

**Trường đại học giao thông vận tải TP.Hồ Chí Minh**

Khoa điện – điện tử viễn thông

Bộ môn điện tử viễn thông

🙣🕮🙡



Báo cáo đồ án viễn thông 1

**Đề tài: Mạch điều khiển tốc độ động cơ DC không phản hồi, điều khiển tốc độ động cơ 1 chiều công suất nhỏ (động cơ máy casssette, video cũ) với khởi động tăng dần tốc độ, giảm tốc từ từ, đảo chiều và hãm động cơ. Có 2 cấp tốc độ cho mỗi chiều. Đo và hiển thị trên LCD. Mạch nạp điện áp cho động cơ là mạch điều chế độ rộng xung.**

**GVHD: Mr. LẠI NGUYỄN DUY**

**SVTH: TRẦN VĂN HẢI TRIỀU**

**LỚP: DV18**

TP. HCM, 12-2020

**LỜI CẢM ƠN**

Xin trân trọng cảm ơn đến các quý Thầy Cô khoa Điện- Điện Tử Viễn Thông, Trường Đại học Giao Thông Vận Tải TP. Hồ Chí Minh, đã hỗ trợ và tạo điều kiện tối đa cho việc tiếp thu và học hỏi những kiến thức bổ ích trong những năm qua. Đó là tiền để để có thể triển khai và thực hiện đề tài hôm nay.

Xin gửi lời cảm ơn đặc biệt nhất đến thầy Lại Nguyễn Duy, giảng viên hướng dẫn Đồ án 1. Nhiệt tình trả lời và hỗ trợ hết mình, sớm tiếp cận được những kiến thức chuyên môn để đề tài được hoàn thiện nhất.

Trong quá trình thực hiện, không tránh khỏi những thiếu xót về kiến thức và khả năng thi công. Do đó rất mong sự đóng góp ý kiến chân thành của Thầy để để tài tiến bộ hơn và có thêm nền tảng cho những đồ án sau này.

**NHẬN XÉT CỦA GIẢNG VIÊN**

Danh mục hình ảnh

Hình 1: Sơ đồ khối của AT89C51

Hình 2: Sơ đồ chân của AT89c51

Hình 3: Bảng chức năng của PORT 3

Hình 4: Sơ đồ chân LCD

Hình 5: Giao tiếp LCD với 8051

Hình 6: Các lệnh cho LCD 16x2

Hình 7: Encoder

Hình 8: Encoder có lỗ định vị

Hình 9: Hình ảnh thực tế của thạch anh

Hình 10 Relay

Hình 11: Transistor TIP42

Hình 12: PWM

Hình 13: Mạch cầu H

Hình 14: Nguyên lí mạch cầu H

Hình 15: Mạch ở trạng thái ban đầu

Hình 16: Mạch khi ấn nut “phai”

Hình 17: Mạch khi ấn nut “trai”

Hình 18: Mạch khi ấn nút “nhanh”

Hình 19: Mạch khi ấn nút “cham”

Hình 20: Mạch mô phỏng

Hình 21: Mạch PCB

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc86048168)

[CHƯƠNG 1: PHÂN TÍCH HỆ THỐNG 2](#_Toc86048169)

[1.1 Ý tưởng chung 2](#_Toc86048170)

[1.2 Xác định bài toán 2](#_Toc86048171)

[CHƯƠNG 2: GIỚI THIỆU LINH KIỆN 3](#_Toc86048172)

[2.1 Vi điều khiển AT89C51 3](#_Toc86048173)

[2.2 LCD hiển thị 5](#_Toc86048174)

[2.3 Động cơ DC và các linh kiện hỗ trợ 9](#_Toc86048175)

[2.3.1 Encoder 9](#_Toc86048176)

[2.3.2 Tụ điện 11](#_Toc86048177)

[2.3.3 Điện trở 12](#_Toc86048178)

[2.3.4 Thạch anh dao động 12](#_Toc86048179)

[2.3.5 Relay 13](#_Toc86048180)

[2.3.6 Transistor TIP42 13](#_Toc86048181)

[CHƯƠNG 3: THIẾT KẾ VÀ MÔ PHỎNG 14](#_Toc86048182)

[3.1 Nguyên lí hoạt động của mạch 14](#_Toc86048183)

[3.2 Lưu đồ thuật toán 21](#_Toc86048184)

[3.3 Mạch mẫu (mô phỏng) 22](#_Toc86048185)

[3.4 Code C lập trình cho vi xử lí để vận hành toàn hệ thống 22](#_Toc86048186)

[KẾT LUẬN 34](#_Toc86048187)

# LỜI NÓI ĐẦU

Ngày nay, với những ứng dụng của khoa học tiên tiến, thế giới đã và đang ngày càng một thay đổi, văn minh và hiện đại hơn. Sự phát triển của kĩ thuật điện tử đã tạo ra hàng loạt thiết bị với những ưu điểm nổi bật như độ chính xác cao, tốc độ nhanh, gọn nhẹ là những yếu tố không thể thiếu để giúp cho hoạt động của con người đạt hiệu quả cao.

Các bộ điều khiển sử dụng vi điều khiển tuy nhìn thoáng qua khá đơn giản nhưng để vận hành và kiểm soát thì rất phức tạp. Các bộ vi điều khiển theo thời gian cùng với sự phát triển của công nghệ bán dẫn đã tiến triển rất nhanh. Điện tử đã trở thành một ngành đa nhiệm vụ bởi chính sự đáp ứng được những đòi hỏi không ngừng từ các lĩnh vực công – nông – lâm – ngư nghiệp cho đến các nhu cầu cần thiết trong hoạt động đời sống hằng ngày.

Một trong những ứng dụng thực tế trong đời sống là đo và hiển thị tốc độ động cơ. Đây là một lĩnh vực quan trọng được áp dụng trong nhiều lĩnh vực và dây chuyền sản xuất. Tốc độ và chiều quay của động cơ sẽ được hiển thị trên một màn hình LCD, do đó dễ dàng quan sát và kiểm tra rủi ro kịp thời đưa ra điều chỉnh cho phù hợp với nhu cầu. Chính vì thế, trong đồ án 1 này, em quyết định nhận làm đề tài về đo, kiểm soát động cơ và hiên thị trên LCD dùng 8051. Cụ thể bằng việc ghép nối vi điều khiển AT89C51 với màn hình LCD để hiển thị và dùng encoder. Xin phép trình bày nội dung cụ thể của đề tài như dưới đây. Kính mong Thầy xem, cho nhận xét và đánh giá để đề tài được đầy đủ hơn.

Xin chân thành cảm ơn!

# CHƯƠNG 1: PHÂN TÍCH HỆ THỐNG

## 1.1 Ý tưởng chung

Hệ thống động cơ thiết kế các chế độ: tăng tốc, giảm tốc, xoay thuận, xoay ngược và hãm động cơ, hiển thị số vòng/phút lên LCD.

* Tăng tốc: nhanh dần và hai cấp tốc độ (mở rộng xung PWM).
* Giảm tốc: chậm dần và hai cấp tốc độ (thu hẹp xung PWM).
* Điều khiển quay thuận hoặc quay ngược tuỳ ý (đổi nguồn cấp thay phiên cho 2 đầu động cơ).
* Có một màn hình LCD 16 cột 2 dòng (dòng 1 hiển thị chiều quay và độ rộng xung PWM theo phần trăm, dòng 2 hiển thị số vòng/phút của động cơ).
* Hệ thống nút nhấn gồm 7 nút: nhanh, chậm, trái, phải, dừng, cấp 1 và cấp 2.
* Phát xung vung với tần số cụ thể và điều chế độ rộng xung cho việc điều khiển.

## 1.2 Xác định bài toán

Có nhiều cách để tiếp cận và giải quyết đề tài, tuỳ theo yều cầu và độ chính xác của mỗi người mà có những cách khác nhau. Ở đây, thiết kế điều khiển động cơ DC dùng vi điều khiển 8051 với những ưu điểm: bố trí đơn giản, linh kiện dễ kiếm, dễ lập trình.

Điều khiển động cơ DC gồm 4 khối chính: khối nút bấm, khối xử lí, khối hiển thị và khối thực thi.

Khối nút bấm gồm các phím điều khiển chế độ và tốc độ quay của động cơ DC

Khối xử lí gồm vi xử lí AT89C51 làm nhiệm vụ chính được cấp nguồn 5V.

Khối hiển thị gồm LCD được cấu hình để hiện thị chữ và số.

Khối thực thi gồm một động cơ DC cùng các linh kiện hỗ trợ khác mắc theo kiểu mạch cầu H.

# CHƯƠNG 2: GIỚI THIỆU LINH KIỆN

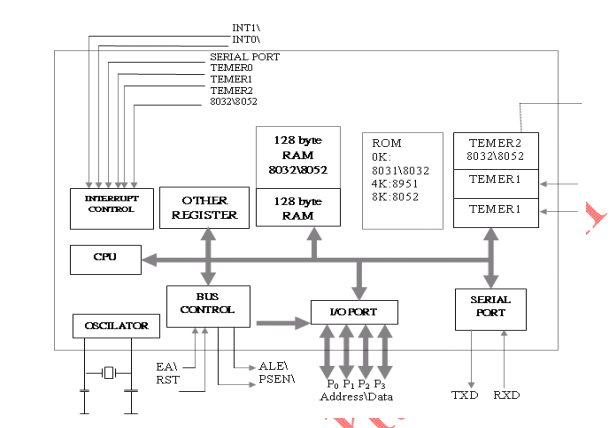
## 2.1 Vi điều khiển AT89C51

AT89C51 là một vi điều khiển 8 bit, chế tạo theo công nghệ CMOS chất lượng cao, công suất thấp với 4KB PEROM (Flash Programeable and Erasable Read Only Memory).

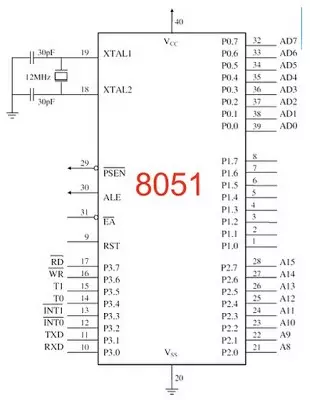
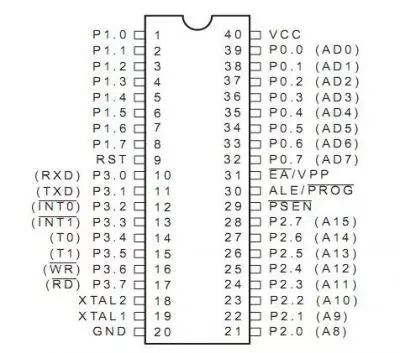
Các đặc điểm của AT89C51 được tóm tắt như sau:

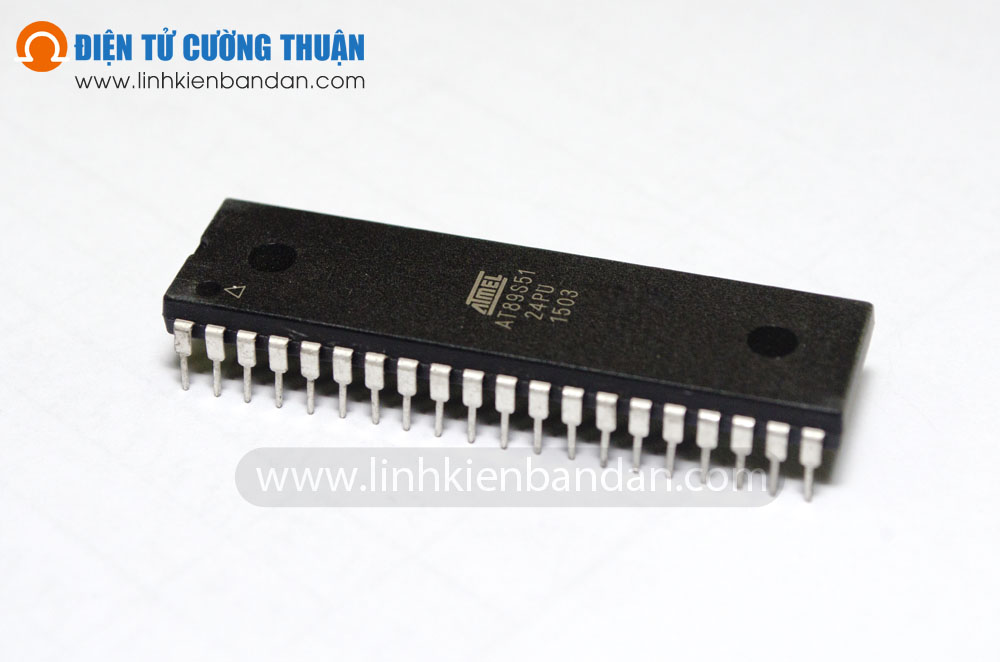
* 4KB bộ nhớ, có thể lập trình nhanh, có thể ghi xoá tới 1000 chu kì.
* Tần số hoạt động từ 0Hz đến 24MHz
* 3 mức khoá bộ nhớ lập trình
* 2 bộ timer/counter 16 bit
* 128 Byte RAM nội
* 4 Port xuất nhập (I/O) 8 bit
* Giao tiếp nối tiếp
* 64 KB vùng nhớ mã ngoài
* 64KB vùng nhớ dữ liệu ngoài
* Xử lí Boolean (hoạt động trên bit đơn)
* 210 vị trí nhớ có thể định vị bit
* 4µs cho hoạt động nhân hoặc chia (với tần số thạch anh 12MHz)

A – Sơ đồ khối và sơ đồ chân của AT89C51



Hình 1 – Sơ đồ khối của AT89C51

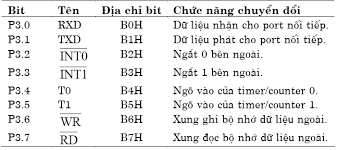




Hình 2 – Sơ đồ chân của AT89C51

B – Sơ lược các chân của AT89C51

* VCC: chân cung cấp điện.
* GND: chân nối đất.
* Port 0: gồm 8 chân 32- 39 (P0.0 –P0.7) ngoài chức năng xuất nhập ra, port 0 còn là bus đa hợp dữ liệu và địa chỉ (AD0 – AD7), chức năng này sẽ được sử dụng khi AT89C51 giao tiếp với thiết bị ngoài có kiến trúc bus.
* Port 1: chân 1- 8 (P1.0 – P1.7) là port I/O, có thể dùng giao tiếp với các thiết bị ngoài nếu cần. Xuất nhập theo bit và byte. Ngoài ra, 3 chân P1.5, P1.6, P1.7 được dùng để nạp ROM theo chuẩn ISP, 2 chân P1.0 và P1.1 được dùng cho bộ Timer 2.
* Port 2: chân 21 – 28 (P2.0 – P2.7) là port có tác dụng kép. Được dùng như các đường xuất nhập hoặc byte cao của bus địa chỉ đối với các thiết bị dùng bộ nhớ mở rộng.
* Port 3: chân 10 – 17 (P3.0 – P3.7) là port xuất nhập 8 bit 2 chiều có các điện trở kéo lên bên trong. Các chân của port này có nhiều chức năn, các công dụng chuyển đổi liên hệ với các đặc tính đặc biệt của 8951

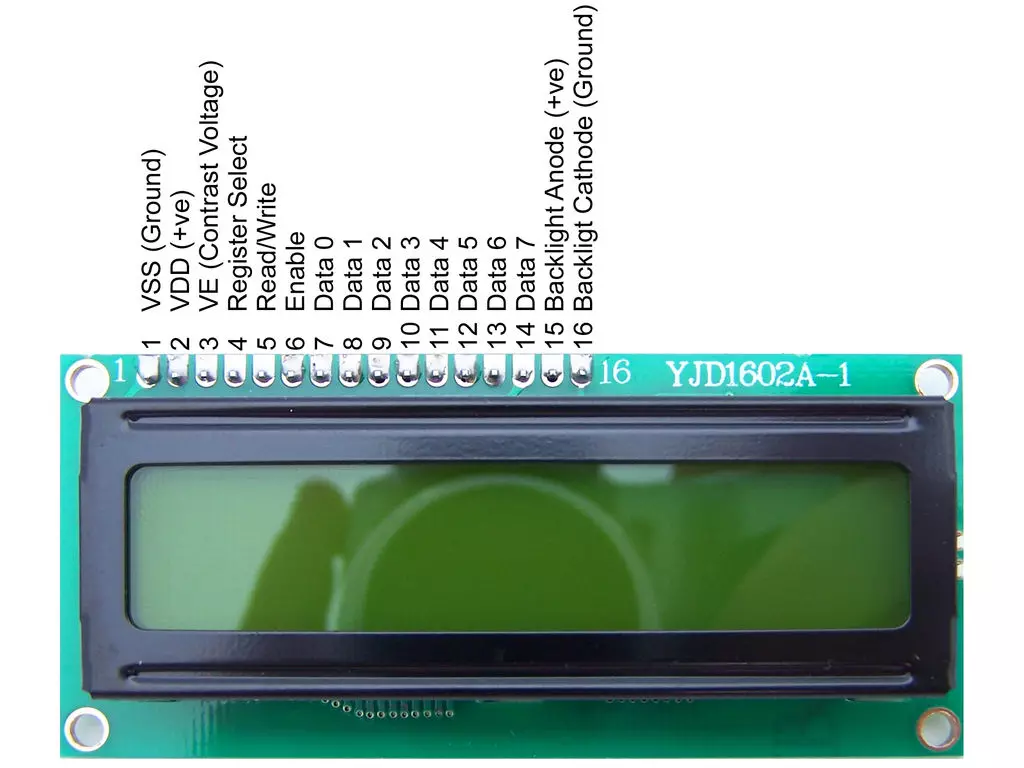


Hình 3 – Bảng chức năng của Port 3

* RST: ngõ vào reset. Mức cao trên chân này trong 2 chu kì máy khi bộ dao động đang hoạt động sẽ RESET AT89C51 (tương đương 2 µs đối với thạch anh 12 MHz).
* ALE (Address Latch Enable): ALE là tín hiệu để chốt địa chỉ vào của một thanh ghi bên ngoài trong nửa đầu của chu kì bộ nhớ. Sau đó các đường Port 0 dùng để xuất và nhập dữ liệu trong nửa chu kì sau của bộ nhớ.
* PSEN (Program Store Enable): PSEN là điều khiển để cho phép chương trình bộ nhớ mở rộng và thường được nối đến chân OE/ (Output Enable) của một EPROM để cho phép đọc các byte mã lệnh. PSEN sẽ ở mức thấp trong thời gian đọc lệnh. Các mã nhị phân của chương trình được đọc từ EPROM qua Bus và được chốt vào thanh ghi lệnh của bộ vi điều khiển để giải mã lệnh. Khi thi hành chương trình trong ROM nội PSEN sẽ ở mức cao.

## 2.2 LCD hiển thị

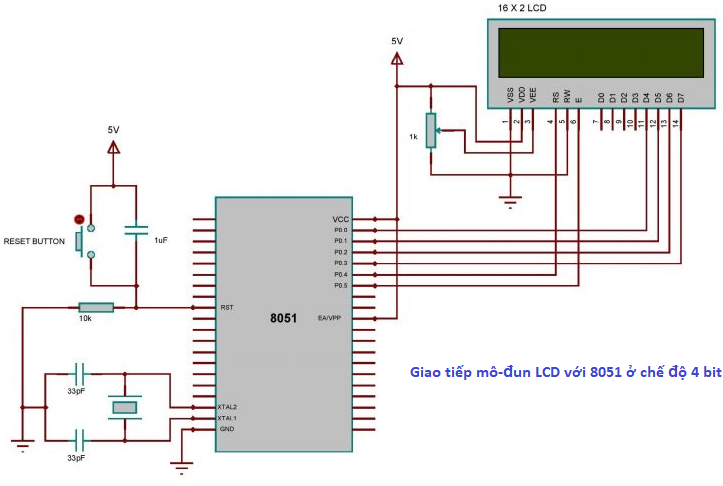
LCD 16x2 là loại LCD rất phổ biến. Nó bao gồm 16 hàng và 2 cột. Số chân, tên của chúng và các chức năng tương ứng được hiển thị trong bảng bên dưới:





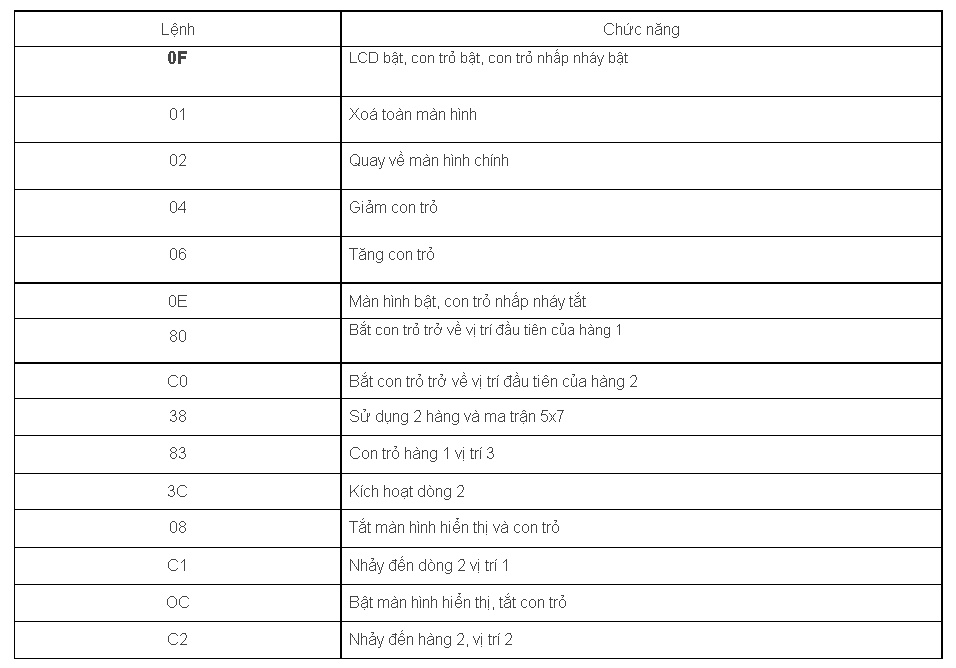
Hình 4 – Sơ đồ chân LCD

* Chân Vee là để điều chỉnh độ tương phản của màn hinhg LCD và độ tương phản có thể được thay đổi bằng cách điều chỉnh điện áp ở chân này. Thực hiện bằng cách nối một đầu của biến trở với Vcc (5V), đầu kia với GND và nối chân giữa của biến trở với Vee.



Hình 5 – Giao tiếp LCD với 8051

* LCD có 2 thanh ghi tích hợp là thanh ghi dữ liệu và thanh ghi lệnh. Thanh ghi dữ liệu để đặt dữ liệu sẽ được hiển thị và thanh ghi lệnh để đặt các lệnh. LCD 16x2 có một bộ lệnh mỗi lệnh sẽ thực thi một công việc cụ thể với màn hình. Nếu mức logic cao đưa vào chân RS thì sẽ chọn thanh ghi dữ liệu và mức logic thấp ở chân RS thì sẽ chọn thanh ghi lệnh. Nếu để chân RS lên mức cao sau đó đặt dòng dữ liệu 8 bit (D0 đến D7), LCD sẽ nhận ra đó là một dữ liệu hiển thị. Ngược lại ở mức thấp, nó sẽ nhận ra đó là một lệnh.
* Chân R/W có nghĩa là để chọn giữa chế độ đọc và ghi. Mức cao ở chân này cho phép đọc và mức thấp cho phép ghi.
* Chân E là cho phép kích hoạt LCD. Hoạt động bằng cách chuyển đổi từ mức cao xuống mức thấp ở chân này.
* D0 đến D7 là các chân dữ liệu. Dữ liệu hiển thị và lệnh được đặt trên các chân này.
* LED+ là cực dương của đèn LED phía sau và chân này phải được kết nối tới Vcc có điện trở hạn dòng thích hợp.
* LED- là cực âm của đèn LED phía sau và chân này phải nối đất.



Hình 6 – Các lệnh cho LCD 16x2

Muốn LCD hoạt động phải cấp nguồn và khởi tạo trước cho LCD

* Khởi tạo LCD

Sau đây là các bước phổ biến dùng hầu hết trong các ứng dụng

B1: Gửi 38H đến dòng dữ liệu 8 bit để khởi tạo

B2: Gửi 0FH để bật LCD, con trỏ Bật và con trỏ nhấp nháy ON

B3: Gửi 06H để tăng vị trí con trỏ

B4: Gửi 01H để xoá màn hình và trả về con trỏ

* Đưa dữ liệu vào LCD

LCD có các chân RS, R/W và E. Chính trạng thái logic của các chân này làm cho LCD xác định xem đầu vào dữ liệu là lệnh hay dữ liệu cần được hiển thị. Các bước gửi dữ liệu đến LCD như sau:

Đặt R/W mức thấp

Đặt RS=0 nếu byte dữ liệu là lệnh và RS=1 nếu byte dữ liệu là dữ liệu sẽ được hiển thị

Đặt byte dữ liệu trên thanh ghi dữ liệu

Xung E từ cao xuống thấp

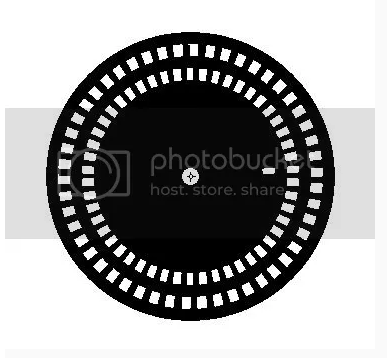
Lặp lại các bước trên để gửi dữ liệu khác.

## 2.3 Động cơ DC và các linh kiện hỗ trợ

### 2.3.1 Encoder

Ở đây em dùng encoder để mô phỏng động cơ DC, mục đích dùng quản lí vị trí góc quay của một đĩa quay, đĩa quay có thể là bánh xe, trục động cơ hoặc bất kì thiết bị quay nào cần xác định vị trí góc.

Encoder có 2 loại: absolute encoder và incremental encoder. Absolute encoder thì tín hiệu nhận được chỉ rõ ràng vị trí của encoder, không cần xử lí gì thêm. Incremental encoder là loại chỉ có 1, 2 hoặc tối đa 3 vòng lỗ. Chẳng hạn khi đục 1 lỗ trên 1 đĩa quay, thì khi đĩa quay 1 vòng sẽ nhận biết được tín hiệu, do đó khi có nhiều lỗ hơn thì thông tin sẽ chi tiết hơn, có nghĩa là 1/4 vòng, 1/8 vòng hoặc 1/n vòng, tuỳ theo số lỗ nằm trên encoder.



Hình 7 – Encoder

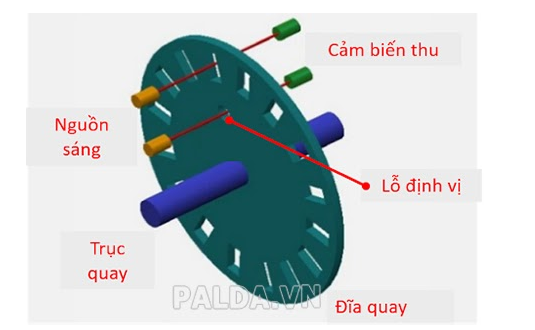
* Nguyên lí hoạt động cơ bản của encoder, LED và lỗ

Nguyên lí cơ bản encoder là một đĩa tròn xoay quay quanh trục. Trên đĩa có các lỗ (rãnh). Người ta dùng một đèn LED để chiếu lên mặt đĩa. Khi đĩa quay chỗ không có lỗ (rãnh), đèn led không chiếu xuyên qua được, chỗ có lỗ (rãnh) thì đèn led chiếu xuyên qua. Khi đó, phía mặt kia người ta đặt một con mắt thu. Với các tín hiệu có, hoặc không có ánh sáng chiếu qua, người ta ghi nhận được đèn led có chiếu qua lỗ hay không. Khi trục quay giả sử trên đĩa có một lỗ duy nhất, cứ mỗi lần con mắt thu nhận được tín hiệu đèn led, thì có nghĩa là đã quay được một vòng. Đây là nguyên lí rất cơ bản của encoder.

* Hoạt động của encoder

Ở đây sẽ xét incremental encoder, cứ mỗi lần quay qua một lỗ encoder sẽ tăng 1 đơn vị trong biến đếm. Tuy nhiên nếu cứ đếm vô hạn như thế, thì không thể biết chính xác khi nào nó quay hết một vòng. Giả sử như encoder có 1000 lỗ và dùng lỗ để đếm thì con số là vô cùng lớn. Bên cạnh đó, có những rung động nào đó mà không quản lí được cũng sẽ ảnh hưởng tới quá trình này. Càng hoạt động lâu dài thì sai số tích luỹ càng nhiều dẫn đến những sai sót không mong muốn.

Để tránh điều này xảy ra, người ta đưa vào thêm một lỗ định vị để đếm số vòng đã quay của encoder. Như vậy cho dù có lệch xung, mà thấy encoder đi qua lỗ định vị này thì sẽ biết encoder bị đếm sai ở đâu đó. Nếu vì một rung động nào đó, mà không thấy encoder đi qua lỗ định vị, thì từ số xung và việc đi qua lỗ định vị cũng dễ dàng phán đoán sai lệch của encoder.



Hình 8 – Encoder có lỗ định vị

Bên cạnh đó để biết encoder đang xoay theo chiều nào, người ta đặt thêm một vòng lỗ ở giữa vòng lỗ thứ nhất và lỗ định vị (Hình 6). Vị trí góc của các lỗ vòng 1 và các lỗ vòng 2 lệch nhau. Các cạnh của lỗ vòng 2 nằm ngay giữa các lỗ vòng 1 và ngược lại.

### 2.3.2 Tụ điện

Tụ điện là một linh kiện thụ động, cấu tạo của tụ điện là hai bảng cực bằng kim loại ghép cách nhau một khoảng d ở giữa hai bản tụ là dung dịch hay chất điện môi cách điện có điện dung C. Đặc điểm của tụ là cho dòng xoay chiều đi qua, ngăn dòng điện một chiều.

Khi tụ nạp điện thì sẽ bắt đầu nạp từ điện áp 0V tăng dần tới điện áp Vdc theo hàm số mũ với thời gian t. Điện áp tức thời trên hai đầu tụ được tính theo công thức sau:

Vc(t)=Vdc(1-e-t/τ)

Khi tụ điện xả thì điện áp trên tụ từ trị số Vdc sẽ giảm dần đến 0V theo hàm số mũ đối với thời gian t. Điên áp trên hai đầu tụ khi xả được tính theo công thức:

Vc(t)=Vdc.e-t/τ

Trong đó: t là thời gian tụ nạp (s)

e=2,71828

τ=RC (-s)

### 2.3.3 Điện trở

Điện trở là linh kiện thụ động có tác dụng cản trở cả dòng và áp. Điện trở được sử dụng rất nhiều trong các mạch điện tử. Điện trở của dây dẫn có chỉ số lớn hay nhỏ phụ thuộc vào vật liệu làm dây, tỉ lệ thuận với chiều dài và tỉ lệ nghịch với tiết diện dây dẫn. Công thức tính:

R=pl/S hoặc R=U/I

Trong đó:

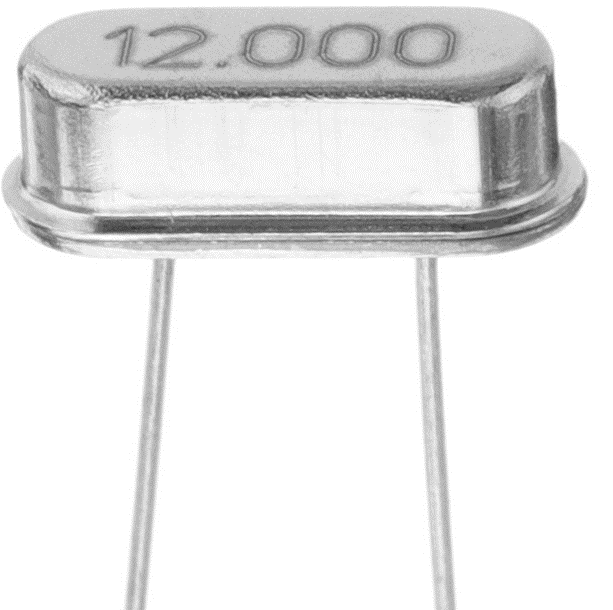
p: là điện trở suất của vật liệu (Ωm hay Ωmm2/m)

S: là tiết diện của dây (m2 hay mm2)

l: là chiều dài của dây (m)

R: điện trở (Ω)

### 2.3.4 Thạch anh dao động



Hình 9 – Hình ảnh thực tế của thạch anh

Thạch anh dao động có nhiệm vụ tạo các xung điều khiển thích hợp phục vụ cho vi điều khiển, ở đây sử dụng thạch anh dao động loại 12MHz.

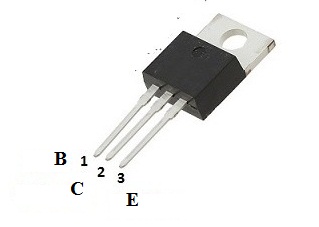
### 2.3.5 Relay



Hình 10 – Relay

Relay là một công tác điện từ được vận hành bởi một dòng điện tương đối nhỏ có thể bật hoặc tắt một dòng điện lớn hơn nhiều. Trái tim của relay là một nam châm điện (một cuộn dây trở thành một nam châm tạm thời khi dòng điện chạy qua nó). Bạn có thể nghĩ về relay như một đòn bẩy điện.

### 2.3.6 Transistor TIP41



Hình 11 – Transistor TIP41

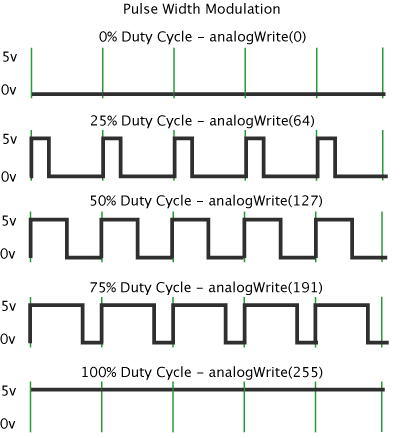
Là một transistor npn silicon khả năng ổn định cao, dùng như một linh kiện điều khiển tốc độ cho động cơ DC, với điện áp từ cực C sang B và từ C sang E lớn đủ để duy trì mức ổn định cho động cơ quay được mượt nhất.

# CHƯƠNG 3: THIẾT KẾ VÀ MÔ PHỎNG

## 3.1 Nguyên lí hoạt động của mạch

* Xung PWM – Pulse Width Modulation

Phương pháp điều chế PWM là phương pháp điều chỉnh điện áp ra tải hay nói cách khác là phương pháp điều chế dựa trên sự thay đổi độ rộng của chuỗi xung vuông dẫn đến sự thay đổi điện áp ra. Trong đồ án này điều khiển động cơ dùng chỉnh độ rộng xung hay còn được gọi là băm xung được dùng phổ biến để thay đổi tốc độ của robot. Điều khiển độ rộng xung được làm bằng cách bật tắt nhanh nguồn điện lên động cơ. Nguồn áp một chiều DC sẽ chuyển thành tín hiệu xung vuông, thay đổi từ 5V xuống 0V.Bằng cách thay đổi chu kì hoạt động của tín hiệu, tức là khoảng thời gian “Bật” thì nguồn điện trung bình động cơ sẽ thay đổi và dẫn đến thay đổi tốc độ.

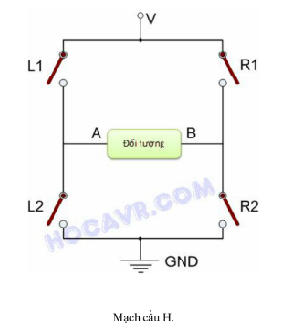


Hình 12 - PWM

Như trên hình, với dãy xung điều khiển 25% Duty Cycle thì xung ON có độ rộng nhỏ hơn nên động cơ chạy chậm. Nếu độ rộng xung ON càng lớn (50%, 75% và 100%) thì động cơ chạy càng nhanh.

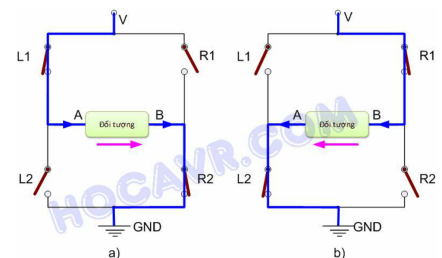
* Mạch cầu H

Mạch cầu H có thể đảo chiều các dòng điện qua tải nên vì thế nó hay được dùng trong các mạch điều khiển động cơ DC và các mạch băm áp. Đối với mạch điều khiển động cơ thì mạch cầu H có thể đảo chiều động cơ một cách đơn giản.



Hình 13- Mạch cầu H

Trong hình 2 đầu v và GND là 2 đầu (+) và (-) của ắc qui, “đối tượng” là động cơ DC cần điều khiển, “đối tượng” này có 2 đầu a và B, mục đích là điều khiển là cho phép dòng điện qua “đối tượng” theo chiều A đến B hoặc B đến A. Thành phần chính tạo nên mạch cầu H chính là 4 “khoá” L1,L2, R1 và R2 (L: Left, R: Right). Ở điều kiện bình thường 4 khoá này “mở”, mạch cầu H không hoạt động. Khảo sát hoạt động của mạch cầu H thông qua các hình minh hoạ dưới đây:



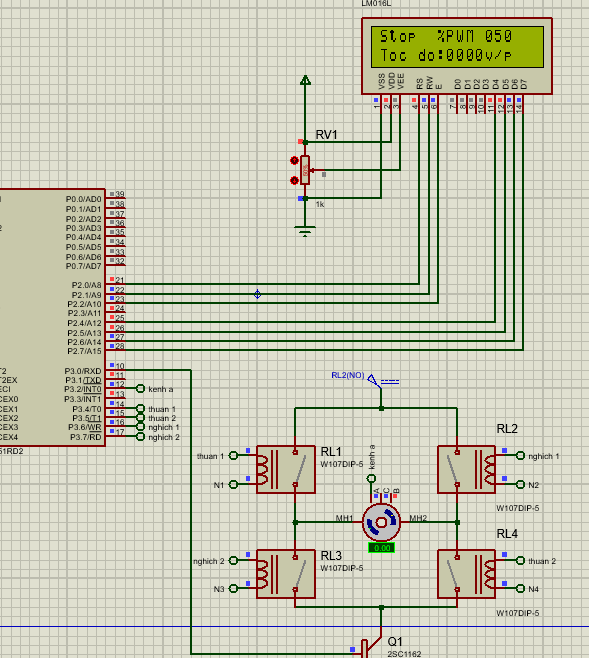
Hình 14- Nguyên lí mạch cầu H

Giả sử bằng cách nào đó mà 2 khoá L1 và R2 được “đóng lại” (L2 và R1 vẫn mở), dễ dàng hình dung có một dòng điện chạy từ V qua khoá L1 đến đầu A và xuyên qua đối tượng đến đầu B của nó trước khi quá khoá R2 và về GND (như hình a). Như thế, với giả sử này sẽ có dòng điện chạy từ A đến B. Bây giờ, xét trường hợp ngược lại R1 và L2 đóng (L1 và R2 mở) sẽ có dòng điện chạy từ B sang a (như hình b). Như vậy có thể dùng nguyên lí này để đảo chiều động cơ dễ dàng.

Nếu đóng đồng thời 2 khoá ở cùng một bên (L1 và L2 hoặc R1 và R2) hoặc thậm chí đóng cả 4 khoá thì sẽ xảy ra hiện tượng “ngắn mạch” (short circuit), V và GND gần như nối trực tiếp với nhau và hiển nhiên ắc qui sẽ bị hỏng hoặc nguy hiểm hơn là cháy nổ mạch xảy ra.

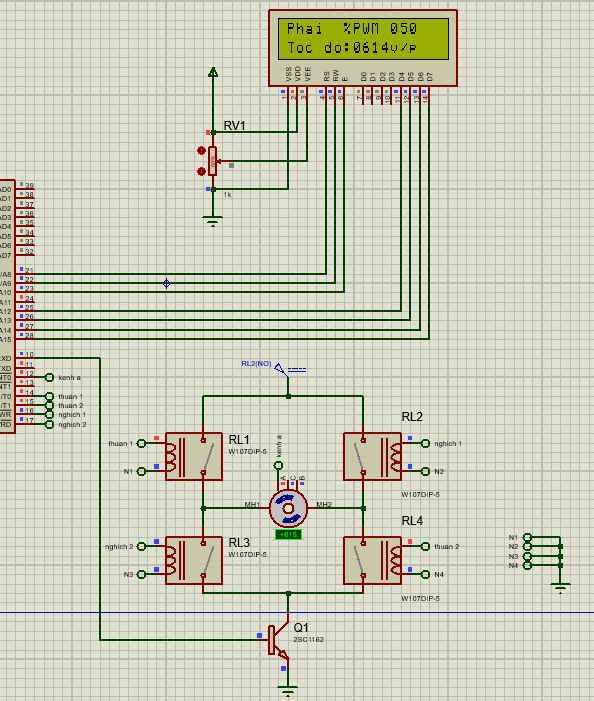
* Nguyên lí hoạt động của toàn mạch

Khi mới cấp nguồn các role trong mạch điều hở, động cơ không hoạt động. Khối hiển thị ban đầu chỉ ra trạng thái “Stop” của động cơ, tốc độ 0 vòng/phút và mức thời gian ON của PWM là 50%.



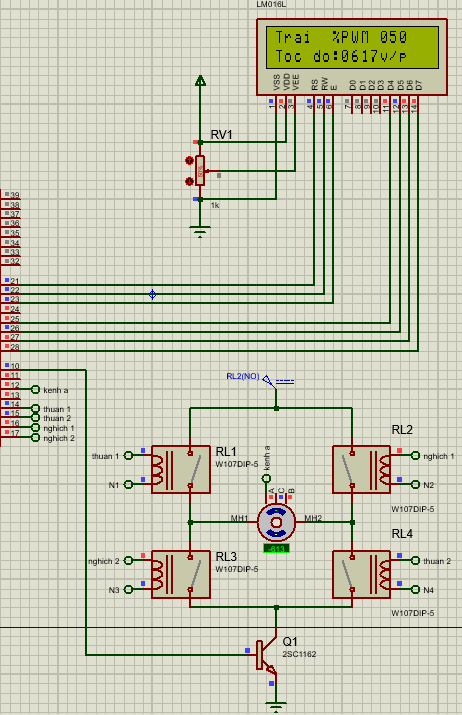
Hình 15 – Mạch ở trạng thái ban đầu

Khi ấn nút “phai” (chân P1.0), khối xử lí tiếp nhận tín hiệu đầu vào và xử lí qua code lập trình. Lúc này khối xử lí xuất ra mức cao ở chân “thuan 1” (P3.4) và “thuan 2” (P3.5) với mức điện áp 5V cấp vào chân tương ứng của RL1 và RL4, đồng thời khối xử lí phát xung PWM 50% ban đầu từ chân P3.0 vào cực B của Q1 để kích cho BJT dẫn. Theo đó, khối thực thi hoạt động với RL1 và RL4 đóng (RL2 và RL3 vẫn hở) điện áp cấp cho động cơ đi qua RL1 tới động cơ và qua RL4, tiếp tục đi qua BJT đang dẫn (50%) xuống đất (GND). Động cơ lúc này quay nhanh dần theo chiều kim đồng hồ dựa vào mức phần trăm PWM cấp cho BJT. Trong lúc động cơ quay thì chân “kenh a” của động cơ nối vào chân P3.2 của vi xử lí gửi tín hiệu xung vuông về khối xử lí để thực hiện việc đếm xung và tính toán tốc độ quay hiển thị lên LCD. Khối hiển thị lúc này chỉ ra trạng thái “Phai”, xung PWM 50% và tốc độ quay đạt tối đa ở mức này là khoảng 600 vòng/phút.



Hình 16- Mạch khi ấn nút “phai”

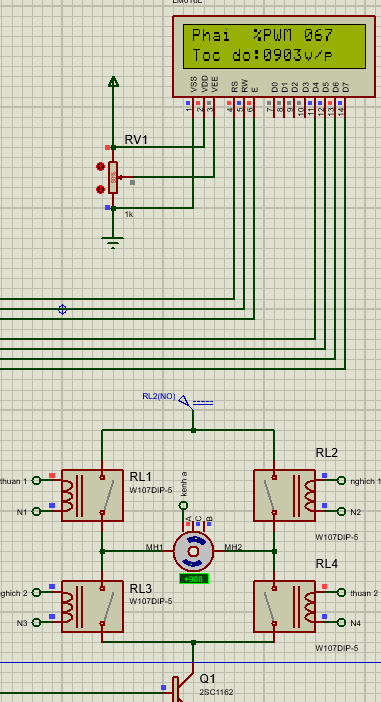
Tương tự khi ấn nút “trai”(P1.1), khối xử lí nhận tín hiệu và đưa ra mức cao ở “nghich 1” (P3.6) và “nghich 2” (P3.7), đồng thời đưa ra mức thấp ở “thuan 1” (P3.4) va “thuan 2” (P3.5) lúc này điện áp qua RL2 đến động cơ rồi tới RL3 (RL1 và RL4 hở) dẫn qua BJT xuống đất. Động cơ lúc này cùng tốc độ với khi ấn nút “phai” nhưng quay theo chiều ngược kim đồng hồ. Khối hiển thị chỉ ra trạng thái “Trai”, xung PWM 50% và tốc độ khoảng 600 vòng/phút.



Hình 17 – Mạch khi ấn nút “trai”

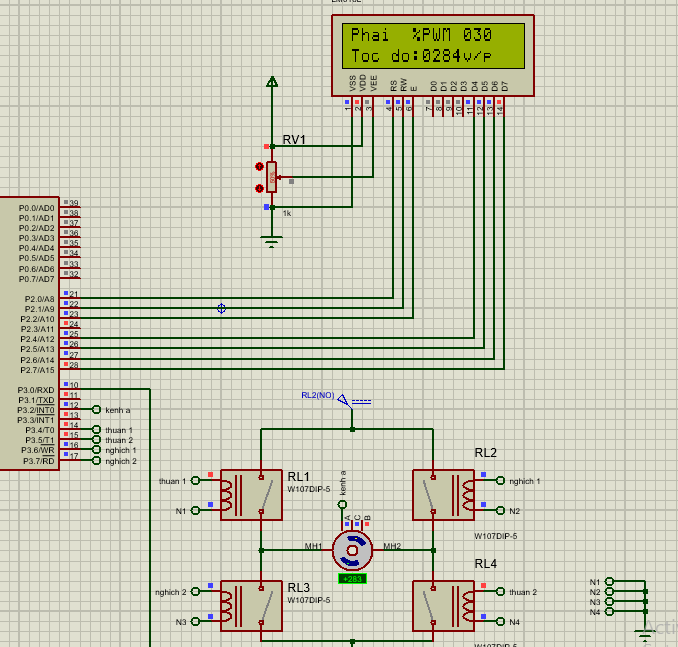
Khi ấn nút “dung” (P1.1), động cơ quay lại trạng thái ban đầu lúc vừa cấp nguồn các role hở, động cơ giảm tốc từ từ đến khi dừng hẳn. Khối hiển thị biểu diễn trạng thái “Stop”, xung PWM không thay đổi và tốc độ là 0 vòng/phút khi dừng hẳn.

Kế tiếp khi ấn nút “nhanh” (P1.2), lúc này khối xử lí nhận tín hiệu và bắt đầu mở rộng dần độ rộng xung PWM được phát ở chân P3.0, ở đây em tính độ rộng xung theo phần trăm tức là tỷ lệ giữa mức cao và mức thấp của xung vuông ( là mức điện áp 0V và 5V), phần trăm PWM mà LCD hiển thị sẽ tương ứng với độ rộng xung mức cao. Ví dụ “PWM: 67%” thì xung mức cao chiếm 67% và mức thấp chiếm 37%. Cứ mỗi lần ấn nút “nhanh” thì PWM cộng thêm 1%. Hình dưới đây là khi ấn nút “nhanh” 17 lần tức là 50 + 17 = 67%. Động cơ vẫn giữ chiều quay khi ấn “phai” hoặc “trai” chỉ khác biệt là tốc độ tăng dần do mức điện áp tăng khi mở rộng xung mức cao (ON) và nhanh chóng đạt tốc độ ngưỡng là khoảng 900 vòng/phút.



Hình 18 – Mạch khi ấn nút “nhanh”

Tương tự với nút “cham” (P1.3), độ rộng xung PWM phát ra ở chân sẽ bị giảm đi 1% mỗi lần nhấn nút “cham”. Ví dụ trong hình, LCD hiển thị PWM = 30%, tốc độ lúc này giảm dần xuống còn 280 vòng/phút.



Hình 19 – Mạch khi ấn nút “cham”

Để giảm bớt số lần ấn nút “nhanh” hoặc “cham” thì tạo thêm 2 nút “cap 1” (P1.5) và nút “cap 2” (P1.6) với nút “cap 1” tương ứng độ rộng xung PWM là 10% và nút “cap 2” sẽ là 100%. Điều này sẽ giúp chọn tốc độ động cơ mong muốn một cách nhanh chóng.

## 3.2 Lưu đồ thuật toán

Khơi tạo xung PWM

Quay thuận

Quay ngược

Tăng tốc

Giảm tốc

Dừng

Chọn cấp tốc độ

Xử lí quay thuận

Xử lí quay ngược

Xử lí tốc độ tăng

Xử lí tốc độ giảm

Cho động cơ dừng

Xử lí cấp 1 hoặc cấp 2

Y text here

Y text here

Y text here

Y text here

Y text here

Y text here

N text here

N text here

N text here

N text here

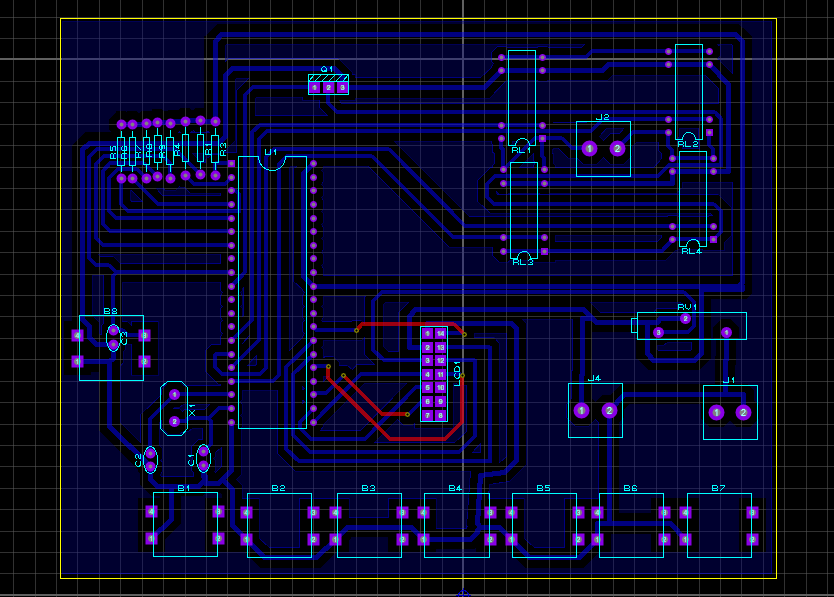
N text here

N text here

## 3.3 Mạch mẫu (mô phỏng)



Hình 20 – Mạch mô phỏng



Hình 21 – Mạch PCB

## 3.4 Code C lập trình cho vi xử lí để vận hành toàn hệ thống

#include<REGX51.H> //khai báo thư viện 8051

sbit pwm\_out=P3^0; // chọn xung PWM ở ngõ ra chân P3.0

unsigned char count\_int;

unsigned char pwm\_duty;

// nối các chân của LCD vào vi xử lí 8051

sbit LCD\_RS=P2^0;

sbit LCD\_RW=P2^1;

sbit LCD\_EN=P2^2;

sbit LCD\_D4=P2^4;

sbit LCD\_D5=P2^5;

sbit LCD\_D6=P2^6;

sbit LCD\_D7=P2^7;

unsigned int count=0,a=0,b=0,f=0,t=0,x=0;

// các hàm delay

void delay\_us(unsigned int t){

unsigned int i;

for(i=0;i<t;i++);

}

void delay\_ms(unsigned int t){

unsigned int i,j;

for(i=0;i<t;i++)

for(j=0;j<125;j++);}

// Tạo xung cạnh xuống cho phép LCD nhận tín hiệu

void LCD\_Enable(void){

LCD\_EN=1;

delay\_us(3);

LCD\_EN=0;

delay\_us(50);

}

// Gửi lần lượt các biết dữ liệu vào LCD

void LCD\_Send4Bit(unsigned char Data){

LCD\_D4=Data & 0x01;

LCD\_D5=(Data>>1)&1;

LCD\_D6=(Data>>2)&1;

LCD\_D7=(Data>>3)&1;

}

// Gửi các lệnh đặc biệt của LCD như hình 5

void LCD\_SendCommand(unsigned char command){

LCD\_Send4Bit(command>>4);

LCD\_Enable();

LCD\_Send4Bit(command);

LCD\_Enable();

}

// Xoá màn hình LCD

void LCD\_Clear(){

LCD\_SendCommand(0x01);

delay\_us(10);

}

// Khởi tạo LCD để LCD bắt đầu hiển thị

void LCD\_Init(){

LCD\_Send4Bit(0x00);

delay\_ms(20);

LCD\_RS=0;

LCD\_RW=0;

LCD\_Send4Bit(0x03);

LCD\_Enable();

delay\_ms(5);

LCD\_Enable();

delay\_us(100);

LCD\_Enable();

LCD\_Send4Bit(0x02);

LCD\_Enable();

LCD\_SendCommand(0x28);

LCD\_SendCommand(0x0c);

LCD\_SendCommand(0x06);

LCD\_SendCommand(0x01);

}

// Chọn hàng và cột để hiển thị

void LCD\_Gotoxy(unsigned char x, unsigned char y){

unsigned char address;

if(!y)address=(0x80+x);

else address=(0xc0+x);

delay\_us(1000);

LCD\_SendCommand(address);

delay\_us(50);

}

// Hiển thị kí tự

void LCD\_PutChar(unsigned char Data){

LCD\_RS=1;

LCD\_SendCommand(Data);

LCD\_RS=0;

}

// Hiện thị chuỗi kí tự

void LCD\_Puts(char \*s){

while(\*s){

LCD\_PutChar(\*s);

s++;

}

}

// Hiển thị số

void LCD\_Number(unsigned long number, unsigned char leng){

unsigned char i;

char CacKyTu[20];

unsigned char temp;

for(i=0;i<leng;i++)

{

temp=number%10;

number=number/10;

CacKyTu[i]=temp;

}

for(i=0;i<leng;i++)

{

LCD\_RS=1;

LCD\_SendCommand(CacKyTu[leng-1-i] + 48);

LCD\_RS=0;

}

}

// Quay phải cho động cơ DC

void phai()

{

if(P1\_0==0)

{

while(P1\_0==0);

P3\_4=1;

P3\_5=1;

P3\_6=0;

P3\_7=0;

TR0=1;

LCD\_Gotoxy(0,0);

LCD\_Puts("Phai");

}

}

// Quay trái cho động cơ DC

void trai()

{

if(P1\_1==0)

{

while(P1\_1==0);

P3\_6=1;

P3\_7=1;

P3\_4=0;

P3\_5=0;

TR0=1;

LCD\_Gotoxy(0,0);

LCD\_Puts("Trai");

}

}

// Dừng động cơ

void dung()

{

if(P1\_2==0)

{

while(P1\_2==0);

TR0=0;

P3\_4=0;

P3\_5=0;

P3\_6=0;

P3\_7=0;

LCD\_Gotoxy(0,0);

LCD\_Puts("Stop");

}

}

// Tăng tốc động cơ

void nhanh()

{

if(P1\_3==0)

{

while(P1\_3==0);

if(pwm\_duty<100)

pwm\_duty++;

else

pwm\_duty=100;

}

}

// Giảm tốc động cơ

void cham()

{

if(P1\_4==0)

{

while(P1\_4==0);

if(pwm\_duty>1)

pwm\_duty--;

else

pwm\_duty=1;

}

}

// Tốc độ cấp 1 của động cơ

void cap1()

{

if(P1\_5==0)

{

while(P1\_5==0);

pwm\_duty=10;

}

}

// Tốc độ cấp 2 của động cơ

void cap2()

{

if(P1\_6==0)

{

while(P1\_6==0);

pwm\_duty=100;

}

}

// Hàm chính

void main(){

P3\_4=0;

P3\_5=0;

P3\_6=0;

P3\_7=0;

TMOD=0x12;

TH0=-100;

ET1=1;

ET0=1;

EX0=1;

IT0=1;

TR0=1;

TR1=1;

EA=1;

pwm\_duty=50;

LCD\_Init();

LCD\_Gotoxy(0,0);

LCD\_Puts("Stop");

while(1){

phai();

trai();

dung();

nhanh();

cham();

cap1();

cap2();

LCD\_Gotoxy(0,1);

LCD\_Puts("Toc do:");

LCD\_Gotoxy(5,0);

LCD\_Puts(" %PWM");

LCD\_Gotoxy(11,0);

LCD\_Number(pwm\_duty,3);

LCD\_Gotoxy(7,1);

LCD\_Number(f,4);

LCD\_Gotoxy(11,1);

LCD\_Puts("v/p");

}

}

// Tạo xung PWM bằng timer 0 của vi xử lí

void timer0(void) interrupt 1

{

count\_int++;

if(count\_int==100)

{

count\_int=0;

}

if(count\_int<pwm\_duty)

{

pwm\_out=1;

}

else

{

pwm\_out=0;

}

}

// Ngắt ngoài 0 và timer 1 dùng đếm xung và tính toán tốc độ cơ

void demxung(void) interrupt 0

{

count++;

}

void timer1(void) interrupt 3

{

t++;

TH1=0xfc;

TL1=0x18;

TR1=1;

if(t==1000)

{

x=count;

count=0;

t=0;

a=x\*2.4247;

f=(a+b)/2.035;

b=a;

}

}

# KẾT LUẬN

Trong quá trình thực hiện đề tài, em rút ra được những điều sau

* Ưu điểm:

Tiếp cận được với các đề tài khoa học kĩ thuật có liên quan, tích luỹ và có thêm chút kinh nghiệm trong việc nghiên cứu tìm hiểu các thông tin cũng như có thêm khả năng lập trình cho vi điều khiển.

Sản phẩm thiết kế ra tạo ra một mô hình nhỏ giúp sinh viên khảo sát và khám phá tài nguyên của họ vi điều khiển 8051. Bên cạnh đó, tạo tiền đề cho việc nghiên cứu sau này.

Hiểu thêm về khả năng giao tiếp của vi điều khiển với thiết bị ngoại vi như LCD, việc điều khiển tốc độ động cơ cung cấp các ứng dụng thực tế. Giúp em có thể tiếp tục khám phá và tạo ra những sản phẩm có ích cho cuộc sống.

* Nhược điểm:

Bên cạnh các ưu điểm thì vẫn còn một số hạn chế:

Việc lựa chọn các linh kiện tối ưu cho mạch, chưa khám phá hết tiềm năng của 8051 và ngôn ngữ lập trình C.

Việc tăng tần số cho xung vuông (xung PWM) gặp khó khăn khi tăng tần số cao lại không thể điều khiển được tốc độ động cơ.

Chỉ tìm được giải pháp tăng thêm giới hạn tốc độ động cơ bằng cách tăng thêm điện áp nạp cho động cơ

Còn gặp khó khăn trong việc thi công mạch thực tế, chưa thành thục.

TÀI LIỆU THAM KHẢO

[1] Giáo trình Vi Xử Lý-Vi Điều Khiển họ 8051 - Pgs.Ts.Nguyễn Hữu Công

[2] Giáo trình vi điều khiển 89S52 - Nguyễn Đình Phú

[3] <http://dammedientu.vn/bai-10-lap-trinh-hien-thi-lcd-16x2-voi-8051/>

[4]<https://mobitool.net/dieu-khien-toc-do-dong-co-dc-bang-pwm-su-dung-vi-dieu-khien-8051.html>