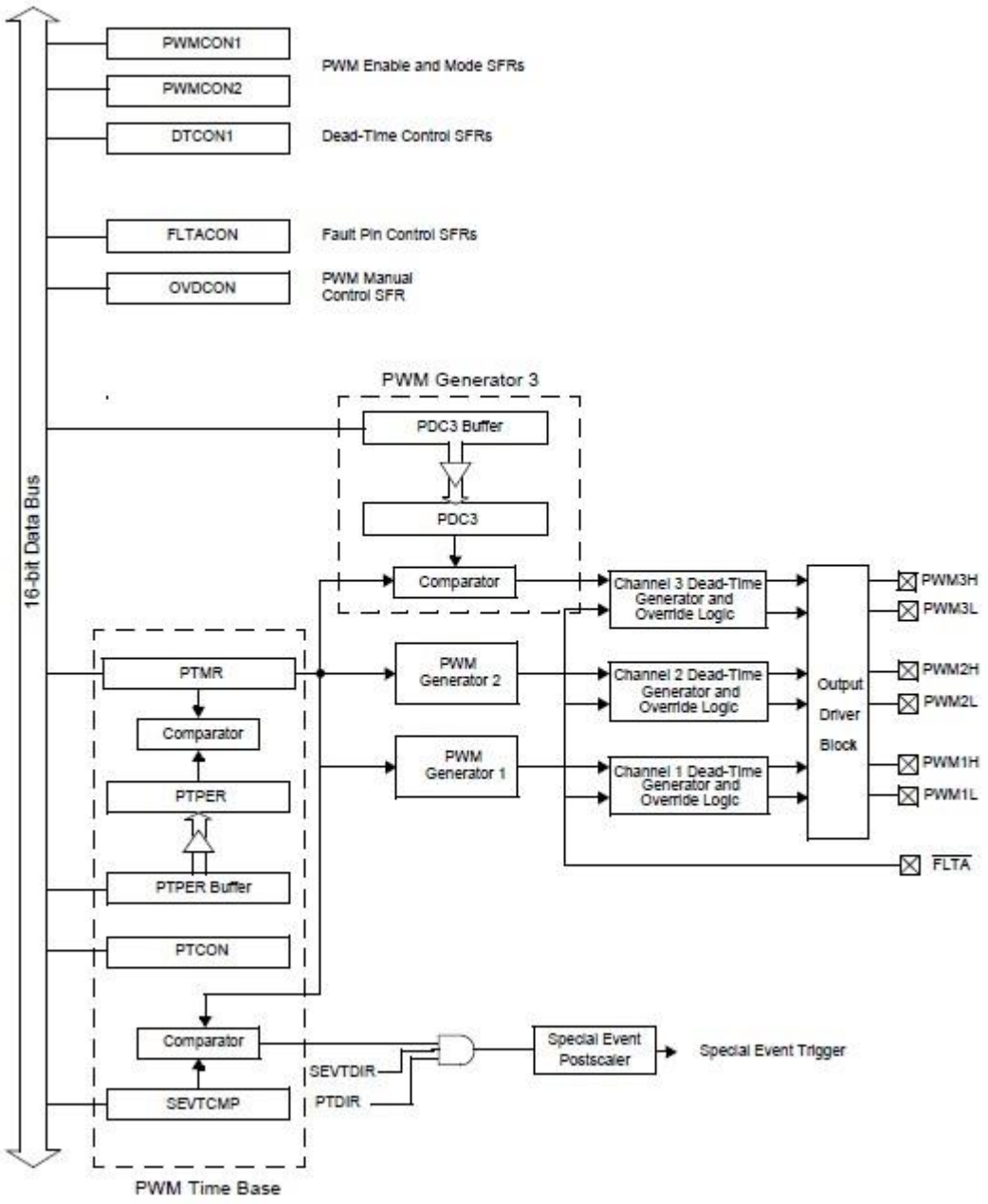
Công thức tính chu kì:

Trong chế độ tạo xung tự do hoặc đơn xung (xem [4]):

T_{PWM} = Tcy*(PTPER+1)*PTMR Prescale value (2.15)

Trong chế độ canh giữa (xem [4]):

T_{PWM} = 2*Tcy*(PTPER+1)*PTMR Prescale value (2.16)



Hình 2.12 Sơ đồ khối của PWM điều khiển động cơ.

2.3 Bộ nghịch lưu:

Có hai lựa chọn chính cho việc sử dụng đóng ngắt trong điều khiển động cơ đó là Mosfet hoặc IGBT. Cả hai loại này đều được điều khiển bằng áp, nghĩa là việc dẫn và ngưng dẫn của linh kiện được điều khiển bằng một nguồn áp thay vì dòng điện trong các bộ nghịch lưu sử dụng. Mosfet sử dụng ở các ứng dụng đòi hỏi tốc độ cao, tuy nhiên Mosfet không có khả năng chịu dòng điện cao. IGBT thích hợp với các ứng dụng ở tốc độ thấp, nhưng IGBT có khả năng chịu được dòng điện cao. IGBT có tần số đóng ngắt thấp hơn Mosfet nên đóng ngắt cũng nhiều hơn.

Yêu cầu đặt ra cho linh kiện sử dụng làm bộ nghịch lưu:

- Điện áp VDS (Mosfet) hay VCE (IGBT) >> VDC/2
- Dòng diện qua linh kiện lớn hơn dòng định mức của động cơ ≈10A ở nhiệt độ hoạt động.
- Chịu được tần số đóng ngắt cao.

Ở ứng dụng này ta dùng IGBT SGW25N120 để điều khiển đóng ngắt.

Điện áp Vec max	Dòng Ic max	Nhiệt độ hoạt động	Thời gian trì hoãn
1200VAC	25A	-55 => 150	10us

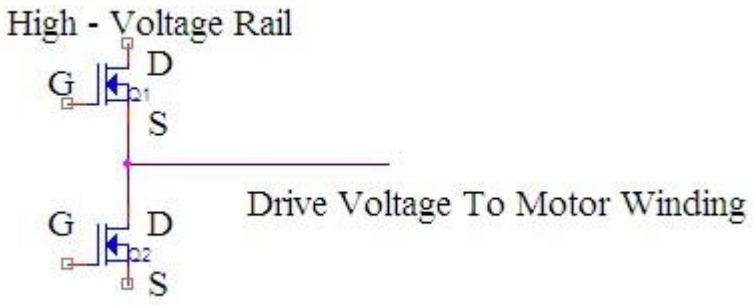
Bảng 2.2 Bảng giá trị IGBT

2.4 IC Lái

Các mạch phát ra tín hiệu để điều khiển mạch công suất dùng bán dẫn phải được cách ly về điện. Điều này có thể thực hiện bằng opto hoặc biến áp xung.

- Biến áp xung: gồm một cuộn sơ cấp và có thể có nhiều cuộn thứ cấp. Với nhiều cuộn dây phía thứ cấp, ta có thể kích đóng nhiều transistor mắc nối tiếp hoặc song song. Biến áp xung cần có cảm kháng nhỏ và đáp ứng nhanh. Trong trường hợp xung điều khiển có tác động kéo dài hoặc tần số thấp, biến áp xung sớm đạt trạng thái bão hòa và ngõ ra sẽ không phù hợp với yêu cầu điều khiển.
- Opto: gồm nguồn phát hồng ngoại bằng diot (IR – LED) và mạch thu dùng phototransistor. Do đó, opto thỏa mãn yêu cầu về điện, đồng thời đáp ứng tốt hơn biến áp xung.

Các mạch điều khiển động cơ dùng mạch nghịch lưu đều cần dùng tới mạch lái để điều khiển đóng ngắt Mosfet hay IGBT. Có hai thành phần cơ bản trong điều khiển đóng ngắt linh kiện công suất là điều khiển van phía cao (high side – Q1) và van phía thấp (low side – Q2).

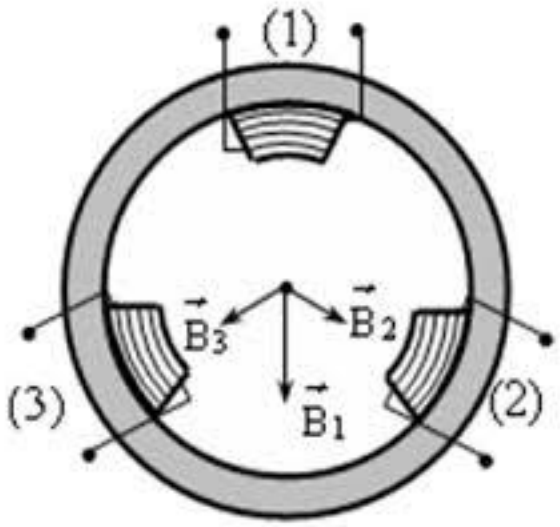


Hình 2.13 Sơ đồ điều khiển Mosfet.

Trong mạch nghịch lưu Q1 và Q2 luôn ở trạng thái làm việc đối nghịch nhau. Khi Q1 mở thì Q2 đóng và ngược lại. Khi Q1 đang ở trạng thái OFF chuyển sang trạng thái ON => chân S (MOSFET) hay chân E (IGBT) của Q1 chuyển từ ground sang điện áp cao (high voltage rail). Do đó muốn kích Q1 tiếp tục ON thì phải tạo điện áp kích VGS1 có giá trị VGS1= VSQ1 + ΔV. Trong khi đó tín hiệu ra của vi xử lý điều khiển đóng ngắt các khóa chỉ có giá trị điện áp +5V (so với ground). Nên cần phải có mạch lái để tạo trôi áp và cách ly trong việc đóng ngắt phía van Q1. Tuy nhiên đối với Q2 thì chân S được nối ground, do đó điện áp kích VGS2 chỉ cần có giá trị ΔV. Do đó việc đóng ngắt khóa low side (Q2) được điều khiển dễ dàng hơn. ΔV là giá trị điện áp cần thiết để kích Q1 hay Q2 dẫn. Đối với MOSFET và IGBT ΔV có giá trị từ 10 đến 15 (V).

Trong ứng dụng này, ta sử dụng ic TLP250 vừa cách quang vừa điều khiển dẫn ngắt IGBT.

2.5 ĐỘNG CƠ BA PHA:



Khi nam châm điện quay(n1 vòng/phút), làm đường sức từ quay cắt qua các cạnh của khung dây cảm ứng gây nên sức điện động E trên khung dây. Sức điện động E sinh ra dòng điện I chạy trong khung dây. Vì dòng điện I nằm trong từ trường nên khi từ trường quay làm tác động lên các khung dây một lực điện từ F. Lực điện từ này làm khung dây chuyển động với tốc độ n vòng/phút. Vì n<n1 nên gọi là không đồng bộ.

Động cơ KĐB 3 pha có cấu tạo gồm: dây quấn 3 pha phía stator, rotor là một bộ dây quấn 3 pha có cùng số cực trên lõi thép stator. Khi stator được cung cấp bởi nguồn 3 pha cân bằng tần số f, từ trường quay được tính theo công thức: $\omega = 2\pi f/p$ với p là số cặp cực

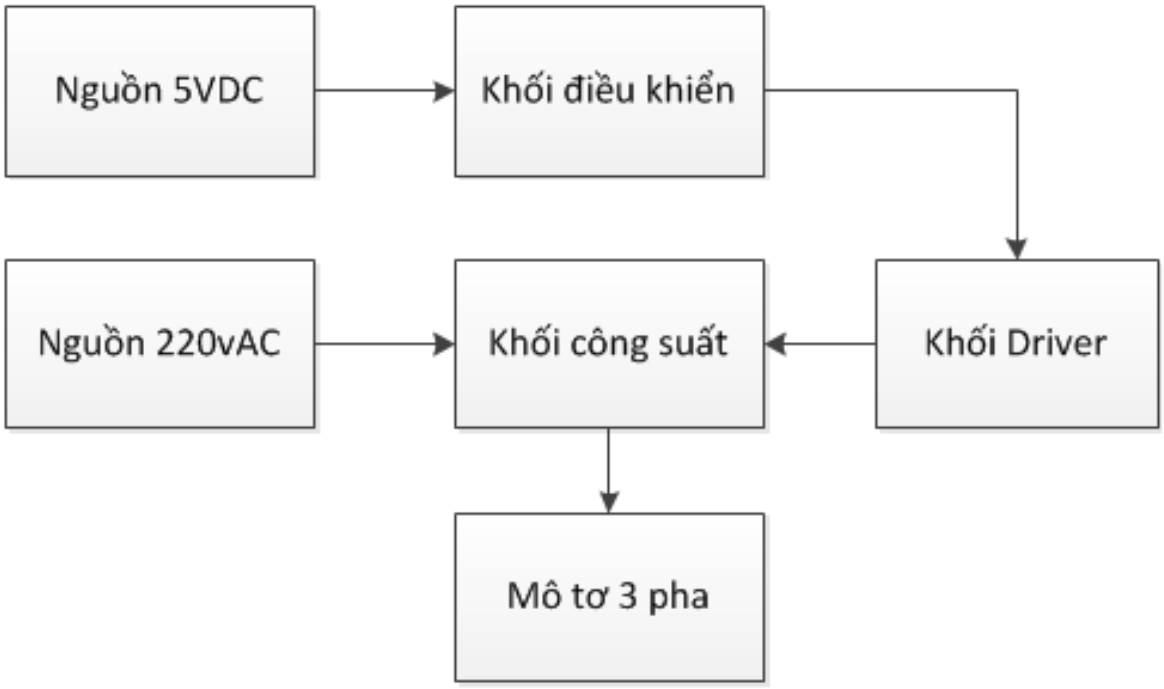
Trong ứng dụng này, ta dùng động cơ 3 pha công suất nhỏ. Động cơ có công suất 120W, điện áp cung cấp 220V-50Hz, tốc độ danh định 1400 vòng/phút.

Hình 2.14 Động cơ 3 pha.

CHƯƠNG 3 TÍNH TOÁN VÀ THIẾT KẾ

3.1 THIẾT KẾ PHẦN CỨNG

3.1.1 Sơ đồ khối của hệ thống



Hình 3.1 Sơ đồ khối hệ thống.

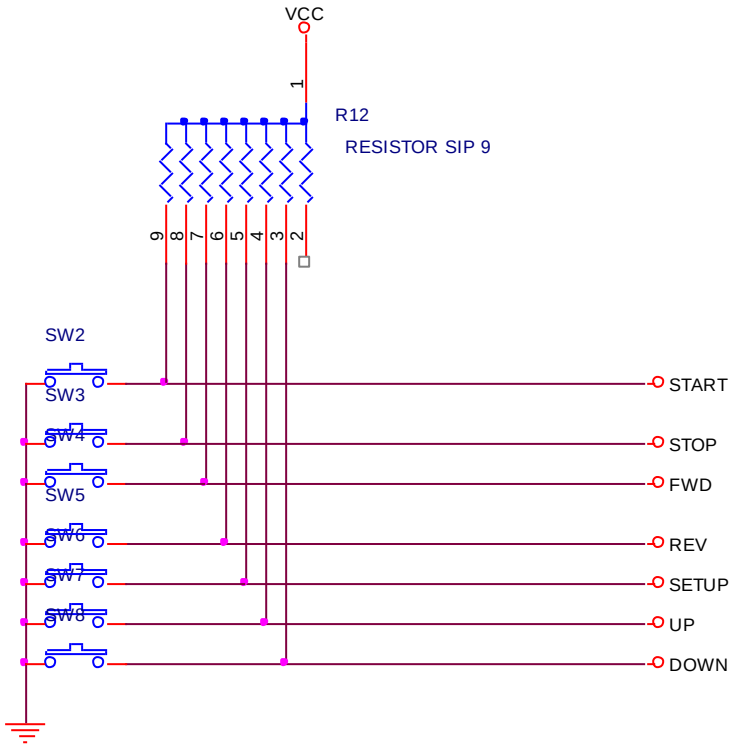
3.1.2 Mạch điều khiển

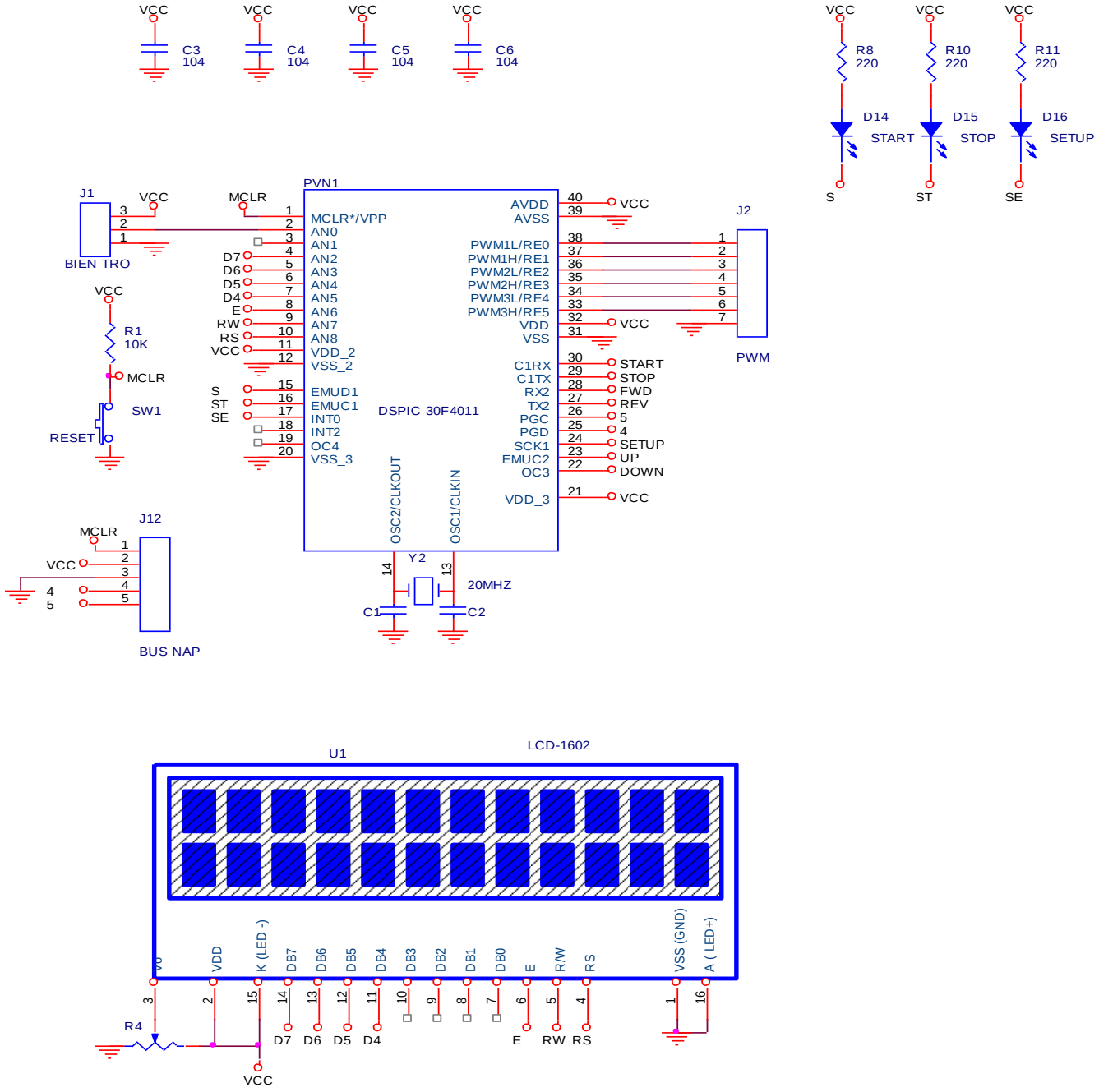
Mạch vi điều khiển có thể coi là trung tâm của hệ thống. Tại đây, vi điều khiển đọc các giá trị, biến đổi, tính toán, điều chỉnh và xuất ra tín hiệu đóng ngắt các IGBT tạo điện áp điều khiển động cơ.

Mạch vi điều khiển được thiết kế riêng, tất cả các chân được nối ra các port nối bus nhằm tăng tính linh hoạt cho vi điều khiển khi muốn thay đổi kết nối. Mạch được cung cấp điện áp 5V từ bộ nguồn bên ngoài. Vi điều khiển hoạt động với tần số thạch anh 20MHz, có nút reset và diode chống dòng ngược.

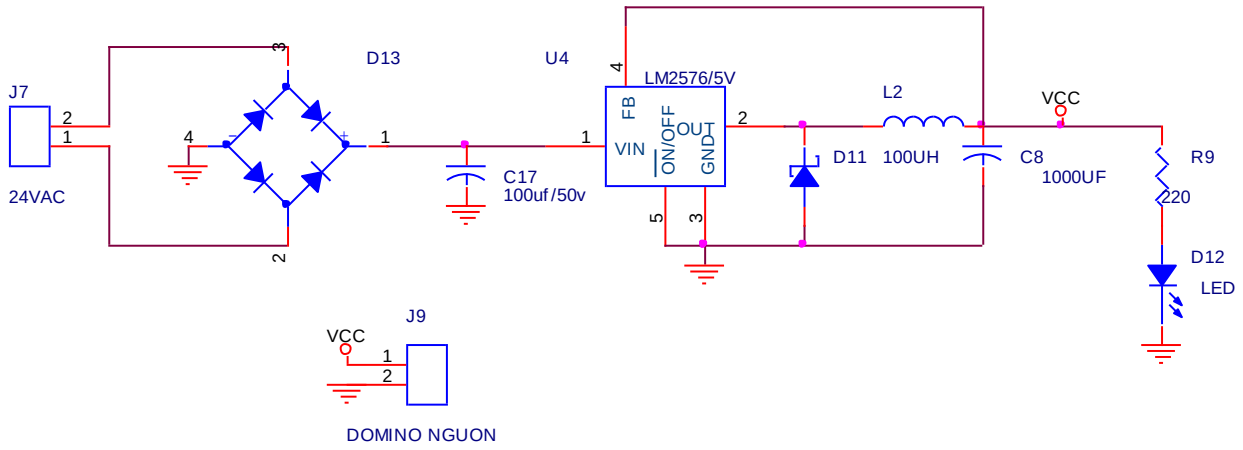
Vi điều khiển được kết nối port như sau:

- PORTE: chân ra của 6 kênh phát PWM.
- PORTA: AN2, AN3, AN4, AN5, AN6, AN7, AN8: kết nối với LCD.
- PORTB: C1RX, CTX1, RX2, TX2, SCK1, EMUC2, OC3 kết nối với các phím.
- AN0: kết nối với biến trở.





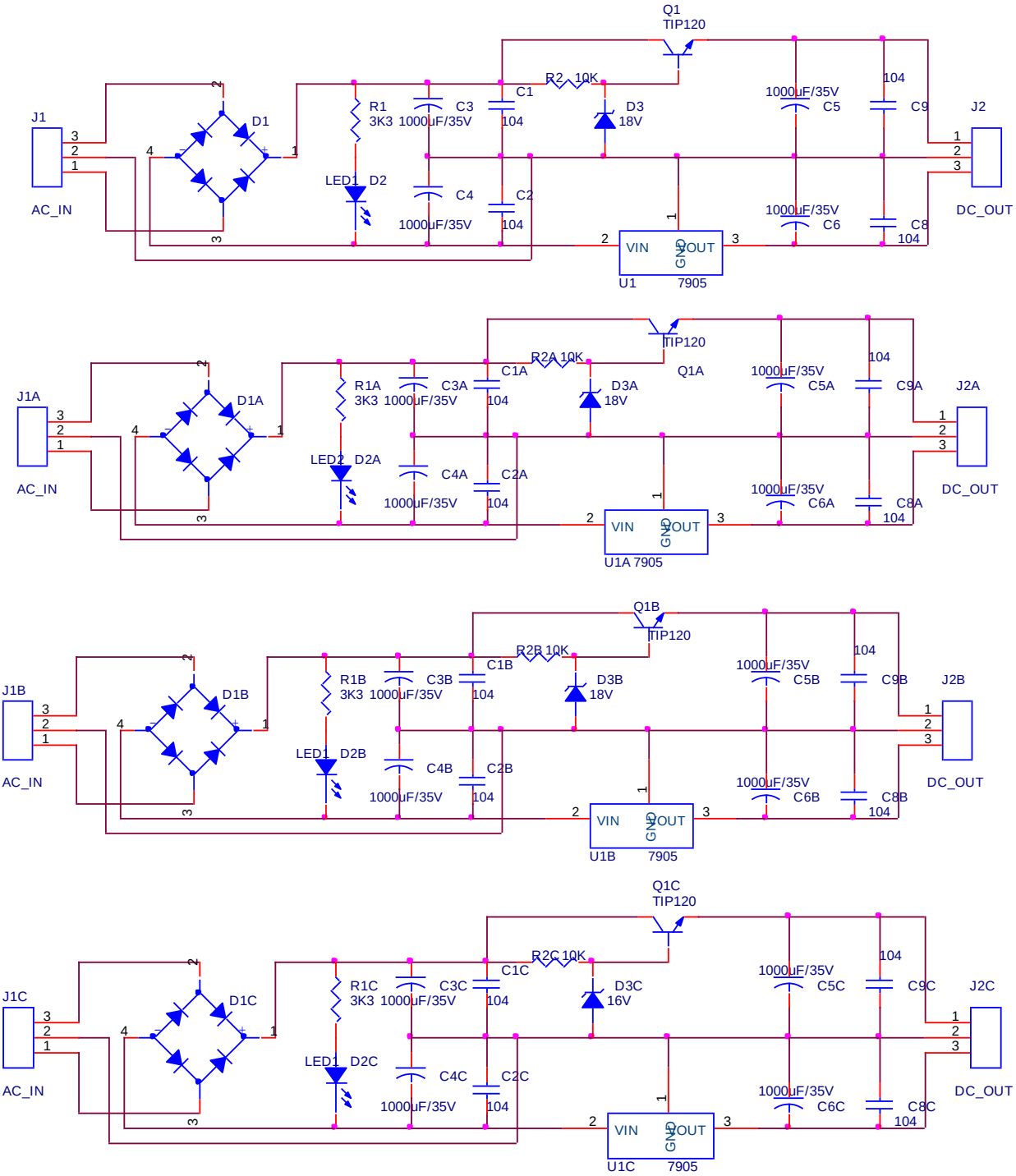
Hình 3.2: Sơ đồ nguyên lý mạch điều khiển



Hình 3.2 Sơ đồ nguyên lí mạch cấp nguồn vi điều khiển

3.1.3 Nguồn kích:

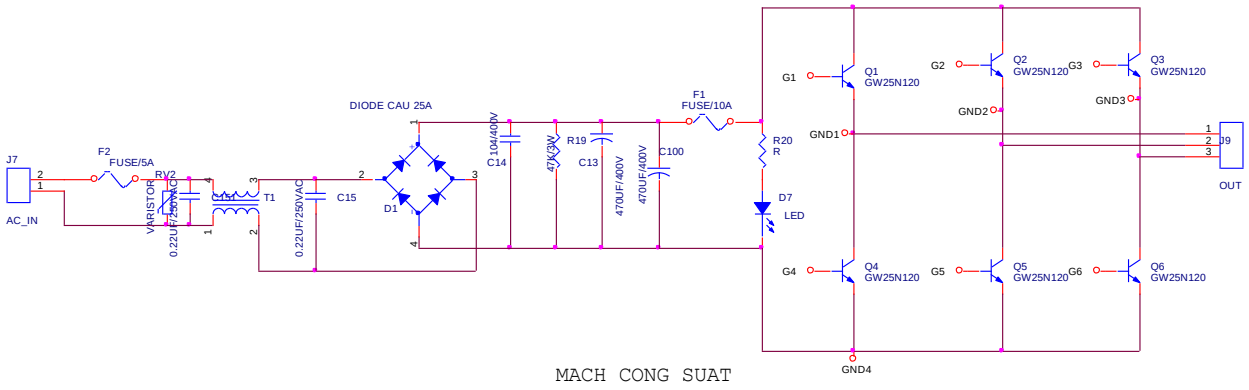
Mạch nguồn kích gồm 4 nguồn kích, trong đó 3 nguồn kích dùng để kích cho 3 IGBT không chung GND. Điện áp vào mỗi nguồn kích là 220VAC qua biến áp tạo điện áp + 15VDC, sau đó qua mạch chỉnh lưu dùng cầu diode. Ngõ ra thứ nhất được qua mạch ổn áp tạo điện áp +15VDC. Ngõ ra còn lại được qua ic 7905 nhằm hạ áp xuống -5VDC. Nguồn kích còn lại dùng để cung cấp điện áp cho 3 IGBT còn lại có chung GND.



Hình 3.3 Sơ đồ nguyên lí mạch nguồn kích.

3.1.4 Mạch công suất:

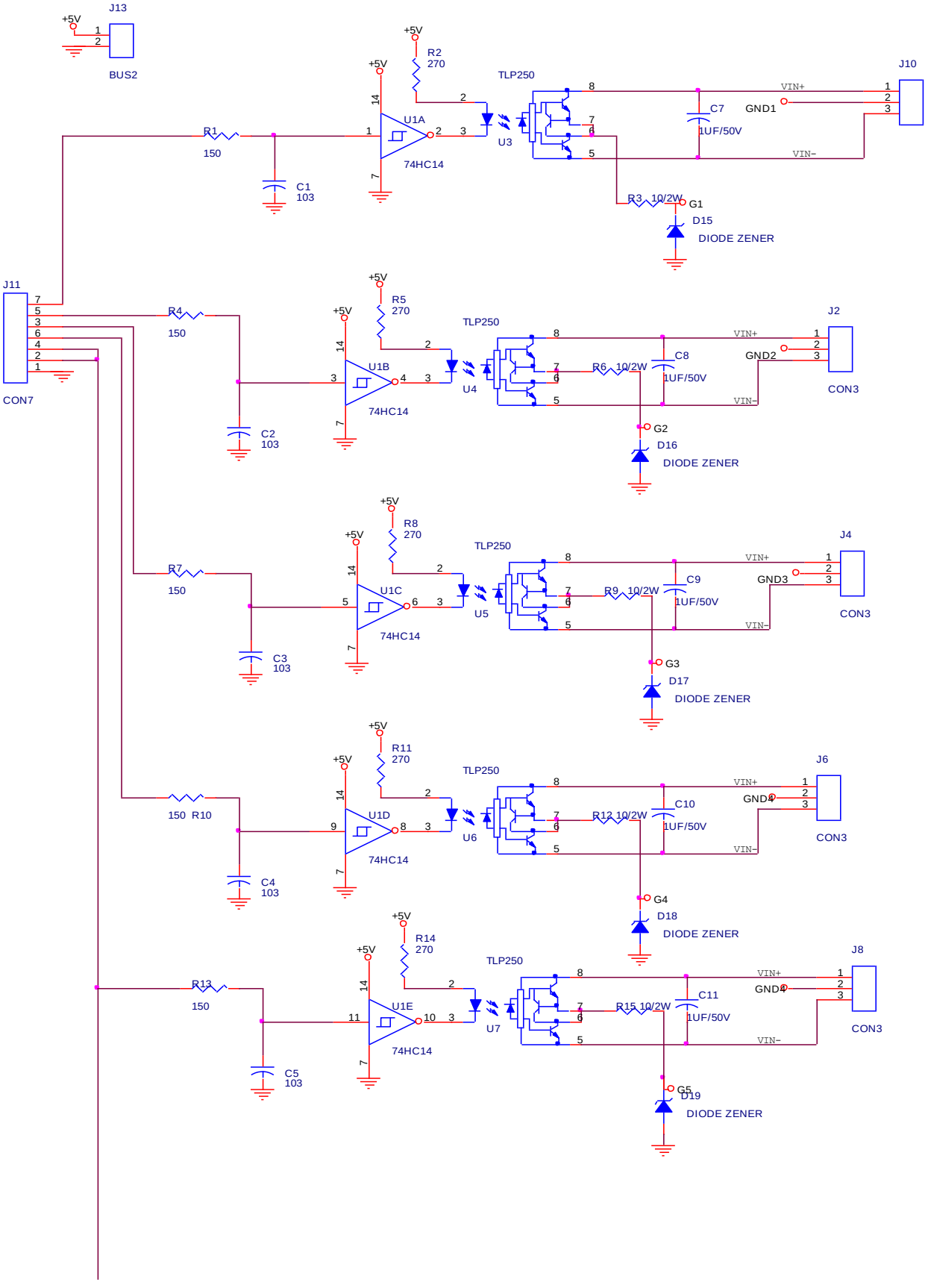
Điện áp AC_in vào 220VAC qua chỉnh lưu cầu có tụ lọc tạo điện áp 310VAC cung cấp cho IGBT. Ngõ vào của mạch là tín hiệu kích từ IC lái, đầu ra nối với 3 đầu dây của động cơ. IGBT có thể rất nóng trong lúc làm việc nên phải có thêm tản nhiệt kèm theo.

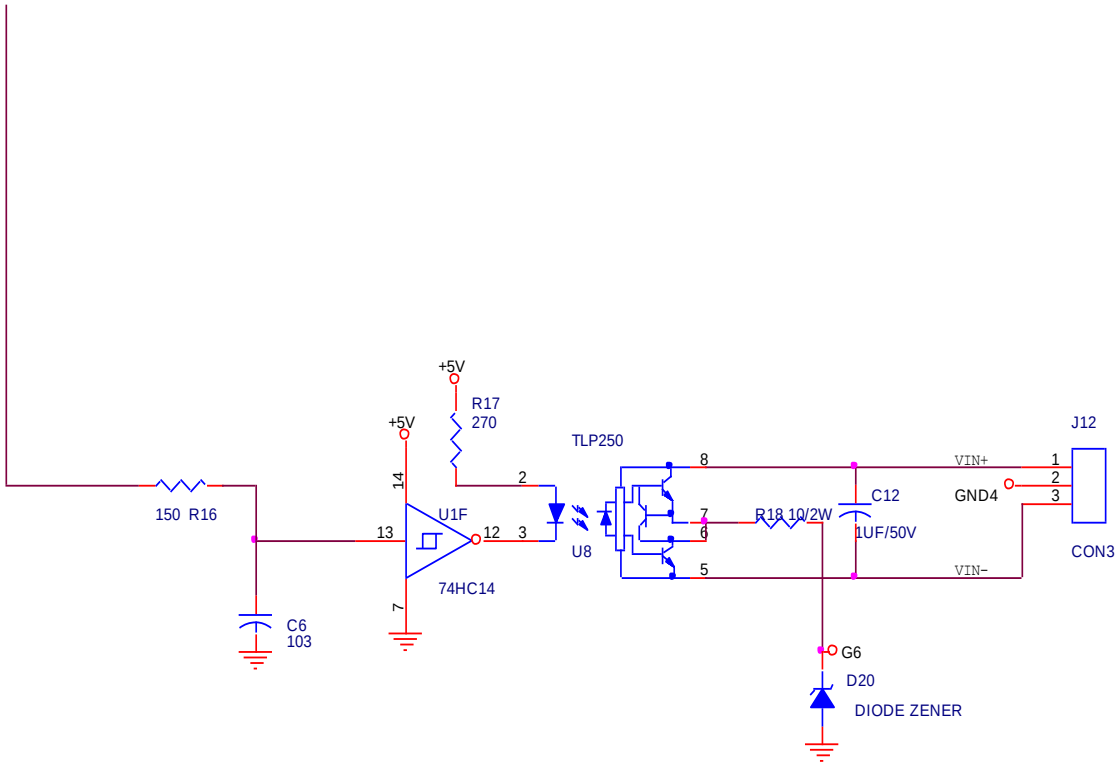


Hình 3.4 Sơ đồ nguyên lí kích mạch công suất.

3.1.5 Mạch Driver:

Mạch gồm 6 kênh điều khiển cho từng IGBT, ngõ vào lấy tín hiệu PWM từ vi điều khiển qua opto TPL250 để điều khiển IGBT.


J13
BUS2
+5V
R1
150
C1
103
U1A
74HC14
R2
270
TLP250
U3
J10
C7
1UF/50V
GND1
VIN+
VIN-
R3 10/2W
G1
D15
DIODE ZENER
J11
CON7
R4
150
C2
103
U1B
R5
270
U4
R6 10/2W
C8
1UF/50V
GND2
J2
CON3
G2
D16
DIODE ZENER
R7
150
C3
103
U1C
R8
270
U5
R9 10/2W
C9
1UF/50V
GND3
J4
CON3
G3
D17
DIODE ZENER
R10
150
C4
103
U1D
R11
270
U6
R12 10/2W
C10
1UF/50V
GND4
J6
CON3
G4
D18
DIODE ZENER
R13
150
C5
103
U1E
R14
270
U7
R15 10/2W
C11
1UF/50V
GND4
J8
CON3
G5
D19
DIODE ZENER



Hình 3.5: Sơ đồ nguyên lý mạch Driver.

CHƯƠNG 4: THI CÔNG HỆ THỐNG

4.1 Giới thiệu:

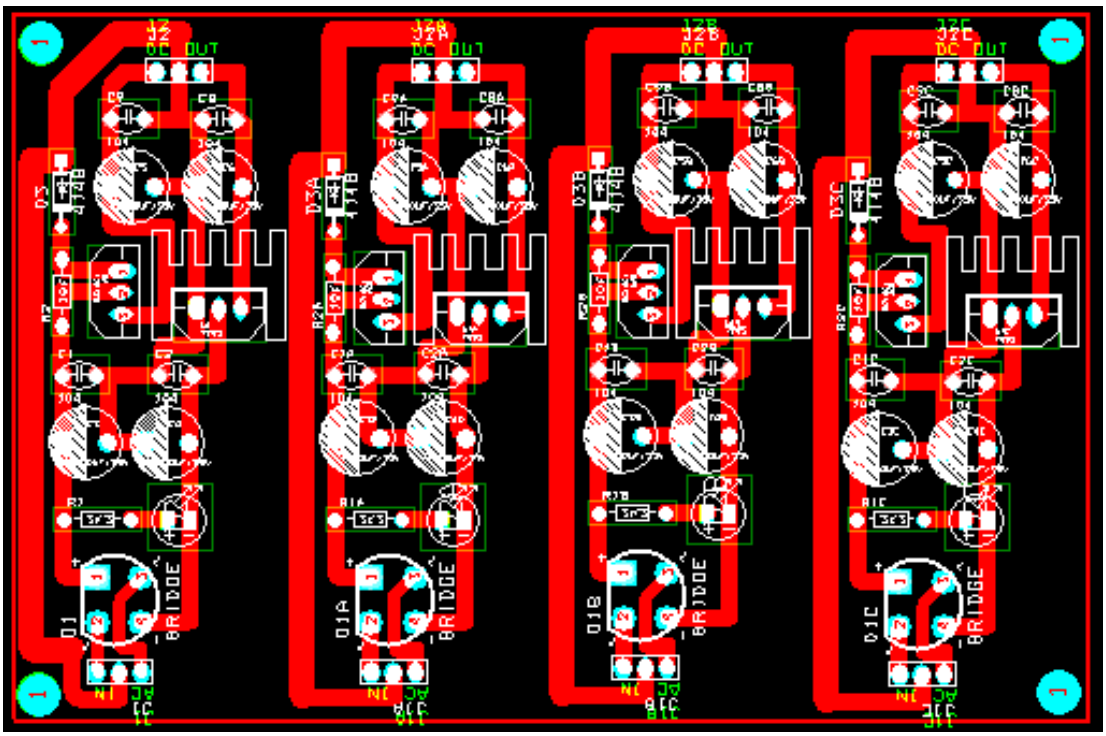
Hệ thống điều khiển động cơ không đồng bộ 3 pha theo phương pháp SPWM gồm 2 phần:

- Thiết kế thi công nguồn kích, mạch công suất, mạch điều khiển.
- Kiểm tra dạng sóng ngõ ra của mạch.

4.2 Thi công hệ thống và kết quả khác:

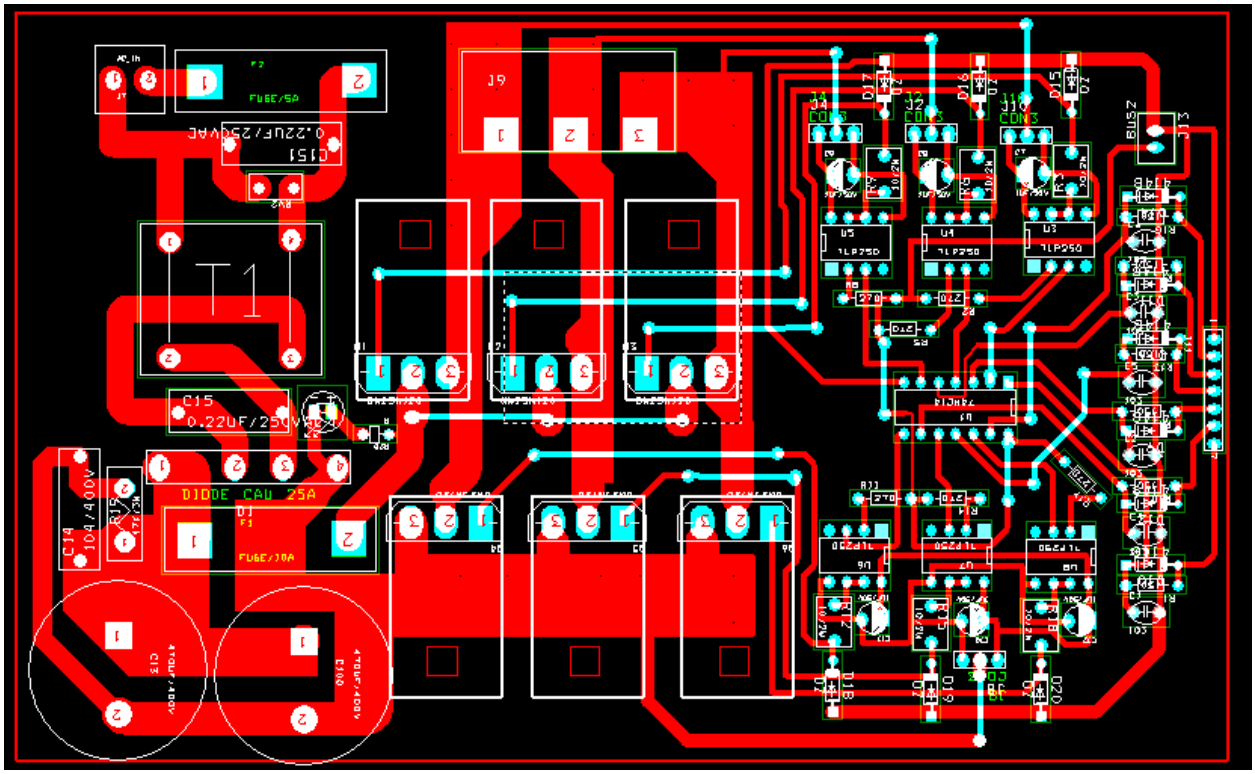
4.2.1 Thi công bo mạch:

- Thi công nguồn kích:



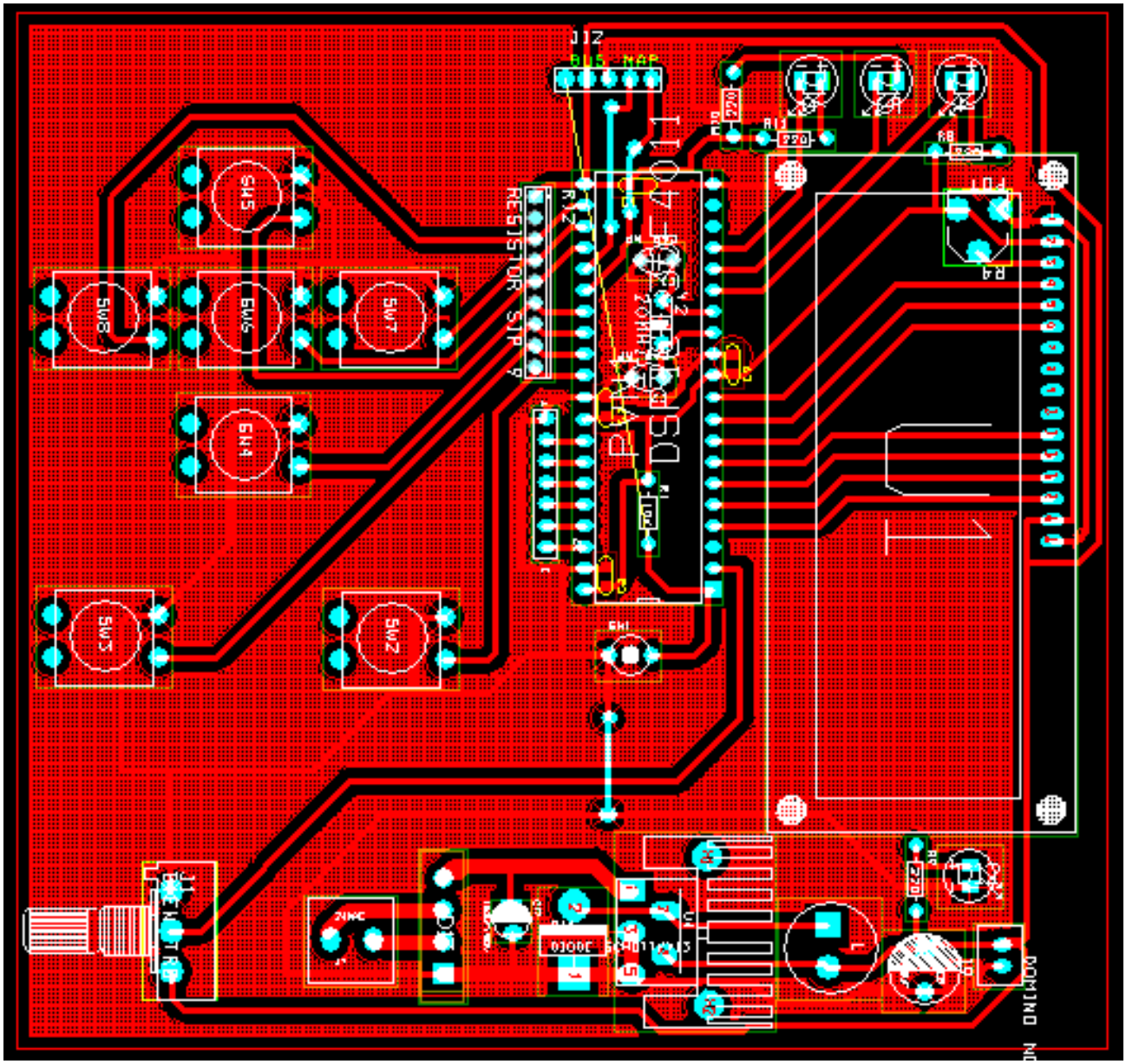
Hình 4.1 Mạch nguồn kích.

-Thi công mạch công suất:



Hình 4.2 Mạch công suất.

-Thi công mạch điều khiển:



Hình 4.3 Mạch điều khiển.

4.2.2 Lắp ráp và kiểm tra:

- Lắp ráp module nguồn kích: Tiến hành lắp ráp các linh kiện cho module nguồn. Tiến hành đo và kiểm tra nguồn vào và ra xem đúng sai, cân chỉnh cho đúng.

- Láp ráp module công suất: lắp ráp các linh kiện và tản nhiệt cho module công suất.

- Lắp ráp module điều khiển: láp ráp các linh kiện cho module điều khiển.

4.3 Lập trình hệ thống:

Với 3 lựa chọn ta có thể khởi động động cơ với 3 chế độ:

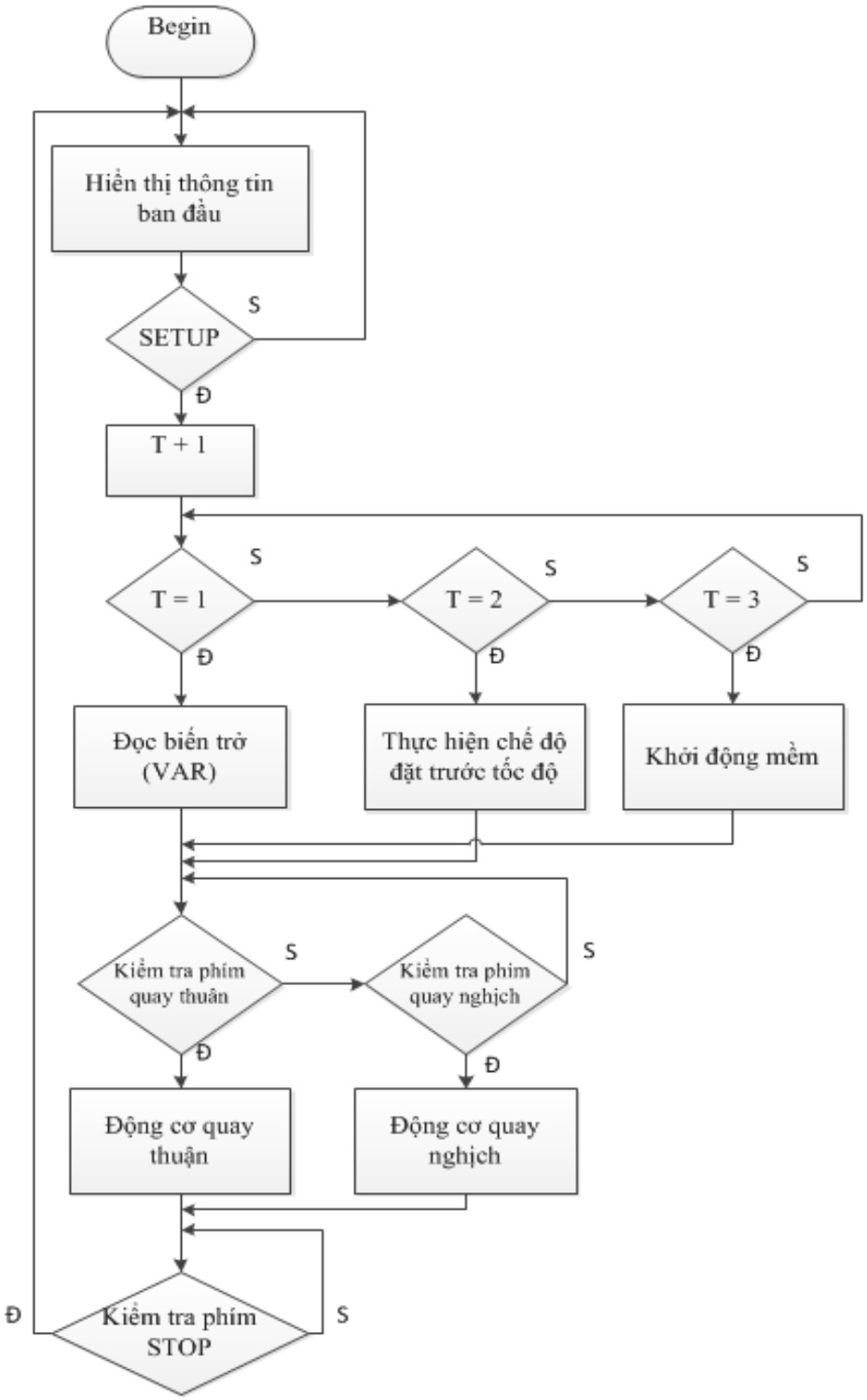
Chế độ thứ 1: Động cơ chạy với tần số đọc từ biến trở.

Chế độ thứ 2: Động cơ chạy chế độ đặt trước bằng bàn phím.

Chế độ thứ 3: Khởi động mềm.

Ở mỗi chế độ ta có thể thay đổi chiều động cơ bằng phím nhấn.

Lưu đồ chương trình:



Hình 4.4 Lưu đồ giải thuật.

4.4 Phần mềm lập trình cho chương trình:

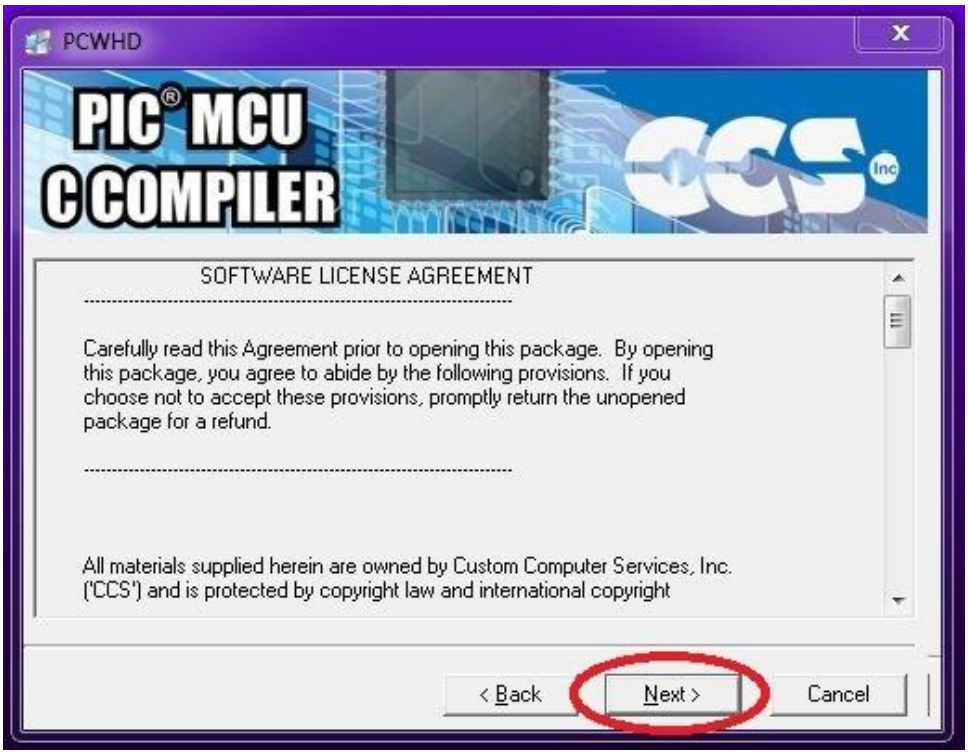
Dùng phần mềm CCS để lập trình cho dsPIC.
Hướng dẫn cài đặt và sử dụng phần mềm CCS lập trình cho dsPIC.

Để tải phần mềm ta truy cập link:
http://www.mediafire.com/download/9ax9cre15c954ov/Pic_CCS5.025.rar để tải phần mềm.

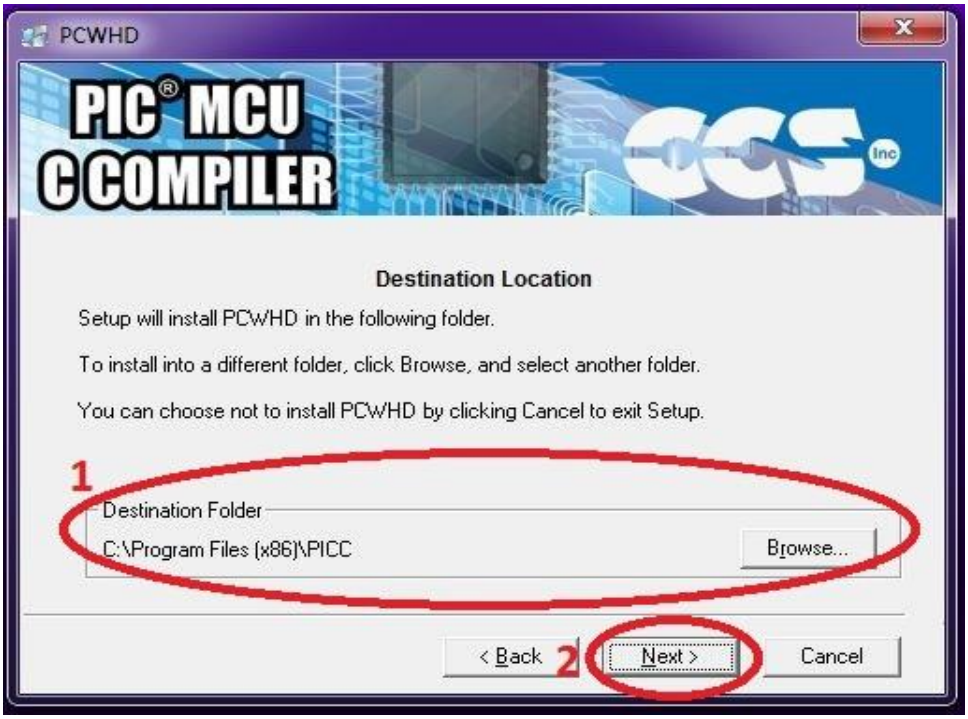
Sau khi tải và giải nén ra chúng ta sẽ thấy thư mực Pic CCS5.025, trong thư mục này có chứa hai file: file cài đặt phần mềm pcwhdupd và file crack phần mềm SCG-PCWHD-keygen.

Để cài đặt, chúng ta chạy file Pic CCS5.025. Hộp thoại cài đặt xuất hiện, nhấn Next để tiếp tục.

Tiếp tục nhấn Next



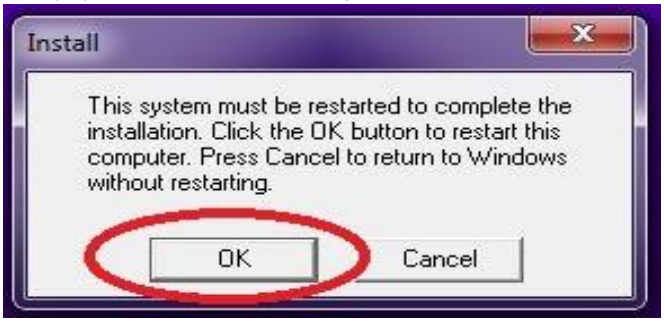
Chọn đường dẫn cài đặt và nhấn Next



Tiếp tục nhấn Next

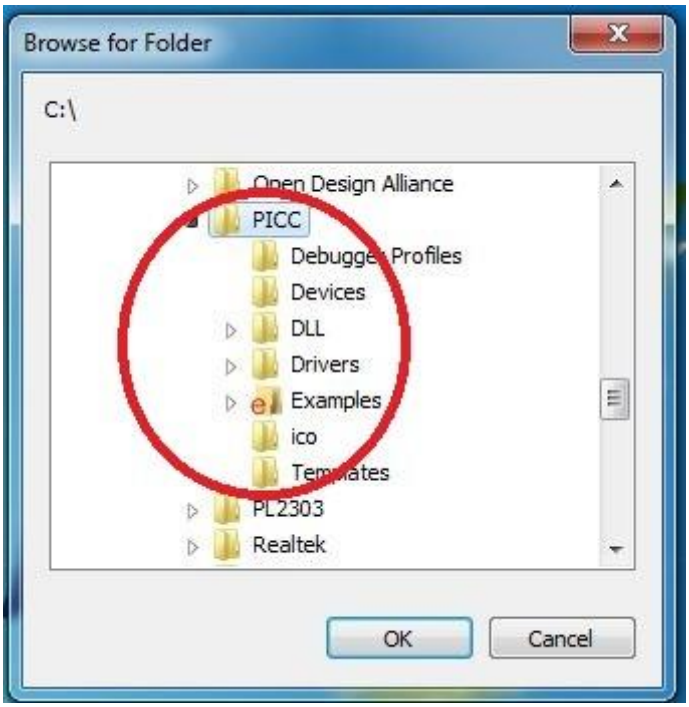
Chờ một lúc sau đó finish chương trình.

Máy yêu cầu khởi động lại, ta nhấn OK để khởi động lại, sau đó tiến hành Crack.



Chạy file crack SCG-PCWHD-keygen. Hộp thoại crack xuất hiện, điền đầy đủ thông tin sau đó nhấn GenerateCRG

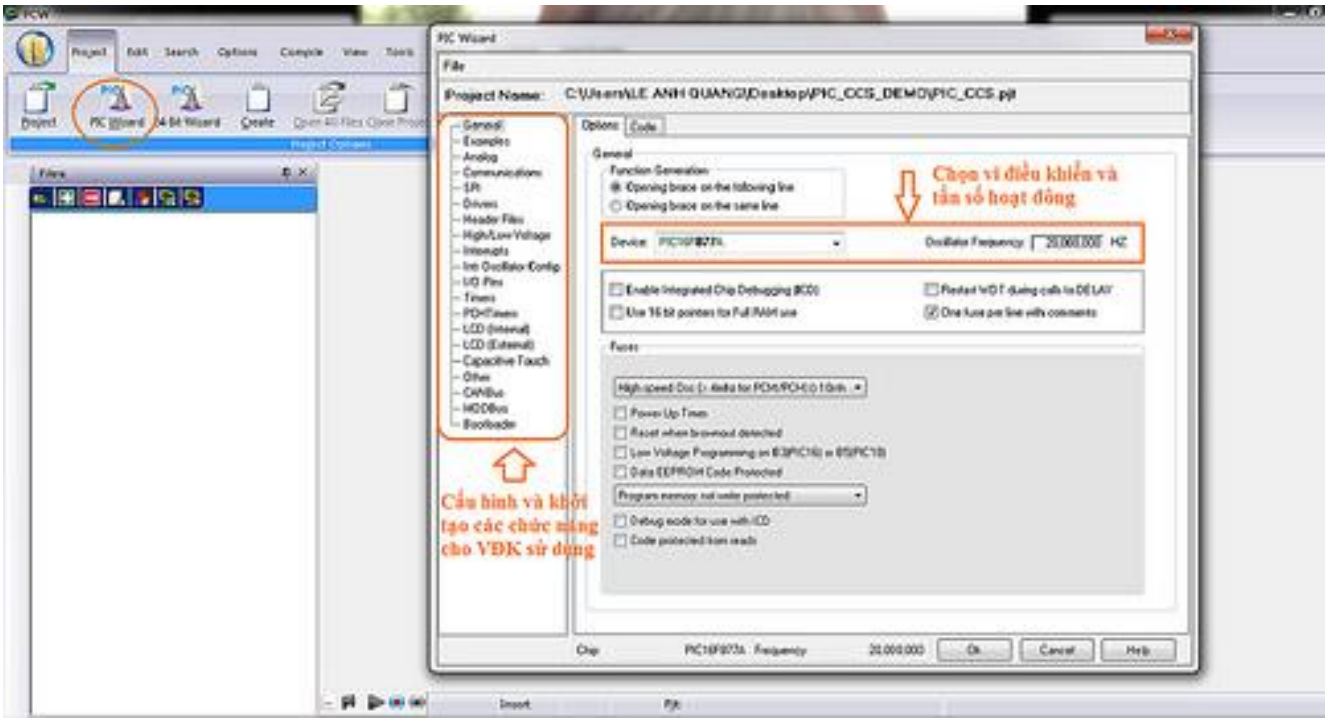
Một hộp thoại xuất hiện yêu cầu chọn nơi lưu file CRG được tạo ra, ở đây ta trỏ tới thư mục cài đặt PICC ở trong ổ C luôn. Sau đó nhấn OK để kết thúc.



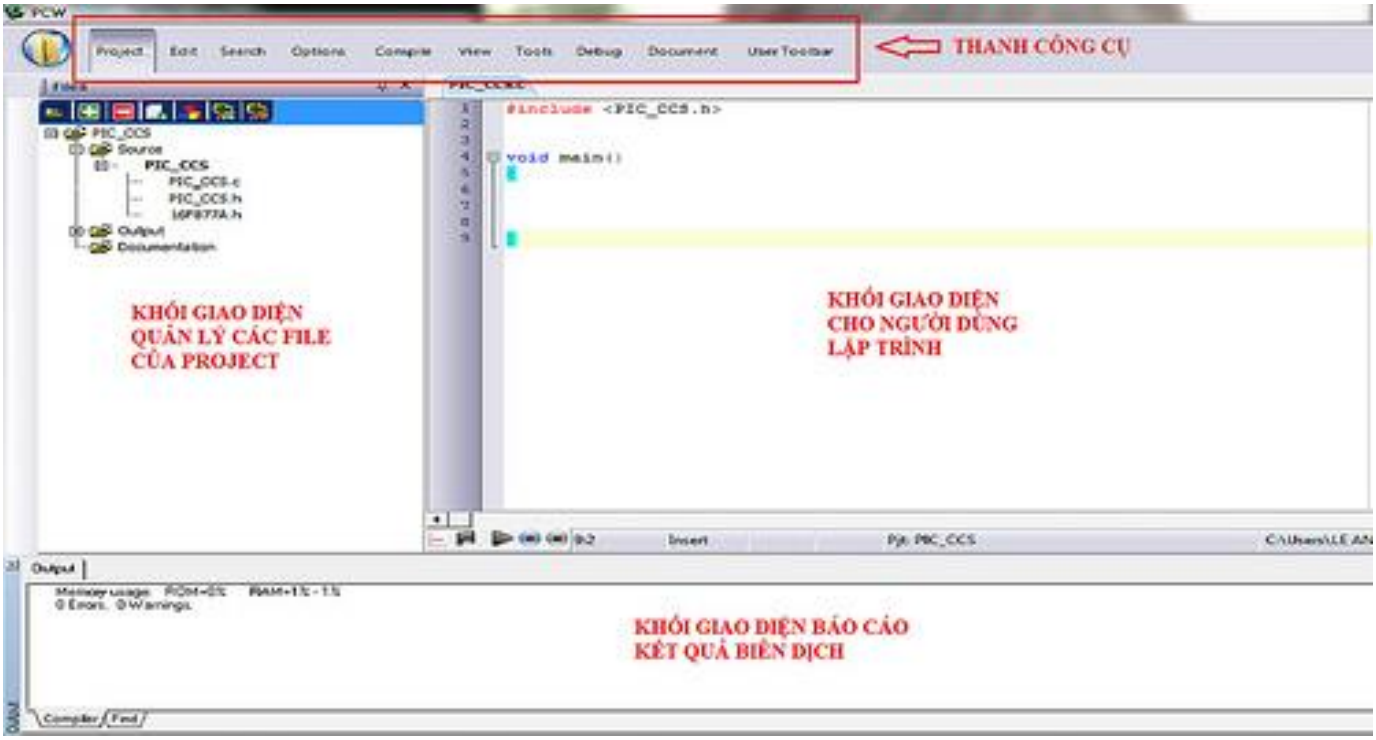
Như vậy là chúng ta đã hoàn thành phần cài đặt và crack cho phần mềm CCS.

Tiếp theo là khởi tạo một Project trong CCS.

Ở giao diện CCS, chọn Project/Pic Wizard. Sau đó lưu lại, cửa sổ giao diện PIC Wizard sẽ hiện ra. Tại đây, ta có thể thiết lập cấu hình chức năng cho các dòng PIC/DSPIC để đáp ứng yêu cầu của Project.



Sau khi hoàn tất nhấn OK, giao diện cho lập trình Project xuất hiện.



Tại đây sẽ gồm các khối chức năng như:

- Khối thanh công cụ: giúp hỗ trợ cho người dùng có thể dễ dàng tìm kiếm, chỉnh sửa, debug...Project.

- Khốigiao diện Files: giúp người dùng quản lý tất cả các file chứa trong Project của mình.

- Khối giao diện dành cho người dùng lập trình.

- Khối giao diện hiển thị kết quả quá trình biên dịch.

Sau khi viết chương trình hoàn thành, ta có file hex. Tiếp theo ta nạp file hex cho vi điều khiển. Để nạp chương trình cho vi điều khiển ta dùng mạch nạp Burn E, để sử dụng mạch nạp ta dùng phần mềm nạp Burn_E beta.
- Burn E là mạch nạp đa năng của người Việt do Pduytech.com phát triên, mạch có thể nạp cho các dòng MCU là PIC, DSPic, AT89S. Software dễ sử dụng, có thể nạp Firmware tự động thông qua cổng usb 2.0.

Hình 4.5 Mạch nạp Burn E.

4.6 Viết tài liệu hướng dẫn sử dụng, thao tác:

Bước 1: Cấp nguồn cho mạch, điện áp cung cấp là 220VAC, đèn báo mạch điều khiển và mạch kích sáng. Sau đó gạt *Switch* sang **ON** để cung cấp nguồn cho bộ công suất. Lúc này đèn nguồn báo mạch công suất được cung cấp điện sáng.

Bước 2: Nhấn *Setup* để lựa chọn chế độ, có 3 chế độ để lựa chọn:

- Chế độ 1: Ở chế độ này động cơ được vận hành bởi biến trở.
- Chế độ 2: Chế độ đặt tần số bằng phím nhấn.
- Chế độ 3: Khởi động mềm.

Lưu ý ở cả 3 chế độ ta đều có thể đảo chiều của động cơ.

Bước 3: sau khi lựa chọn chế độ ta nhấn Star, để khởi động chế độ.

- *Ở chế độ 1*: Nhấn *Setup* để chọn chế độ 1, nhấn *Start* để bắt đầu, nhấn *Forw/Rev* để chọn chiều quay của động cơ. Sau đó chỉnh biến trở để thay đổi tốc độ của động cơ. Nhấn *Stop* để dừng động cơ.

- *Ở chế độ 2*: Nhấn *Setup* để chọn chế độ 2, nhấn *Up/Down* để chọn tần số mong muốn. Sau đó nhấn *Start* để khởi động cơ, nhấn *Forw/Rev* để lựa chọn chiều quay của động cơ, động cơ bắt đầu chạy với tốc độ được đặt trước. Nhấn *Stop* để dừng động cơ.

- *Ở chế độ 3*: Nhấn *Setup* để chọn chế độ 3, ở đây hệ thống yêu cầu lựa chọn tần số hoạt động cho động cơ, nhấn *Up/Down* để cài đặt tần số mong muốn. Nhấn *Setup* để tiếp tục, hệ thống tiếp tục yêu cầu thời gian để khởi động, nhấn *Up/Down* để lựa chọn thời gian (đơn vị giây), nhấn *Start* để khởi động, ở đây hệ thống yêu cầu chọn chiều quay động cơ, nhấn *Forw/Rev* để lựa chọn. Động cơ bắt đầu khởi động với tần số và thời gian cài đặt. Nhấn *Stop* để dừng động cơ.

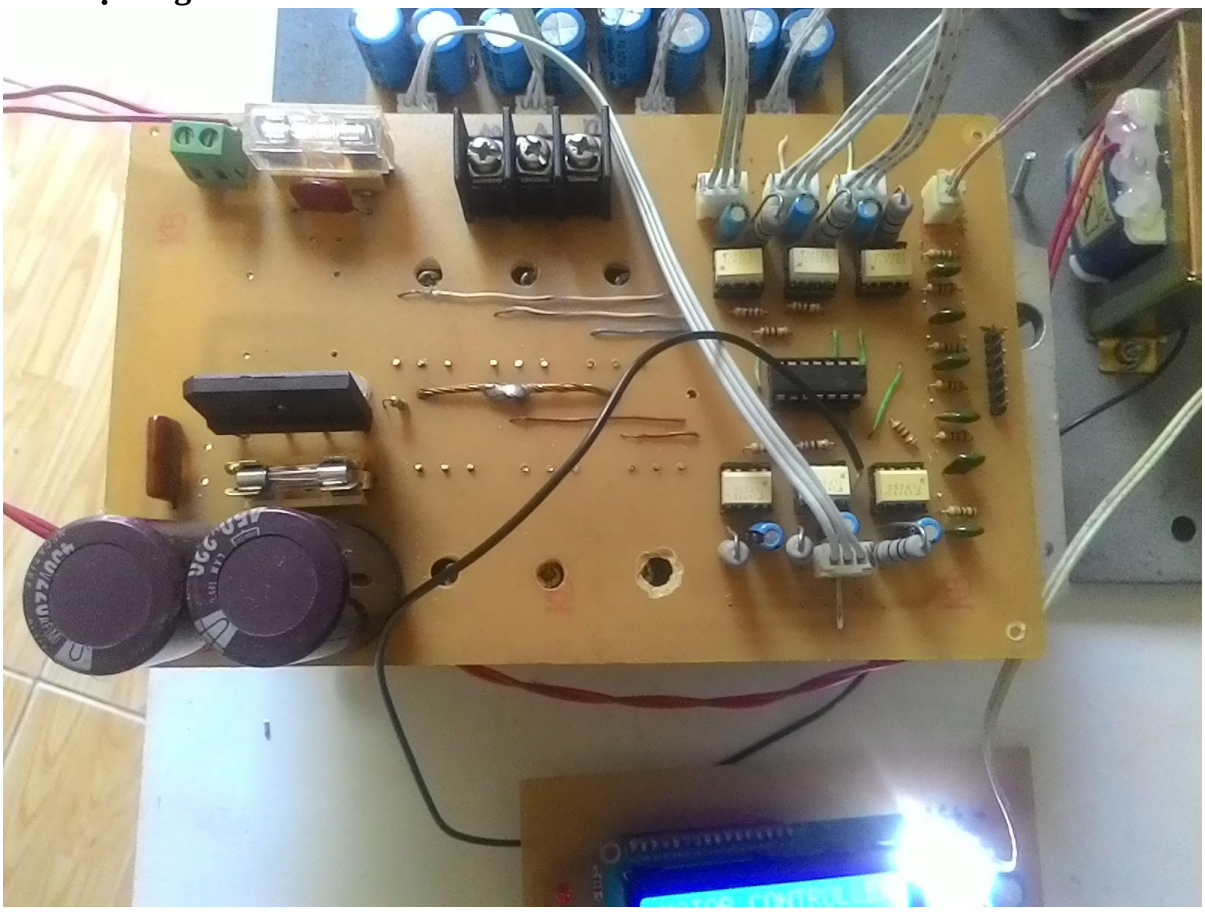
Bước 4: Sau khi sử dụng, muốn dừng hoạt động của độg cơ ta cần phải nhấn *Stop* để dừng động cơ. Sau đó tắt bộ công suất bằng *Switch* gạt sang **OFF**. Ngắt điện toàn hệ thống. Đèn báo nguồn mạch công suất tắt.

CHƯƠNG 5 KẾT QUẢ VÀ NHẬN XÉT ĐÁNH GIÁ

Sau thời gian tìm hiểu, nghiên cứu, làm mạch, mô hình đã hoàn thành và hoạt động đạt yêu cầu. Tần số đặt được hiển thị trên LCD. Giá trị đặt được được nhập từ bàn phím ma trận, biến trở để tạo sự linh hoạt, có thể thay đổi giá trị đặt một cách nhanh chóng.

Tuy nhiên, do lần đầu tiên làm đề về động cơ ba pha nên không tránh khỏi một số hạn chế về phần cứng làm cho mô hình hoạt động chưa được tốt.

5.1 Bộ công suất:

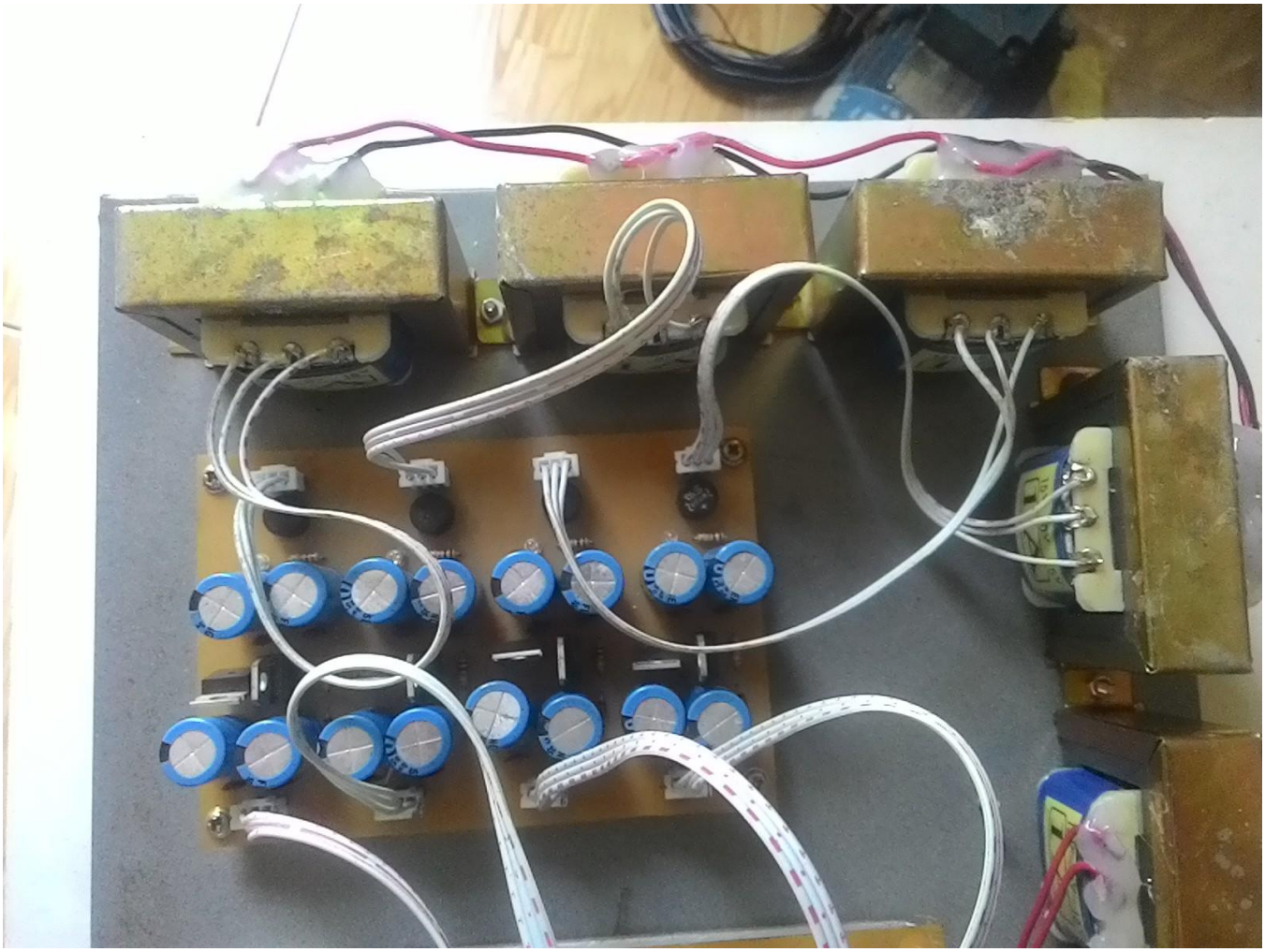


Hình 5.1 Mạch công suất.

5.2 Bộ điều khiển:

Hình 5.2 Mạch điều khiển.

5.3 Mạch nguồn kích:



Hình 5.3 Mạch nguồn kích.

5.4 Bộ vi xử lý:

Màn hình khởi động khi nguồn được cung cấp:

Hình 5.4 Màn hình khởi động khi mạch được cấp nguồn.

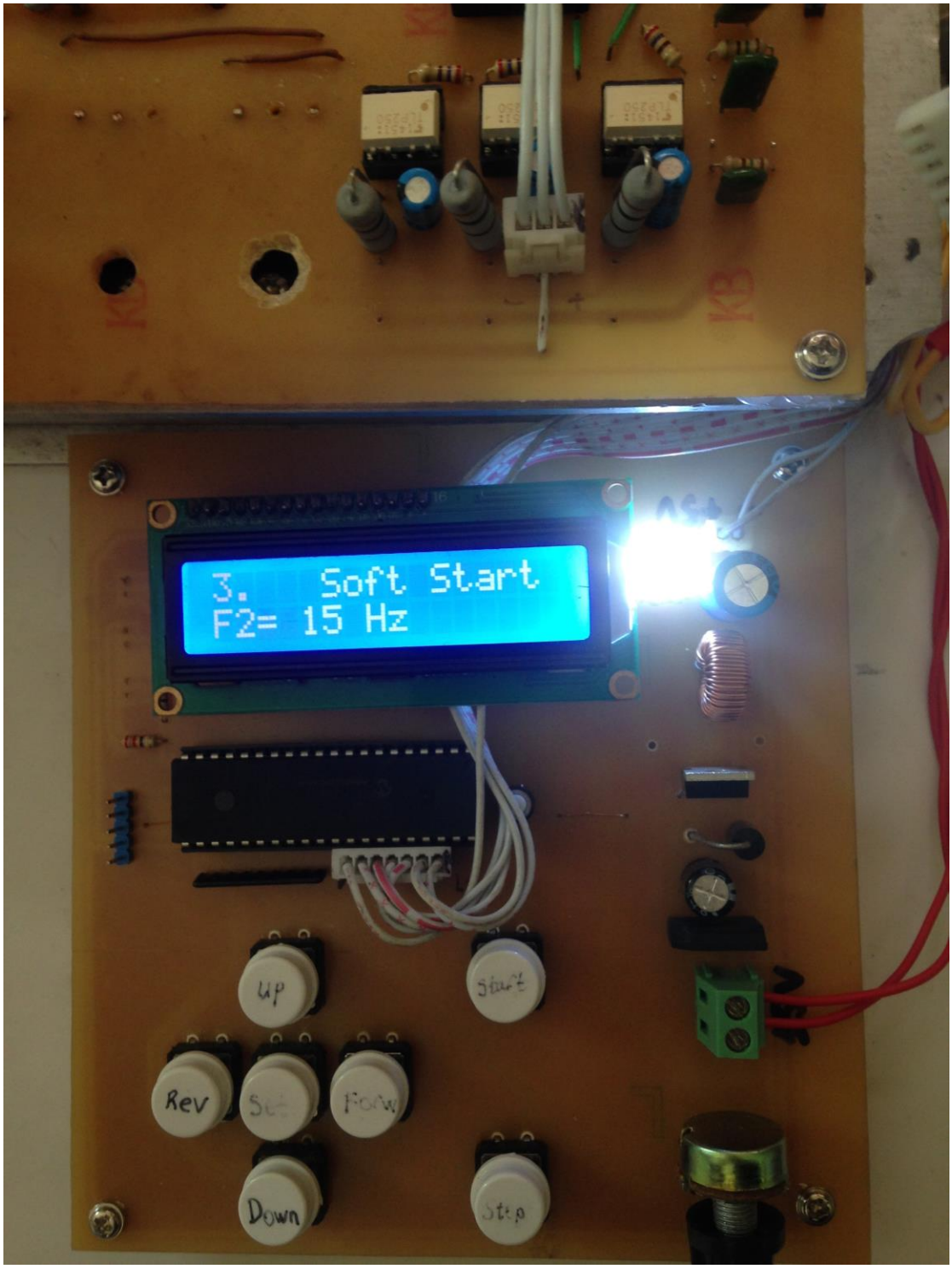
Chế độ 1:

Hình 5.5 Hệ thống hoạt động ở chế độ 1.

Chế độ 2:

Hình 5.6 Hệ thống hoạt động ở chế độ 2.

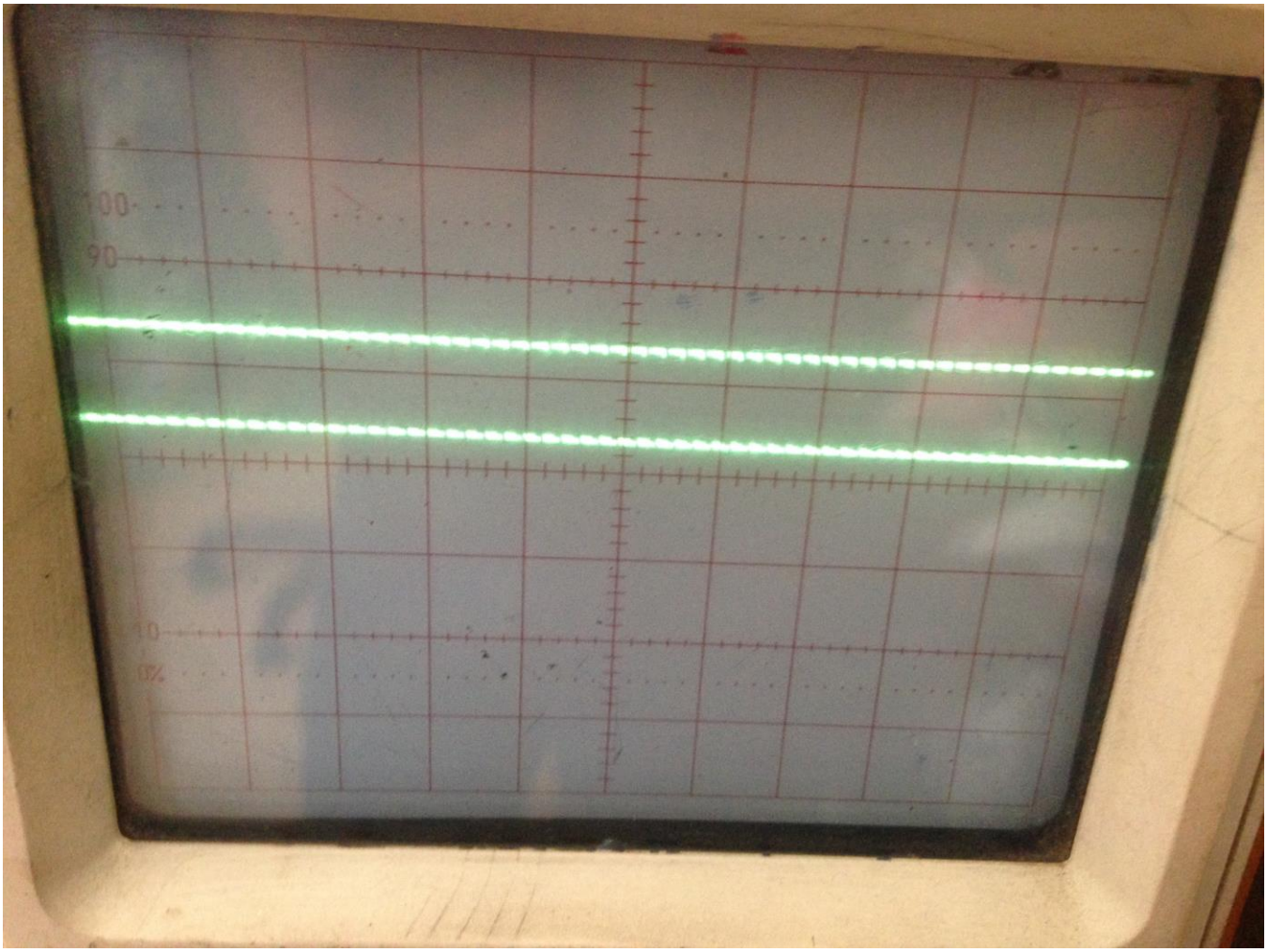
Chế độ 3:



Hình 5.7: Hệ thống hoạt động ở chế độ 3.

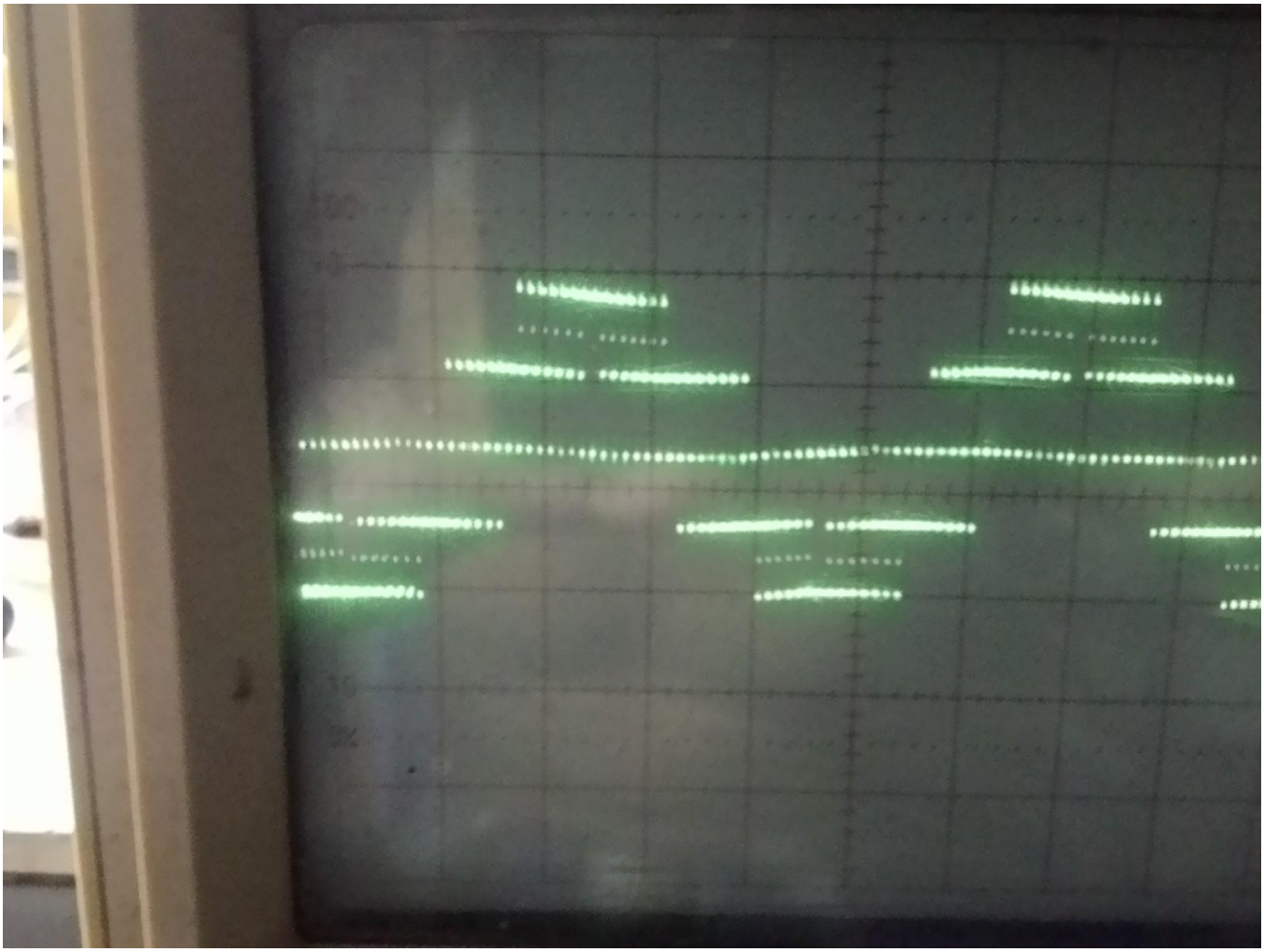
5.4 Kết quả mô phỏng:

Dạng sóng ngõ ra ở port E (PWM):



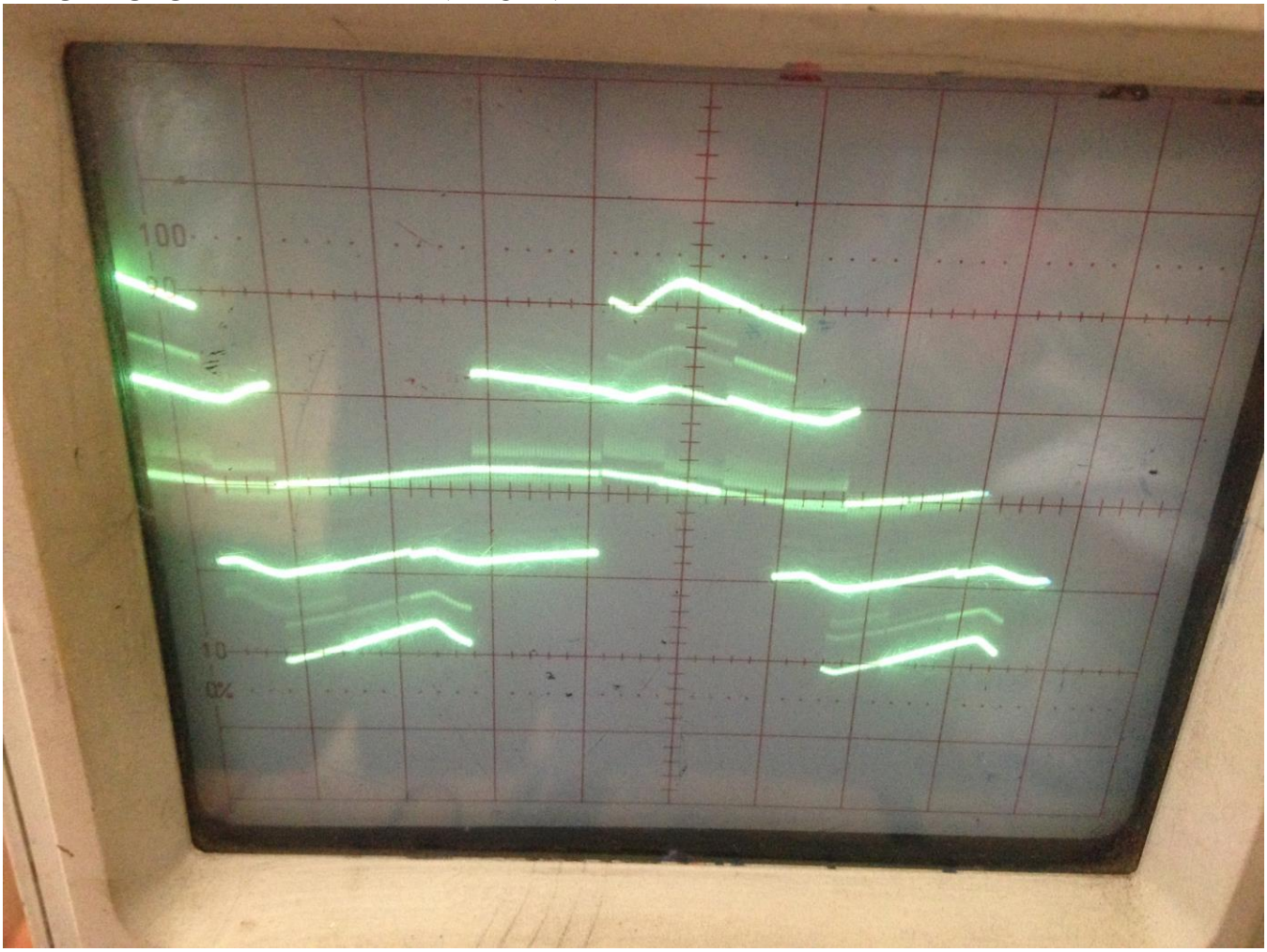
Hình 5.8: Dạng sóng ngõ ra PWM.

Dạng sóng ngõ ra của một pha :



Hình 5.9 Dạng sóng ngõ ra ở tải (1 pha).

Dạng sóng ngõ ra trên tải thực tế (động cơ):



Hình 6.0. Dạng sóng ngõ ra trên tải thực tế (động cơ).

CHƯƠNG 6 KẾT LUẬN VÀ HƯỚNG GIẢI QUYẾT

Mô hình đã hoàn thành nhưng vẫn còn nhiều vấn đề chưa tốt cần phải sửa đổi nếu muốn phát triển và hoàn thiện.

- Về phần cứng, cần có encoder và cảm biến có độ phân giải cao để đo được tốc độ động cơ khi thay đổi tần số.
- Về giao tiếp bổ sung thiết kế giao tiếp được với máy tính.

Tuy vậy, mô hình này cũng là cơ sở, bước đầu, làm nền tảng cho việc hoàn thiện và phát triển sau này.

TÀI LIỆU THAM KHẢO

1. Sách *Truyền động điện thông minh* của thầy Nguyễn Phùng Quang, Nhà xuất bản khoa học và kĩ thuật Hà Nội – 2004.
2. Giáo trình *Lý thuyết điều khiển tự động* của Nguyễn Thị Phương Hà trường Đại học Bách Khoa TP. Hồ Chí Minh -2005
3. Tài liệu *Using the dsPIC30F for Vector Control of an ACIM* của hãng Microchip**.**
4. Datasheet của *dsPIC30F4011*.
5. Tài liệu *dsPIC30F Family Reference Manual* của hãng Microchip.
6. Datasheet của IC *cách ly quang TPL250,*
7. Datasheet của IC *IGBT SGW25N120.*
8. Datasheet của IC 7905.

PHỤ LỤC

□ CHƯƠNG TRÌNH TRÊN VI ĐIỀU KHIỂN

```
/************************************************************
**********************
************************************************************
**********************/
#include <30F4011.h>
#include <math.h>
#include <lcd.h>
#FUSES NOWDT, HS, NOPUT, NOPROTECT, NODEBUG, BROWNOUT, NOWRT, NOPWMPIN
#use delay(clock=20000000)
unsigned int docadc=0,thuannghich=0,setup=0;
unsigned int gt_p1=0,gt_p2=0,gt_p3=0,step=0,t=0,i,tt=0,ttt=0,chayngung=0,tanggiam=0;
const unsigned int bang[]={
1110,1120,1129,1139,1149,1158,1168,1178,1187,1197,1207,1216,1226,
1236,1245,1255,1264,1274,1284,1293,1303,1312,1322,1331,1341,1350,
1360,1369,1378,1388,1397,1406,1416,1425,1434,1444,1453,1462,1471,
2220,2220,2219,2219,2219,2218,2217,2217,2216,2215,2214,2213,2212,
2211,2209,2208,2206,2205,2203,2202,2200,2198,2196,2194,2192,2190,
2187,2185,2182,2180,2177,2175,2172,2169,2166,2163,2160,2157,2153,
2150,2147,2143,2140,2136,2132,2128,2124,2121,2116,2112,2108,2104,
2100,2095,2091,2086,2081,2077,2072,2067,2062,2057,2052,2047,2042,
2036,2031,2025,2020,2014,2009,2003,1997,1991,1985,1979,1973,1967,1961,
1955,1949,1942,1936,1929,1923,1916,1909,1903,1896,1889,1882,1875,
1868,1861,1854,1847,1839,1832,1825,1817,1810,1802,1794,1787,1779,
1771,1764,1756,1748,1740,1732,1724,1716,1708,1699,1691,1683,1675,1666,
1658,1649,1641,1632,1624,1615,1607,1598,1589,1580,1572,1563,1554,
1545,1536,1527,1518,1509,1500,1491,1482,1473,1464,1455,1445,1436,
1427,1418,1408,1399,1389,1380,1371,1361,1352,1342,1333,1323,1314,1304,
1295,1285,1276,1266,1257,1247,1237,1228,1218,1208,1199,1189,1179,
1170,1160,1150,1141,1131,1121,1112,1102,1092,1083,1073,1063,1054,
1044,1034,1025,1015,1005,996,986,977,967,957,948,938,929,919,910,
```

```
900,891,881,872,862,853,843,834,825,815,806,797,787,778,769,760,
750,741,732,723,714,705,696,687,678,669,660,652,643,634,625,617,
608,599,591,582,574,565,557,548,540,532,524,515,507,499,491,483,
475,467,459,451,444,436,428,421,413,406,398,391,383,376,369,362,
355,348,341,334,327,320,313,306,300,293,287,280,274,267,261,255,
249,243,237,231,225,219,213,208,202,197,191,186,180,175,170,165,
160,155,150,145,140,136,131,127,122,118,113,109,105,101,97,93,89,
85,82,78,75,71,68,64,61,58,55,52,49,46,44,41,38,36,34,31,29,27,25,
23,21,19,17,16,14,13,11,10,9,7,6,5,4,4,3,2,2,1,1,0,0,0,0,0,0,0,1,
1,1,2,3,3,4,5,6,7,8,9,10,12,13,15,16,18,20,22,24,26,28,30,32,35,
37,40,42,45,48,51,53,56,60,63,66,69,73,76,80,83,87,91,95,99,103,
107,111,115,120,124,129,133,138,142,147,152,157,162,167,172,177,183,
188,194,199,205,210,216,222,228,233,239,245,252,258,264,270,277,
283,290,296,303,309,316,323,330,337,344,351,358,365,372,379,387,
394,402,409,417,424,432,439,447,455,463,471,479,487,495,503,511,519,
527,536,544,552,561,569,577,586,595,603,612,620,629,638,647,656,
664,673,682,691,700,709,718,727,736,745,755,764,773,782,792,801,
810,819,829,838,848,857,866,876,885,895,904,914,923,933,943,952,
962,971,981,991,1000,1010,1019,1029,1039,1048,1058,1068,1077,1087,1097
};
#define    chay    input(pin_F0)
#define    ngung   input(pin_F1)
#define    thuan   input(pin_F4)
#define    nghich  input(pin_F5)
#define    caidat  input(pin_F6)
#define    tang    input(pin_D0)
#define    giam    input(pin_D2)

void pwmthuan()
  {
```

```
set_tris_e(0b00000000); //cau hinh output port e
setup_motor_pwm(1,MPWM_FREE_RUN,1,0,1000); duty cycle maximum độ rộng xung, thuc nghiem
set_motor_unit(1,1,MPWM_ENABLE| MPWM_FAULT_NO_CHANGE,10,10); 10: dead time, thoi gian tre
set_motor_pwm_duty(1,1,bang[gt_p1]); //gan do rong xung cho pwm1
setup_motor_pwm(1,MPWM_FREE_RUN,1,0,1000);
set_motor_unit(1,2,MPWM_ENABLE| MPWM_FAULT_NO_CHANGE,10,10);
set_motor_pwm_duty(1,2,bang[gt_p2]);
setup_motor_pwm(1,MPWM_FREE_RUN,1,0,1000);
set_motor_unit(1,3,MPWM_ENABLE| MPWM_FAULT_NO_CHANGE,10,10);
set_motor_pwm_duty(1,3,bang[gt_p3]);
  }
void pwmnghich()
  {
set_tris_e(0b000000000);
setup_motor_pwm(1,MPWM_FREE_RUN,1,0,1000);
set_motor_unit(1,1,MPWM_ENABLE| MPWM_FAULT_NO_CHANGE,10,10);
set_motor_pwm_duty(1,1,bang[gt_p1]);

setup_motor_pwm(1,MPWM_FREE_RUN,1,0,1000);
set_motor_unit(1,3,MPWM_ENABLE| MPWM_FAULT_NO_CHANGE,10,10);
set_motor_pwm_duty(1,3,bang[gt_p3]);

setup_motor_pwm(1,MPWM_FREE_RUN,1,0,1000);
set_motor_unit(1,2,MPWM_ENABLE| MPWM_FAULT_NO_CHANGE,10,10);
set_motor_pwm_duty(1,2,bang[gt_p2]);
  }
void chaythuan()
{

        if(gt_p1>719) gt_p1=0;
```

```
		gt_p2=gt_p1+240;
		if(gt_p2>719) gt_p2-=720;
		gt_p3=gt_p1+480;
		if(gt_p3>719) gt_p3-=720;
		 pwmthuan();
		gt_p1+=step;
}
void chaynghich()
{
		if(gt_p1>719) gt_p1=0;
		gt_p2=gt_p1+240;
		if(gt_p2>719) gt_p2-=720;
		gt_p3=gt_p1+480;
		if(gt_p3>719) gt_p3-=720;
		pwmnghich();
		gt_p1+=step; // thoi gian thay doi tu bien tro de thay doi tan so(la t trong bang exel
}

void chaythuan1()
{

		if(gt_p1>719) gt_p1=0;
		gt_p2=gt_p1+240;
		if(gt_p2>719) gt_p2-=720;
		gt_p3=gt_p1+480;
		if(gt_p3>719) gt_p3-=720;
		 pwmthuan();
		gt_p1+=tanggiam;
}
void chaynghich1()
```

```
{
        if(gt_p1>719) gt_p1=0;
        gt_p2=gt_p1+240;
        if(gt_p2>719) gt_p2-=720;
        gt_p3=gt_p1+480;
        if(gt_p3>719) gt_p3-=720;
        pwmnghich();
        gt_p1+=tanggiam;
}
void ADC_INIT()
{
setup_adc(ADC_CLOCK_INTERNAL | ADC_TAD_MUL_0);
setup_adc_ports(sAN0,VSS_VDD);
set_adc_channel(0);
docadc= read_adc();
}
/*****************************************************/

#INT_TIMER3
void interrupts_timer3()
{
SET_TIMER3(1); //NGAT 2 MS

if((chay==0)&&(t>0)){chayngung=1;setup=1;tt=0;}
if(ngung==0)
 { chayngung=2;setup=0;t=0;ttt=0;}
if(nghich==0){thuannghich=2;tt=1;}
if(thuan==0){thuannghich=1;tt=1;}

    docadc=read_adc();
```

```
     docadc=(unsigned int)docadc/150; // chia nhỏ độ phân giải
    step=docadc/5;
}
void pwmthuan();
void pwmnghich();
void chaythuan();
void chaynghich();
void ADC_INIT();
void main()
{
SETUP_TIMER3(TMR_INTERNAL|TMR_DIV_BY_1);
SET_TIMER3(1);
ENABLE_INTERRUPTS(INT_TIMER3);
enable_interrupts(INTR_GLOBAL);
ADC_INIT();
lcd_init();
for(i=0;i<=250;i++)
    {  lcd_gotoxy(1,1);printf(lcd_putc,"MOTOR CONTROLLER    ");
       setup_motor_pwm(1,MPWM_DISABLED,1,0,1000);
       lcd_gotoxy(1,2);
       printf(lcd_putc," 3 Phase/120W              ");
    }
while(TRUE)
{
if(ttt==0)
{
}
if(chayngung==2)
                  {  setup_motor_pwm(1,MPWM_DISABLED,1,0,1000);
                     lcd_gotoxy(1,2);
```

```
                    printf(lcd_putc,"          STOP");
                    tt=0;
                   }
if(setup==0)
{if(!caidat)
  {delay_ms(20);
    if(!caidat)
      t++;chayngung=0;
      if(t>=4){t=4;}
  }
     while(!caidat);
}
if(t==1)
  {
    ttt=1;
     setup_motor_pwm(1,MPWM_DISABLED,1,0,1000);
      lcd_gotoxy(1,1);
      printf(lcd_putc,"         VAR             ");
      lcd_gotoxy(1,2);
      printf(lcd_putc,"       F                 ");
      if(chay==0){chayngung=1;setup=1;}
     }
if(t==2)
    { ttt=2;
    if(tang==0)
    {  delay_ms(200);
      tanggiam++;
      if(tanggiam>=80){tanggiam=80;}
    }
    if(giam==0)
```

```
    {  delay_ms(200);
       tanggiam--;
       if(tanggiam<=0){tanggiam=0;}
    }
       step=tanggiam;
    setup_motor_pwm(1,MPWM_DISABLED,1,0,1000);
    lcd_gotoxy(1,1);
    printf(lcd_putc,"   SET SPEED              ");
    lcd_gotoxy(1,2);
     printf(lcd_putc,"F=%3d Hz  ",tanggiam);
    if(chay==0){chayngung=1;setup=1;}
}
 if(t==3)
 {
  if(tang==0)
    {  delay_ms(200);
       tanggiam++;
       if(tanggiam>=100){tanggiam=100;}
    }
  if(giam==0)
    {  delay_ms(200);
       tanggiam--;
       if(tanggiam<=0){tanggiam=0;}
    }
   tsd=tanggiam
  lcd_gotoxy(1,2);
  printf(lcd_putc,"F1=%3d Hz  ",tanggiam);
if(t==4)
 {
  if(tang==0)
```

```
    {  delay_ms(200);
       tanggiam1++;
       if(tanggiam1>=10){tanggiam1=10;}
    }
 if(giam==0)
    {  delay_ms(200);
       tanggiam1--;
       if(tanggiam1<=0){tanggiam1=0;}
    }
 tam=tam+0.1*tsd/tanggiam1;
 step=(int)tam;
}
if(chayngung==1)
{
if(tt==0)
  {lcd_gotoxy(1,2);
  printf(lcd_putc,"Select F/R  Stop");
  }
if((tt==1)&&(ttt==1))
{
if(thuannghich==2)
{
chaynghich();
 lcd_gotoxy(1,2);
  printf(lcd_putc,"F=%3d Hz  ",step);
  lcd_gotoxy(10,2);
  printf(lcd_putc," REV          ");}
if(thuannghich==1)
  {
  chaythuan();
```

```
  lcd_gotoxy(1,1);printf(lcd_putc,"MOTOR CONTROLLER    ");
 lcd_gotoxy(1,2);
  printf(lcd_putc,"F=%3d Hz  ",step);
  lcd_gotoxy(10,2);
  printf(lcd_putc,"  FORW          ");
}
}
if((tt==1)&&(ttt==2))
{
if(thuannghich==2)
{
chaynghich1();
 lcd_gotoxy(1,2);
  printf(lcd_putc,"F=%3d Hz  ",tanggiam);
  lcd_gotoxy(10,2);
  printf(lcd_putc," REV          ");}
if(thuannghich==1)
  {
  chaythuan1();
  lcd_gotoxy(1,1);printf(lcd_putc,"MOTOR CONTROLLER    ");
 lcd_gotoxy(1,2);
  printf(lcd_putc,"F=%3d Hz  ",tanggiam);
  lcd_gotoxy(10,2);
  printf(lcd_putc,"  FORW          ");
 }
  }
}
}
}
```

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
www.hcmute.edu.vn