

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH
KHÓA 10

ĐINH ĐẠT THÔNG

ĐỒ ÁN MÔN HỌC
THIẾT KẾ LED QUAY

TP. HỒ CHÍ MINH, 2018

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH
KHÓA 10

ĐINH ĐẠT THÔNG – 15520853

ĐỒ ÁN MÔN HỌC
THIẾT KẾ LED QUAY

GIẢNG VIÊN HƯỚNG DẪN
PHAN ĐÌNH DUY

TP. HỒ CHÍ MINH, 2018

MỤC LỤC

MỞ ĐẦU	4
Chương 1. TỔNG QUAN.....	5
1.1. Khái niệm LED quay	5
1.2. Ứng dụng LED quay.....	6
1.3. Các từ khóa liên quan	6
Chương 2. CƠ SỞ LÝ THUYẾT	7
Chương 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	10
3.1. Ý tưởng và giải thuật phần mềm	10
3.2. Lắp ghép phần cứng	24
Chương 4. KẾT QUẢ THỰC NGHIỆM	39
Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	40
5.1. Kết luận	40
5.2. Hướng phát triển.....	40
TÀI LIỆU THAM KHẢO	41

MỞ ĐẦU

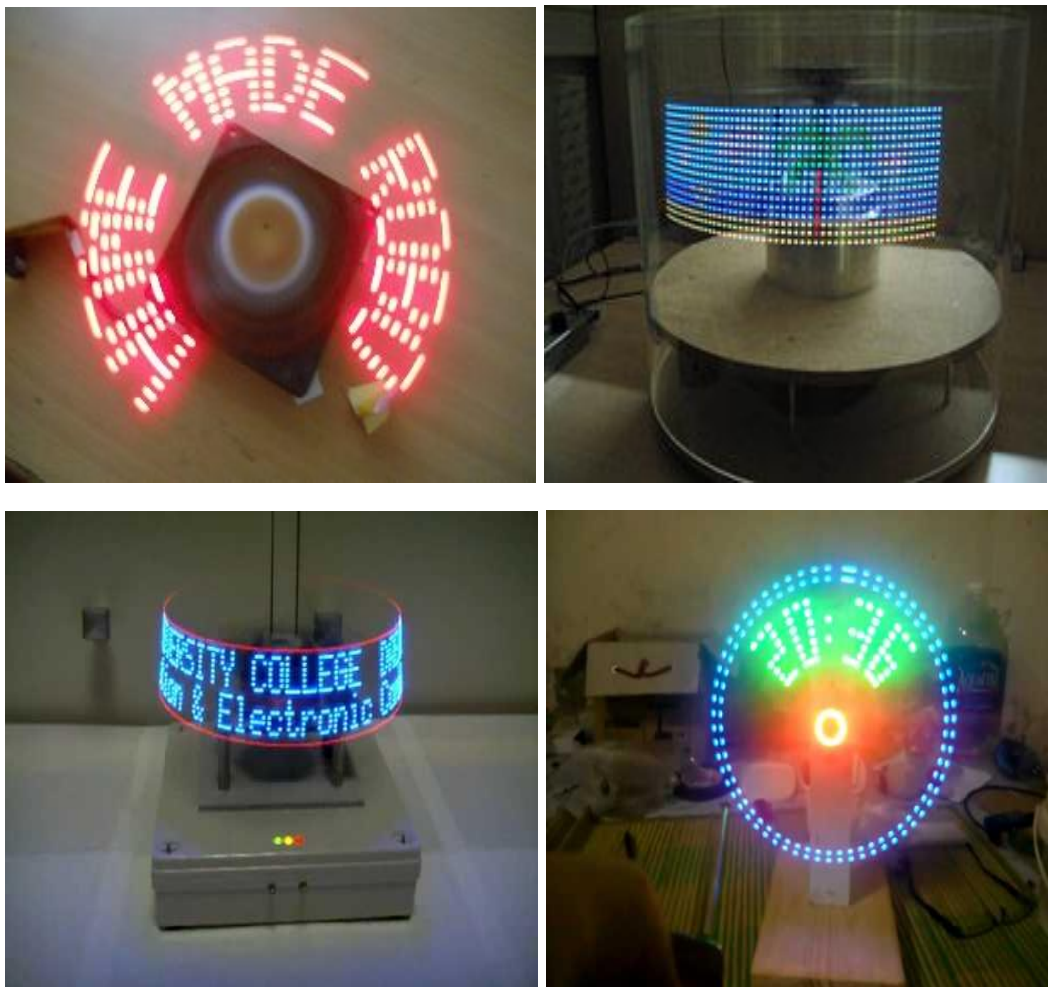
- Kính chào thầy và người đọc, ở bài báo cáo này, em muốn viết như một bài tường thuật lại quá trình làm đồ án, cũng như hướng dẫn thật cụ thể cho các bạn chưa biết bắt đầu từ đâu cũng có thể tiếp thu và hiểu (hoặc nếu như đồ án của em vẫn không hoàn thiện trước hạn nộp, các bạn đọc vẫn có thể tiếp tục cải thiện thiếu sót của em). Em xin phép được xưng hô là “mình”. Vì mục đích hiển thị rõ ràng bố cục cho người đọc dễ theo dõi, nên kích cỡ chữ ở bài này khá nhỏ, khi đọc bằng máy tính, mọi người nên phóng to màn hình để đọc rõ hơn. Mình nghĩ thầy và các bạn nên đọc trên máy hoặc in màu, một phần vì hình ảnh trắng đen trên bản photo rất khó nhìn, một phần là không thể truy cập link ngay lúc đọc được.
- Bài viết khá dài vì mình cố gắng viết thật chi tiết. Nếu người đọc mới tiếp cận với việc làm LED quay, mình khuyên nên đọc kỹ bài này để biết được những sai sót và khó khăn mình gặp phải, vừa tránh các lỗi giống mình, vừa rút ngắn được thời gian tìm hiểu - thực hành.
- Trong quá trình làm đồ án, mình đã tham khảo rất nhiều nguồn và đúc kết lại kiến thức để viết bài báo cáo này. Mình cam kết đây là bài viết tự viết, không sao chép hay dịch từ tài liệu nước ngoài.
- Các hình ảnh trong bài viết này hầu hết mình tự vẽ/chụp, một vài hình thì lấy từ Google.
- Nguồn tham khảo của mình chủ yếu là tra Google, rất nhiều bài hướng dẫn cũng như video dành cho chủ đề này bằng nhiều ngôn ngữ khác nhau. Hướng dẫn của mỗi nguồn có điểm giống lẫn điểm khác, phụ thuộc vào tác giả.

Chương 1. TỔNG QUAN

1.1. Khái niệm LED quay

- Trước khi bắt tay vào làm, ta cần tìm hiểu LED quay là gì, ứng dụng thực tế của nó, và các kiến thức cần có để thiết kế nó.
- Đối với những ai chưa tìm hiểu hay nghe về LED quay thì mình liệt kê một số ví dụ hình ảnh lẫn video cho mọi người xem dưới đây

+ *Hình ảnh:*



+ *Video (truy cập vào link):*

https://www.youtube.com/watch?v=AxO3RVG_35E

<https://www.youtube.com/watch?v=zGymLtFieWo>

<https://www.youtube.com/watch?v=sUrFQFgQutg>

<https://www.youtube.com/watch?v=2bskUpQkSjc>

- Sau khi quan sát từ các ví dụ trên, các bạn có thể hình dung LED quay đơn giản giống như một bộ cánh quạt đang quay với động cơ quay nhanh, trên cánh quạt có đính thêm các đèn LED xếp theo hàng/cột, và bằng cách nào đó, nếu ta có thể khiến cho từng đèn LED sáng tắt trong thời gian nhất định, thì khi động cơ quạt quay đủ nhanh (nhanh bao nhiêu mình sẽ nói chi tiết ở phần sau), ánh sáng nhấp nháy của các LED trên cánh quạt khiến ta cảm thấy chúng tạo ra hình ảnh. Tên gọi “LED quay” cũng từ đó mà ra.

1.2. Ứng dụng của LED quay

- LED quay có thể được dùng để hiển thị các thông tin cần thiết ở ga tàu, trạm xe buýt, đồng hồ số, đồng hồ quay. Vì tính đa chiều của hình ảnh hiển thị bởi LED quay, người nhìn có thể quan sát và đọc thông tin từ nhiều hướng. Thay vì muốn hiển thị hình ảnh bằng ma trận LED (giống bảng quảng cáo LED bạn thường thấy ở các cửa tiệm), ta có thể tiết kiệm số lượng đèn LED và chi phí nếu dùng LED quay để hiển thị. Sau đây là video ví dụ:

https://www.youtube.com/watch?v=hu_eY-Oj1-c

1.3. Các từ khóa liên quan

Fan LED; POV (Persistence Of Vision); LED quay; AV Clock; Spinning Display; Propeller LED; POV Display; Rotating LED Display;... .

Chương 2. CƠ SỞ LÝ THUYẾT

- Trước hết, để có thể hiểu được những nội dung tiếp theo, người đọc **bắt buộc** phải trang bị trước những kiến thức sau đây:

+ *Lý thuyết mạch điện, Các thiết bị điện tử, Kiến trúc máy tính, Vi xử lý – Vi điều khiển (vi điều khiển họ 8051), Nhập môn mạch số, Nhập môn lập trình, khả năng sử dụng ngôn ngữ Assembly, khả năng sử dụng phần mềm Proteus để mô phỏng mạch điện, và kỹ năng làm mạch in +*

- Đối với cấu trúc máy tính cơ bản và mạch số, cấu tạo thanh ghi, RAM, ROM, cấu tạo ALU đến CPU cơ bản, các bạn có thể tham khảo chuỗi video dưới đây (tiếng Anh)

<https://www.youtube.com/watch?v=tpIctyqH29Q&list=PL8dPuuaLjXtNIUrzYH5r6jN9uIIgZBpdo>

- Mình thấy chuỗi video này rất hay, đặc biệt dành cho những ai luôn tự hỏi bản chất của máy tính là gì, hay bên trong máy tính rốt cuộc là những gì, chuỗi video đã là tiền đề, là nền tảng của mình trước khi mình thực sự học các môn trong trường đại học. Nếu bạn đủ kiên nhẫn xem đến clip thứ 8, mình đảm bảo bạn sẽ nhìn ra được hầu như mọi thứ diễn ra bên trong vi điều khiển.

- Đối với kiến thức về vi điều khiển, mạch điện sẽ mất rất nhiều thời gian, đặc biệt là khả năng lập trình ngôn ngữ Assembly, kiến thức về Assembly. Mình đã mất khá nhiều thời gian để tự tìm hiểu mọi thứ cần thiết, quan trọng ta phải luyện đi luyện lại. Đối với mình, khi làm đồ án này, mình vẫn chưa viết một chương trình thực thụ nào trên C, nên mình nghĩ chuỗi video trên sẽ giúp ích rất nhiều cho việc hiểu về ngôn ngữ Assembly. Và dĩ nhiên đối với những ai theo ngành công nghệ thông tin, các bạn cần chú trọng những bước cơ bản khi lập trình, cách tư duy và cách thể hiện giải thuật.

- Các bạn chỉ nên tiếp tục bài đọc này sau khi đã hoàn thành và nắm thật vững các kiến thức trên, nếu thiếu một trong những kiến thức mình liệt kê ở trên, các bạn sẽ rất khó nắm bắt được những gì được viết bên dưới.

- LED quay là một thiết bị dùng để hiển thị hình ảnh nào đó. Thiết bị này tương tự như một chiếc quạt máy, gồm một động cơ quay, một bảng mạch in PCB (Printed Circuit Board) đóng vai trò như một cánh quạt, trên bảng mạch đó tích hợp các đèn LED, các con chip vi điều khiển, thiết bị điện tử nhằm cho mục đích điều khiển sự sáng tắt các đèn LED theo từng thời điểm nhất định.

- Khi động cơ quay đủ nhanh, ánh sáng của các đèn LED sẽ tạo ra hiệu ứng lưu ảnh trong mắt, khiến mắt ta bị đánh lừa rằng các ánh sáng đó tạo ra một hình ảnh. Hiện tượng lưu ảnh trong mắt - tiếng Anh là Persistence Of Vision - tạm dịch là sự duy trì của thị giác, hiện tượng này xảy ra khi mắt người nhìn vào một điểm ảnh, và một chùm sáng ảm - hiện tại điểm ảnh đó với tần số ảm - hiện cực nhanh, cụ thể là 25 lần ảm và hiện trong một giây, thì lúc này, ta sẽ không cảm nhận được sự ảm - hiện của chùm sáng tại điểm ảnh, và sẽ tưởng rằng chùm sáng luôn hiện hữu tại điểm ảnh đó. Tưởng tượng bạn dán bất cứ gì phát sáng màu đỏ lên cánh quạt bất kỳ, giả sử động cơ quạt quay thật nhanh, thứ ta thấy được sẽ là một vòng tròn màu đỏ.

*** Loại LED quay mình nêu trên là loại mà mắt phẳng của hình ảnh hiển thị song song với mặt đất (đúng hơn là mắt phẳng mà motor đặt lên). Còn một loại khác là loại mắt phẳng của hình ảnh vuông góc với mặt đất (đã nêu trong các ví dụ ở mục tổng quan). Mỗi loại sẽ có cách thiết kế giải thuật khác nhau. Trong bài này, mình chọn loại song song. ***

- Tóm lại ta sẽ cần một bộ động cơ quay, một bảng mạch in tự thiết kế, trên bảng mạch đính các đèn LED (số lượng LED và cách sắp xếp LED tùy các bạn chọn, phụ thuộc vào hình ảnh các bạn muốn hiển thị).

- Như mình đã đề cập, ta phải làm cách nào đó để các đèn led sáng tắt theo quy trình nhất định và tạo ra hình ảnh khi động cơ quay. để thực hiện việc cho đèn LED sáng tắt một cách có trình tự, ta có rất nhiều cách thiết kế sơ đồ mạch giúp ta thực hiện được mục đích, có thể sử dụng các bộ chip tạo dao động, các linh kiện và mạch điện liên quan đến việc tạo độ trễ, Trong đó, lựa chọn tốt nhất mình nghĩ là dùng một chip vi điều khiển bởi chức năng điều khiển linh hoạt của nó.

- Vì bài này là đồ án môn học vi xử lý - vi điều khiển, mình chỉ mới làm quen và sử dụng vi điều khiển họ 8051, các chip ta có thể chọn là AT89C51 hoặc AT89S51, AT89C52 hoặc AT89S52 để lập trình, cấu tạo AT89C52 / AT89S52 không khác nhiều so với chip AT89C51 / AT89S51 ngoài việc có thêm Timer, ... (các bạn có thể tham khảo datasheet của từng chip và so sánh, mình sẽ không đề cập ở đây). Các bạn chọn chip nào cũng được. Mình chọn chip AT89S52. AT89C52 có vẻ khó nạp Code hơn.

- Ta sẽ thiết kế mạch, lập trình cho vi điều khiển và mô phỏng mạch trên phần mềm máy tính. Có nhiều phần mềm phục vụ cho mục đích này. Ở đây mình sử dụng phần mềm Proteus 8.0 Professional (có sẵn phần soạn thảo và biên dịch Code luôn). Các bạn tra Google và tải về.

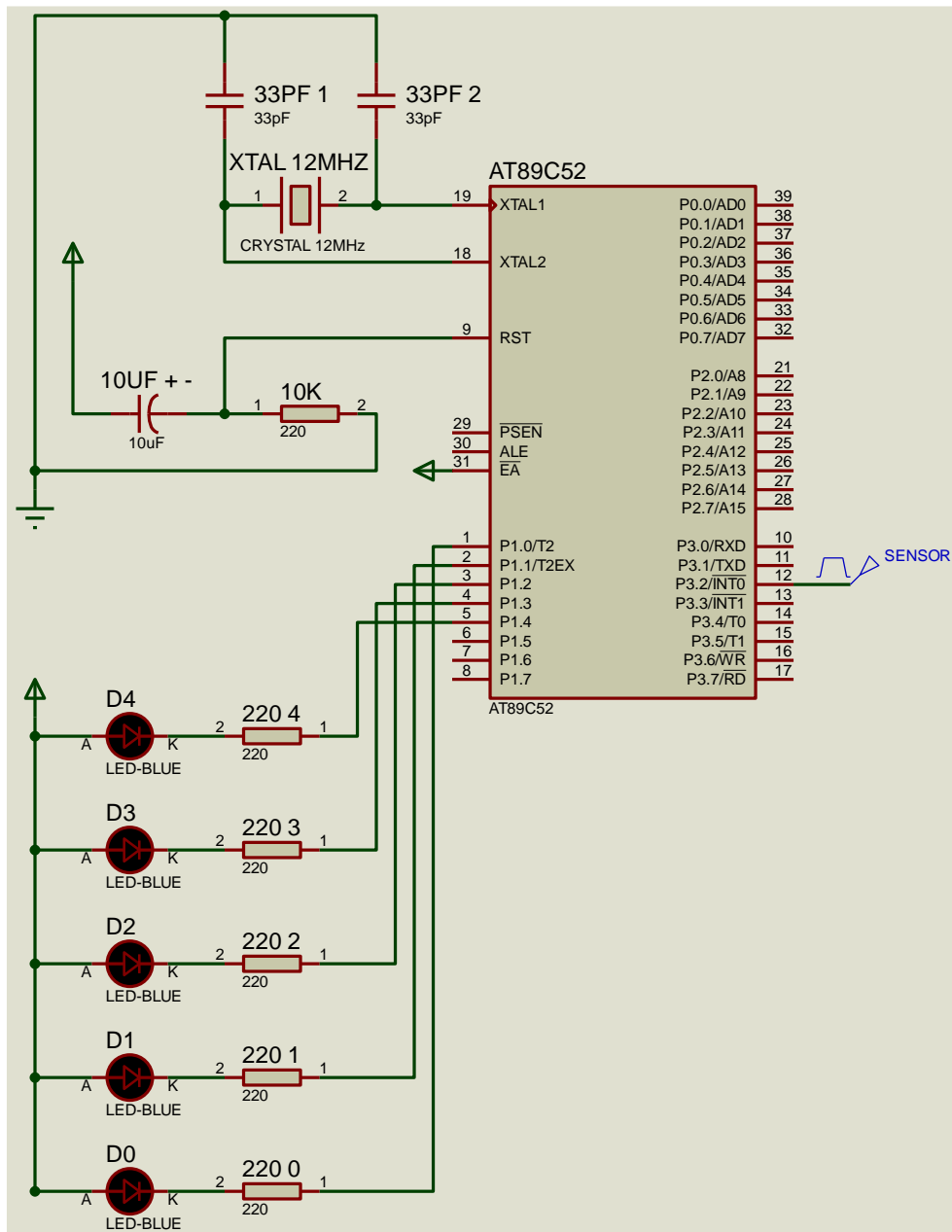
- Đến đây, ta đã xác định một số thành phần chính chuẩn bị cho các bước kế tiếp. Ở những phần sau, mình sẽ đi sâu hơn về giải thuật phần mềm và cách lắp ghép phần cứng.

- Trong quá trình thực hành, mình hình dung giải thuật, mô phỏng sơ đồ + mạch in trên Proteus trước, sau đó viết chương trình, rồi đến mô phỏng ý tưởng, mua linh kiện, hàn mạch, Loại ngôn ngữ mình sử dụng là ngôn ngữ Assembly cho 8051, cũng là yêu cầu của đồ án này. Vì vậy, mình trình bày giải thuật phần mềm trước tiên, sau đó sẽ đến phần lắp ghép phần cứng.

Chương 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

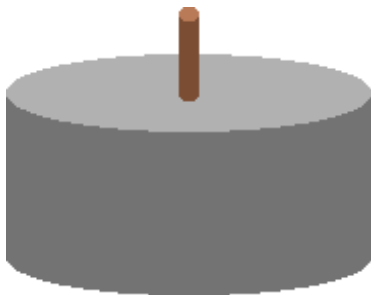
3.1. Ý tưởng và giải thuật phần mềm

- Mục này mình đưa ra ý tưởng rồi từ đó dẫn mọi người đến một ít giải thuật chương trình cho vi điều khiển.
- Trước khi bước vào giải thuật, đầu tiên mình mô phỏng sơ đồ trên Proteus:



- Quan sát bên trái vi điều khiển AT89C52, từ chân EA (External Access) trở lên là các linh kiện cần có để vi điều khiển hoạt động, mình đặt tên tụ điện phân cực 10uF là “10UF + -“ để phân biệt với tụ không phân cực (sẽ có lợi khi đọc và mô phỏng mạch in). Từ chân EA trở xuống là 5 đèn LED nối với điện trở như sơ đồ, với P1.0 = 0 thì LED sáng, P1.0 = 1 thì LED tắt. (số lượng LED phụ thuộc vào hình ảnh bạn muốn hiển thị, ở đây mình muốn hiển thị dãy chữ cái in hoa, mình chọn 5 LED là đủ)
- Quan sát bên phải có một tín hiệu dạng xung nối với chân số P3.2 (ngắt ngoài 0), mình đặt tên là SENSOR (cảm biến), các bạn đừng quan tâm đến tín hiệu này vội, mình sẽ nói kỹ hơn sau.
- Vẽ sơ đồ xong, ta đi vào thiết kế giải thuật phần mềm. Tuy chưa lắp ghép phần cứng hoàn chỉnh, nhưng ta vẫn có thể suy nghĩ ra giải thuật thông qua hình vẽ đơn giản. Sau đây là hình 2D mình vẽ mô phỏng phần cứng tương đối của một chiếc LED quay (không vẽ chi tiết).

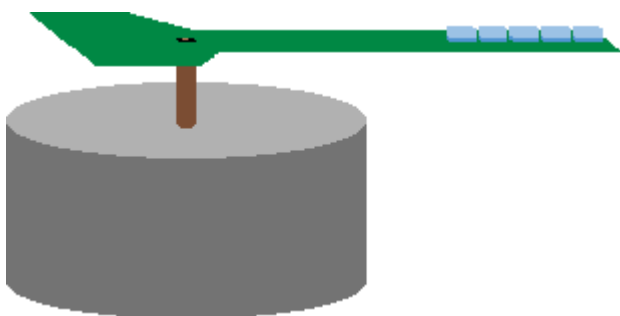
đây là motor



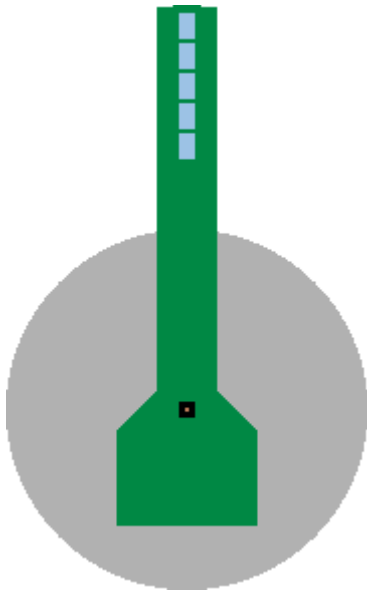
đây là bảng mạch (5 LED)



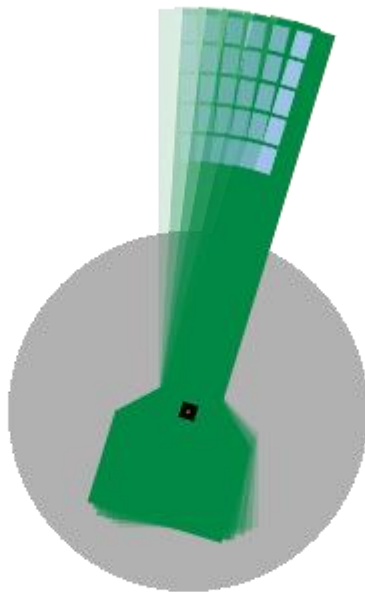
ghép lại



nhìn từ trên xuống

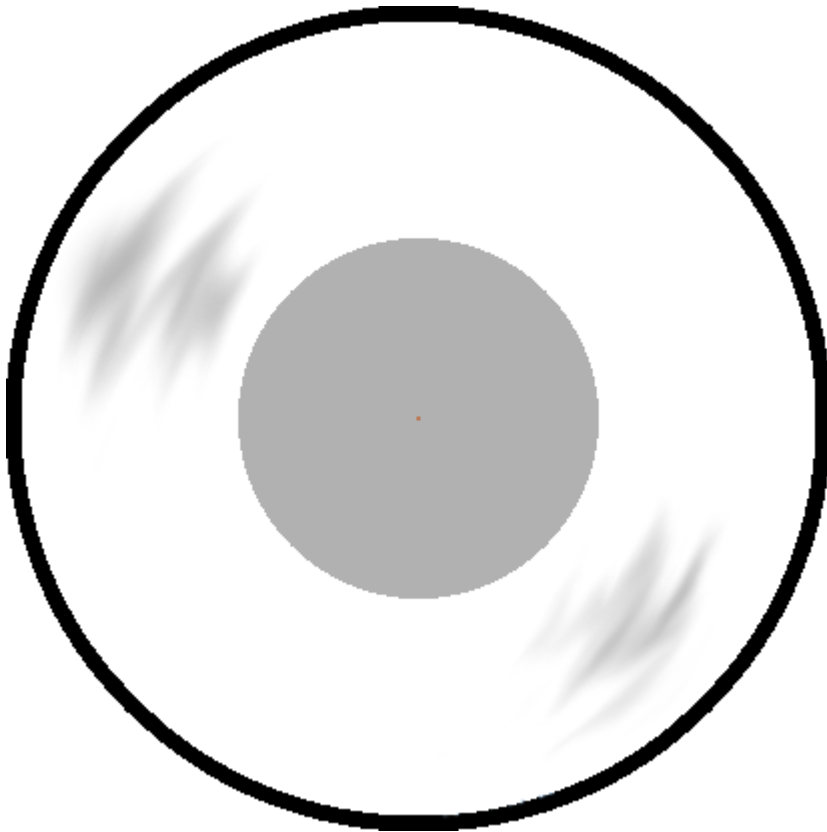


lúc motor bắt đầu quay (giả sử các LED luôn sáng, hình này mô phỏng tàn dư ánh sáng của các LED khi motor quay theo chiều kim đồng hồ)

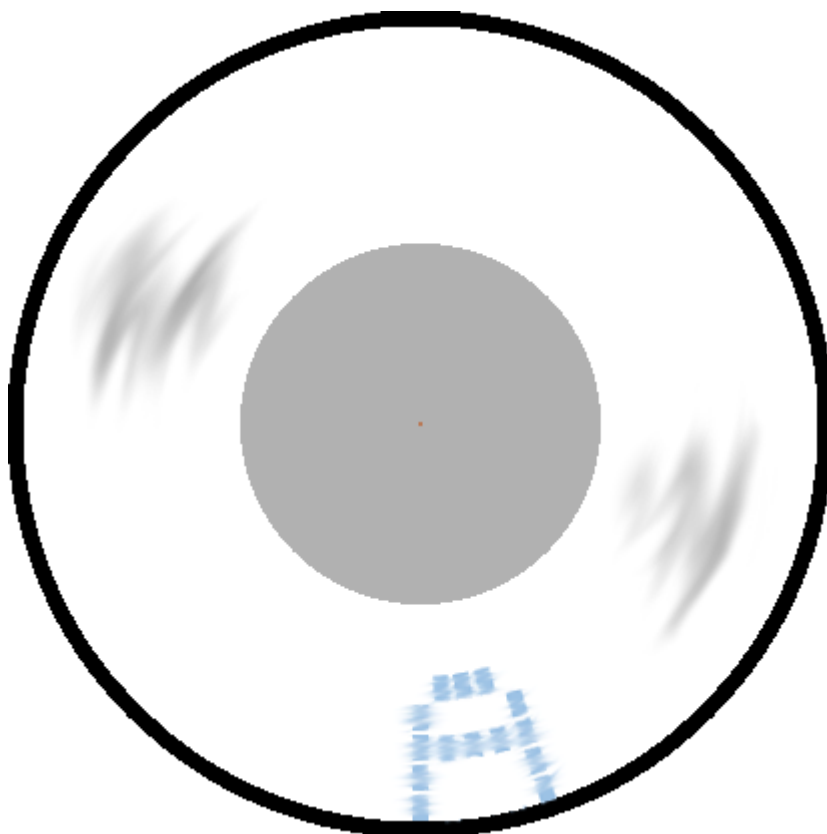


motor quay nhanh quá không thấy bảng mạch nữa

(giả sử các LED bị tắt hết, vòng tròn màu đen tượng trưng cho viền của bảng mạch khi xoay)



Điều khiển sự sáng tắt của từng LED để hiển thị chữ A



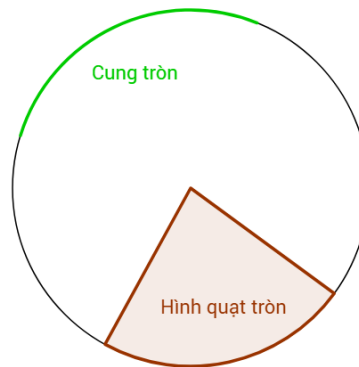
- Quan sát chữ A và vòng tròn màu đen, ta thấy rằng chữ A được tạo bởi 5 cột LED, mỗi cột được đánh số phía trên như sau (ô xanh là đèn sáng, ô trắng là đèn tắt):

1	2	3	4	5

- Khi quay theo chiều kim đồng hồ, bảng mạch sẽ quét từ cột 5 đến cột 4, cứ thế lần lượt và cuối cùng là cột 1. Độ rộng mỗi cột LED sẽ lớn hơn hoặc bằng với độ rộng

của một đèn LED tùy vào bạn muốn độ rộng của chữ cái là bao nhiêu. Ví dụ một đèn LED rộng 3mm, một cột LED bạn muốn bằng 6mm, thì chữ A cấu tạo từ 5 cột như trên sẽ rộng bằng $6 \times 5 = 30\text{mm}$.

- Mỗi cột có thể xem như một hình quạt (ta gọi là cột LED để dễ hiểu hơn).



- Quan sát hình trên. Ta nhận ra được, độ rộng cung tròn của hình quạt sẽ quyết định số lượng hình quạt có thể vẽ (với một đường tròn với bán kính cho trước). Vậy, khi suy ra giải thuật, ta cần phải biết chiều dài bán kính đường tròn (bán kính được đo từ tâm đường tròn đến LED chính giữa, nếu bán kính được đo từ tâm đến một trong hai LED ngoài rìa, chữ cái sẽ bị xòe ra nhiều hơn), xác định được độ rộng của đèn LED, từ đó xác định độ rộng của một cột LED (kích cỡ cung tròn của hình quạt), từ đó ta xác định được số lượng cột LED (số lượng hình quạt bằng nhau có thể vẽ).

- Cách tính các thông số cần thiết dưới đây sẽ giúp bạn hiểu rõ hơn nhiều:

- Như đã nói ở các mục trước, mắt ta sẽ bị đánh lừa khi nhìn một điểm ảnh với chùm sáng nhấp nháy 25 lần trong 1 giây. Do đó motor phải quay 25 vòng trong 1 giây để tạo hình ảnh đánh lừa mắt.

- Giả sử chọn motor 1500 RPM (Revolution Per Minute) quay 1500 vòng trong một phút, tức 25 vòng trong một giây

1500 vòng : 60 giây

1 vòng : ? giây

Suy ra

1 vòng : $60 / 1500 = 0.04 \text{ (s)} = 40 \text{ (ms)}$

- Vậy thời gian Motor quay được 1 vòng là 40 mili giây

- Bán kính của vòng tròn quay sẽ là chiều dài từ đèn LED chính giữa đến tâm vòng tròn

- Bán kính của vòng tròn quay (radius) : $r = 6 \text{ (cm)}$

- Chu vi của vòng tròn quay (Perimeter) : $P = 2\pi r = 2 \times 3,14 \times 6 = 37.68 \text{ (cm)}$

- Độ rộng của một đèn LED : $w = 3 \text{ (mm)} = 0.3 \text{ (cm)}$

- Độ rộng của một cột LED : $W = n \times w = 1 \times w = 0.3 \text{ (cm)}$

n tùy chọn, mình chọn $n = 1$

- Số lượng cột LED có thể có : $x = P / W = 37.68 / 0.3 = 125.6 = 125 \text{ (cột)}$

- Vậy xem như thiết bị quay 1 vòng mất 40 mili giây, hay nói cách khác là quét 125 cột trong khoảng thời gian 40 mili giây

- Tính thời gian hiển thị của 1 cột LED cũng là thời gian trì hoãn giữa các cột

- Thời gian hiển thị của 125 cột : $T = 40 \text{ (ms)}$

- Thời gian hiển thị của 1 cột : $t = T / x = 40 / 125 = 0.32 \text{ (ms)} = 320 \text{ (}\mu\text{s)}$

- Ta tự lựa chọn rằng: mỗi cột LED chứa 5 đèn LED và mỗi chữ cái sẽ được hiển thị bằng 5 cột LED liên tiếp như sau

1	2	3	4	5

- Thời gian hiển thị 5 cột LED liên tiếp : $t_{5 \text{ cột}} = 5 \times t = 5 \times 320 = 1600 \text{ (}\mu\text{s)}$
- Độ rộng của 5 cột LED liên tiếp : $W_{5 \text{ cột}} = 5 \times W = 5 \times 0.3 = 1.5 \text{ (cm)}$
- Số lượng ký tự có thể hiển thị : $z = P / W_{5 \text{ cột}} = 37.68 / 1.5 = 25.12$
= 25 (ký tự)

- Nếu đã hiểu được những khái niệm và thông số trên, các bạn có thể hình dung được giải thuật chương trình đơn giản như sau: (giải thuật này là giải thuật chung, các dòng lệnh khởi tạo thanh ghi chức năng mình sẽ lược bỏ)

Chương trình chính

Bắt đầu chương trình

Xuất dữ liệu cột 1 ra Port

Gọi hàm tạo trễ

Xuất dữ liệu cột 2 ra Port

Gọi hàm tạo trễ

Xuất dữ liệu cột 3 ra Port

Gọi hàm tạo trễ

.....

.....

Xuất dữ liệu cột cuối cùng ra Port

Gọi hàm tạo trễ

Quay về lệnh “Xuất dữ liệu cột 1 ra Port”

Chương trình con tạo trễ

Vòng lặp hữu hạn (hoặc dùng Timer)

Thoát chương trình con

- Một giải thuật hay hơn:

Chương trình chính

Bắt đầu chương trình

Tạo mảng dữ liệu cần xuất (sắp xếp sao cho phù hợp với chiều quay của motor + chiều muốn hiển thị)

Gọi hàm xuất lần lượt từng byte dữ liệu ra Port

Quay về lệnh “Gọi hàm xuất dữ liệu”

Chương trình con xuất từng byte dữ liệu

Con trỏ chỉ vào byte đầu tiên cần xuất

Xuất byte ra Port

Gọi hàm tạo trễ

Con trỏ chỉ vào byte kế tiếp cần xuất

Quay về lệnh “Xuất byte ra Port” nếu chưa xuất hết dữ liệu

Thoát chương trình con nếu đã xuất hết dữ liệu

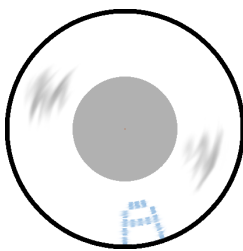
Chương trình con tạo trễ

Vòng lặp hữu hạn (hoặc dùng Timer)

Thoát chương trình con

- Nhược điểm của hai giải thuật trên là các ký tự sẽ không thể hiển thị ổn định, lý do nằm ở sai số khi ta tính số lượng cột LED, dẫn đến việc dư một ít thời gian sau khi hiển thị cột cuối cùng, do đó nội dung hiển thị không nằm cố định một chỗ mà sẽ xô dịch liên tục sau mỗi vòng quay. Chính vì thế, chương trình cần có điểm xuất phát cụ thể (hay còn gọi là điểm gốc/điểm tham chiếu). Sử dụng một điểm làm điểm gốc để vi điều khiển biết được rằng khi bảng mạch quay trở về điểm gốc thì vi điều khiển sẽ quay về lệnh hiển thị cột LED đầu tiên. Điều này giúp nội dung hiển thị bằng LED quay ổn định rất nhiều.

- Đây là lúc ta quay lại tín hiệu xung SENSOR đã nói ở đầu mục, tín hiệu đó đóng vai trò ngắt vi điều khiển mỗi khi bảng mạch quay được một vòng.



- Xem xét hình trên, bất cứ điểm nào trên đường tròn đen hoặc những điểm thuộc đường tròn đồng tâm với đường tròn đen cũng có thể là điểm gốc. Cái này các bạn tự chọn. Khi viết chương trình, ta chưa cần phải quan tâm đến việc thiết kế mạch ngắt như thế nào hay chọn cảm biến loại nào, chỉ cần biết rằng ta có sử dụng dạng

ngắt ngoài kích cạnh là được. Việc chọn vị trí điểm gốc sẽ quyết định phần nào vị trí của cảm biến trên bảng mạch (hoặc các bạn có thể tự chế biến nếu khéo léo).

- Mình thiết kế một giải thuật có sử dụng ngắt ngoài như sau:

Chương trình chính

Bắt đầu chương trình

Cho vi điều khiển biết ta có sử dụng ngắt ngoài kích cạnh xuống

Tạo mảng dữ liệu cần xuất

Không làm gì (chờ ngắt xảy ra)

Quay về lệnh Không làm gì

Chương trình con phục vụ ngắt ngoài

Gọi hàm xuất từng byte dữ liệu ra Port

Thoát chương trình con

Chương trình con xuất từng byte dữ liệu

Con trỏ chỉ vào byte đầu tiên

Xuất byte ra Port

Gọi hàm tạo trễ

Chỉ vào byte kế tiếp

Quay về lệnh “Xuất byte ra Port” nếu chưa xuất hết dữ liệu

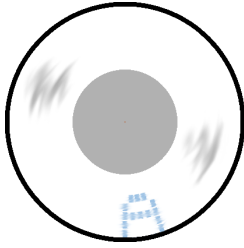
Thoát chương trình con nếu đã xuất hết dữ liệu

Chương trình con tạo trễ

Vòng lặp hữu hạn (hoặc dùng Timer)

Thoát chương trình con

- Dĩ nhiên là trong quá trình viết Code Assembly, các bạn cần các dòng lệnh khởi tạo các thanh ghi đặc biệt nhằm kích hoạt các chức năng như Timer, UART, ... của chip 8051.
- Nhược điểm của cả ba giải thuật trên là cả ba đều phụ thuộc vào tốc độ động cơ, nên sẽ khó khăn cho ta đi tìm động cơ có tốc độ phù hợp. Ưu điểm là dễ hiểu cho các bạn mới bắt đầu.
- Khi đã hiểu rõ đến đây, các bạn có thể tham khảo hoặc tự nghĩ ra nhiều giải thuật khác, giải thuật mình thấy nên tham khảo là giải thuật đo tốc độ động cơ để từ đó tự suy ra giá trị cho hàm tạo trễ phù hợp. Trên Youtube đã có một anh người Việt hướng dẫn.
- Đi đến bài của mình, mình muốn hiển thị hai nội dung, một là dòng chữ “HELLO”, hai là dòng chữ “WELCOME TO UIT”, hai dòng chữ sẽ luân phiên nhau hiển thị sau một khoảng thời gian nhất định, vì vậy mình cải tiến Code một ít. Đến đây thầy và mọi người hãy vừa đọc Code của mình vừa đọc tiếp bài báo cáo để xem cách mình lưu dữ liệu vào ROM và truy xuất. (tập tin PROPELLER LED CODE trong thư mục SOURCE CODE đính kèm)
- **Đặc biệt cần chú ý chiều quay của motor và chiều hiển thị của chữ cái để sắp xếp dữ liệu xuất ra cho hợp lý. Chiều quay motor mình chọn là chiều thuận kim đồng hồ, chiều chữ hiển thị ngược chiều với chữ A dưới đây:**



- Hai hình bên dưới thể hiện mảng dữ liệu được lưu trong ROM. Dựa vào cách tính ở những trang trước, ta có được thông số dãy ký tự muốn hiển thị tối đa là 25 ký tự (125 cột LED).

HELLO

WELCOME TO UIT

- Giải thích mỗi hình:

- + Mỗi hình gồm 8 hàng, 125 cột.
- + Mỗi ô tượng trưng 1 bit dữ liệu.
- + Mỗi cột tượng trưng một ô nhớ 8 bit.
- + Số lượng cột tượng trưng cho kích thước của mảng dữ liệu. 125 cột tượng trưng 125 byte.
- + Ô màu xanh tương ứng LOGIC 0 (LED sáng) vì cách nối LED với PORT.
- + Ô màu trắng tương ứng LOGIC 1 (LED tắt) vì cách nối LED với PORT.
- + Theo cột, từ trên xuống dưới: giá trị bit P1.7 => bit P1.0 tương ứng với 8 ô vuông trong 1 cột.
- + Các bạn tùy ý lựa chọn vùng nhớ chứa dữ liệu. Trong chương trình này mình chọn vùng nhớ bắt đầu từ ô nhớ thứ 1000 đến ô nhớ thứ 1125 trong ROM để lưu mảng “HELLO”, từ ô nhớ thứ 1126 đến 1252 lưu mảng “WELCOME TO UIT”.

+ Theo hình của mảng “HELLO”, cột ngoài cùng bên phải tương ứng ô nhớ 1001 (đọc code sẽ rõ vì sao không phải 1000 mà là 1001, tùy cách người viết Code điều khiển con trở dữ liệu), hàng cuối cùng tương ứng ô nhớ thứ 1125. Trong Code của mình, xét mảng “HELLO”, mình xuất dữ liệu từ ô nhớ thứ 1125 ngược trở về ô nhớ thứ 1001, tức xuất từ cột ngoài cùng bên trái lần lượt đến cột ngoài cùng bên phải.

- Cuối cùng ta đã hoàn tất phần viết chương trình.

- Mọi người kiểm tra lại Code và sẵn sàng nạp Code vào vi điều khiển. (cách nạp Code mình không nói ở đây, các bạn tự tìm hiểu để nhớ lâu, quan trọng là hiểu cách thức truyền dữ liệu từ một thiết bị sang thiết bị khác)

3.2. Lắp ghép phần cứng

- *Đọc thêm: nếu ai định lựa chọn sử dụng AT89C2051 để bảng mạch nhỏ gọn và tiết kiệm chi phí hơn, thì nhược điểm của lựa chọn này là:*

+ *Chi phí cao khi mua máy nạp TL866 (khoảng 1 triệu đồng). Link hướng dẫn:*

<https://www.youtube.com/watch?v=H7ZVruYvXG4>

+ *Mất khá thời gian nếu tự làm mạch nạp. Link hướng dẫn:*

1. *http://dulieu.tailieuhocTap.vn/books/khoa-hoc-ky-thuat/dien-dien-tu/file_goc_772659.pdf*

2. *http://www.woe.onlinehome.de/e_projects.htm*

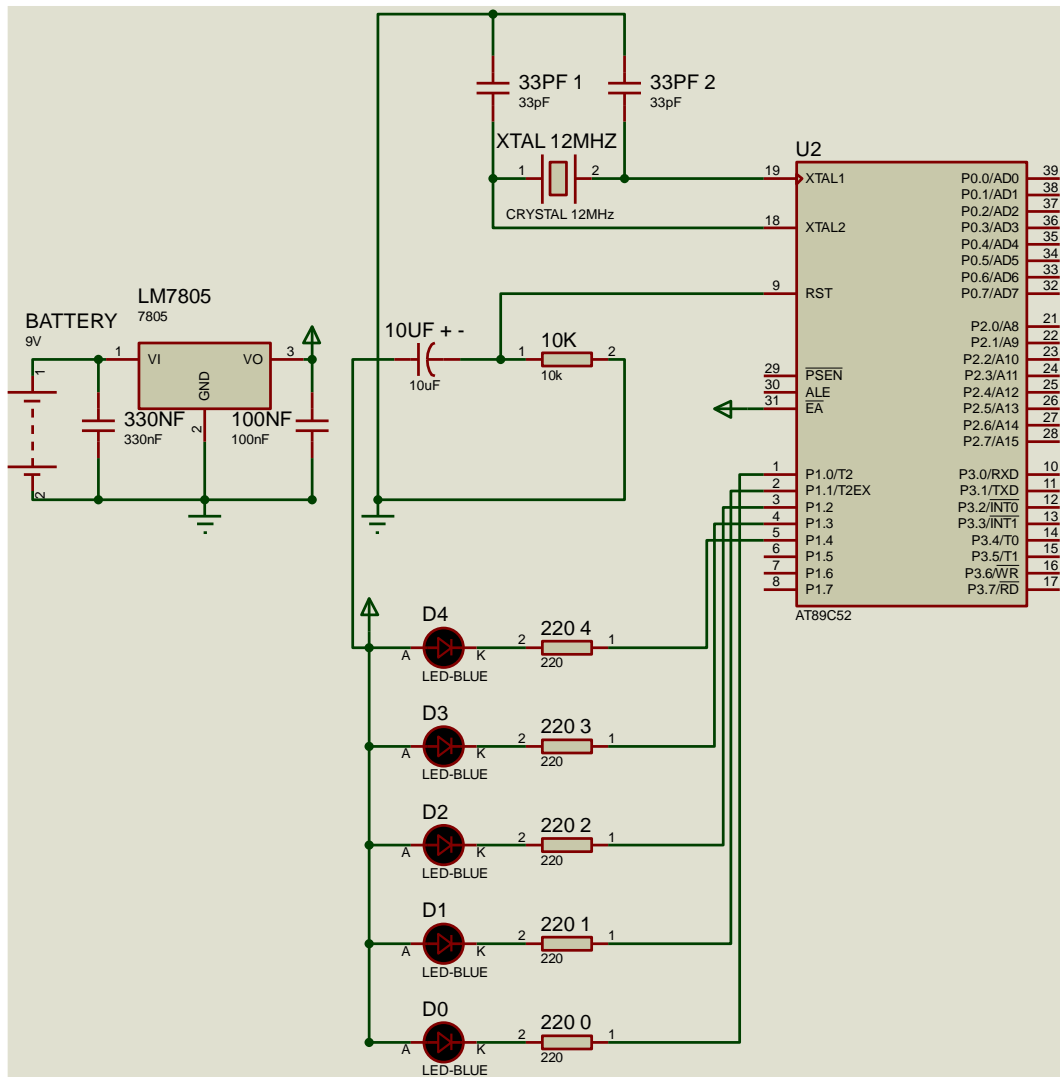
- *Còn AT89S2051 khá hiếm trên thị trường.*

- *Riêng mình thì chỉ dừng ở mức tìm hiểu.*

- Mục này mình trình bày từng bước xây dựng và lắp ghép phần cứng cho LED quay. Đây là phần nhiều thách thức khó khăn nhất, **hãy đọc thật kỹ để tránh các sai sót giống mình**. Giải thuật phần mềm càng hay, độ khó lắp ghép phần cứng càng giảm. Datasheet của từng Chip mình sẽ liệt kê ở cuối mục. (đọc Datasheet để làm chủ được chip ta sử dụng)
- Nếu tự làm mạch thủ công. Các bạn cần tự tìm hiểu và có kinh nghiệm làm mạch in. (cần có nhiều sai sót và rút kinh nghiệm)
- Mình chọn DC motor 6000 RPM 12V.



- Thêm vào sơ đồ IC nguồn LM7805. Mục đích là cung cấp cho toàn hệ thống điện áp 5V, dòng 1A. Tụ điện 330nF và 100nF đóng vai trò như một nguồn điện dự phòng, giúp ổn định điện áp nếu xuất hiện nhiễu. Tụ 330nF lọc nhiễu ở đầu vào, tụ 100nF lọc nhiễu ở đầu ra.



- Về phần ý tưởng thiết kế mạch ngắt, mình sẽ thiết kế một mạch làm sao mà đầu ra của mạch ngắt nối với chân P3.2, bit P3.2 mặc định bằng 1, và khi có tín hiệu ngắt xảy ra (tức cảm biến đi qua điểm gốc), đầu ra của mạch ngắt sẽ kéo chân P3.2 xuống điện áp mức 0 rồi sau đó quay về điện áp mức 1, ý tưởng của mạch ngắt này tương tự như mạch kéo áp có nút bấm nối với chân RESET của vi điều khiển. Có thể chọn lựa giữa cảm biến từ trường hoặc cảm biến quang,... . Tùy các bạn chọn lựa, trong bài này mình chọn cảm biến quang, tên Chip là HY860F (mình đã tra nhưng chưa tìm được Datasheet loại này), hình dạng như sau:



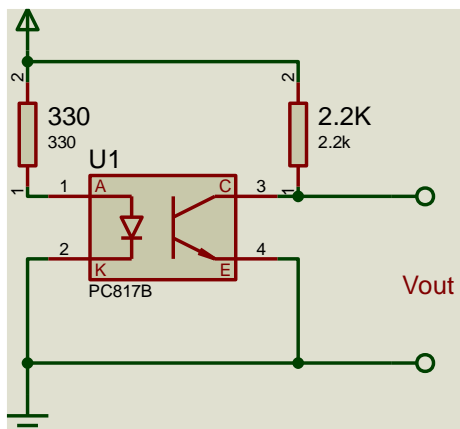
- Các bạn đọc 2 link hướng dẫn dưới đây để hiểu vì sao cảm biến có hình dạng này, cũng như cách nối và hoạt động của cảm biến này:

<http://www.martyncurrey.com/connecting-an-photo-interrupter-to-an-arduino/>

<http://heliosoph.mit-links.info/photointerrupter-schematic-details/>

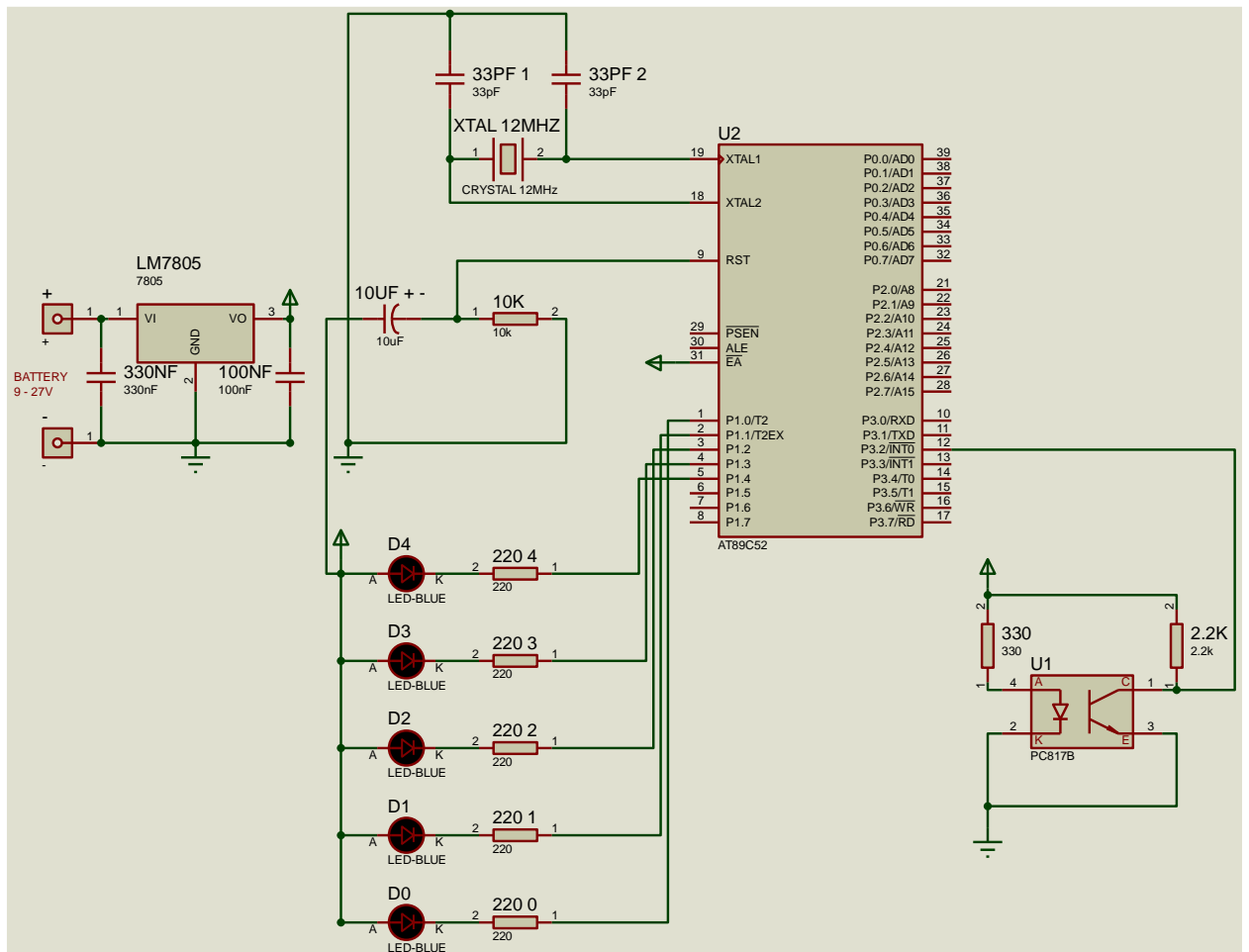
- Cảm biến chữ U có nhiều loại, các bạn chọn làm sao để giảm độ khó khi hàn mạch là được.

- Mình tiến hành vẽ sơ đồ mạch ngắt (đừng quên kiểm tra trên breadboard và đo điện áp đầu ra):



- Nguyên lý: mỗi lần đưa vật nào đó chắn tia hồng ngoại (tức đặt vật vào khe của cảm biến chữ U), điện áp đầu ra sẽ là mức 1, sau đó lấy vật chắn ra thì điện áp đầu ra đang ở mức 1 bị kéo xuống mức 0, tức lúc này xuất hiện sự chuyển từ 1 sang 0 (còn gọi là cạnh rơi – falling edge). (cách phân cực Transistor như trên gọi là Fixed Bias – tạm dịch là phân cực cố định)

- Nối mạch ngắt với vi điều khiển, thay nguồn là 2 đầu cắm, mục đích là để điều chỉnh ở mạch in:

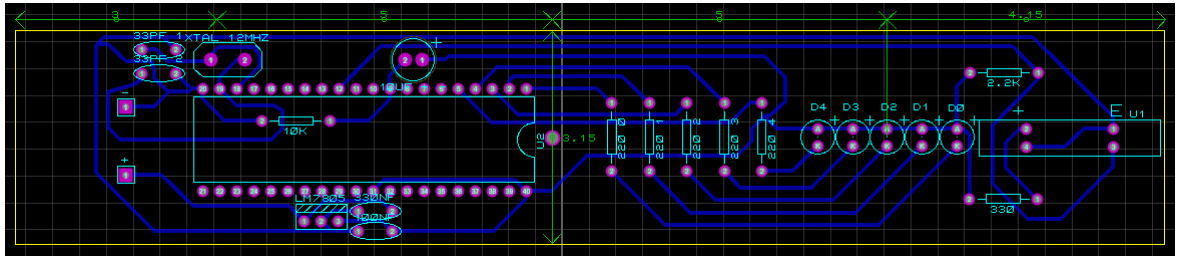


- Nhược điểm của bước thay nguồn là mạch không mô phỏng được.

- **Lưu ý lớn 1:** Phải tự làm mô hình mạch in cho cảm biến chữ U bởi thư viện Proteus không có sẵn. Không nên để cảm biến dốc ngược xuống đất vì sẽ gây khó khăn khi hàn mạch. (bởi lần đầu mình làm, mình để cảm biến dốc ngược xuống). Chọn lỗ khoan sao cho phù hợp với chân cắm IC LM7805 vì IC này có kích cỡ chân to hơn bình thường một ít.

- **Lưu ý lớn 2:** Nếu ai không muốn tháo Chip ra mỗi lần nạp Code mới thì phải thiết kế các chân cắm ngay phục vụ việc nạp Code ngay trên bảng mạch. Mạch của mình làm thì không có phần này.

- Sơ đồ mạch in:



(Hình khá nhỏ, xin lỗi những ai đọc bản photo)

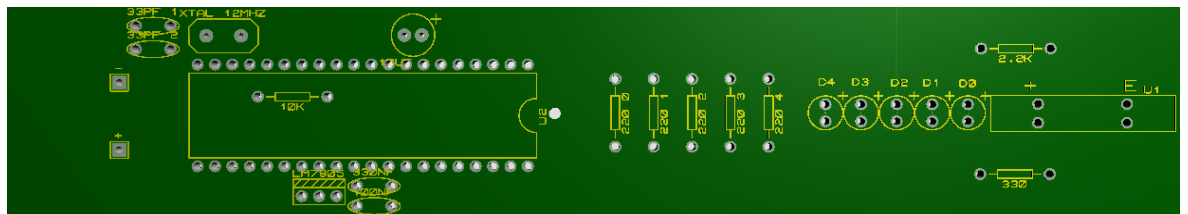
- Những đường màu xanh lá cây có mũi tên là thước đo centimeter.

- Đường vàng là rìa bảng mạch.

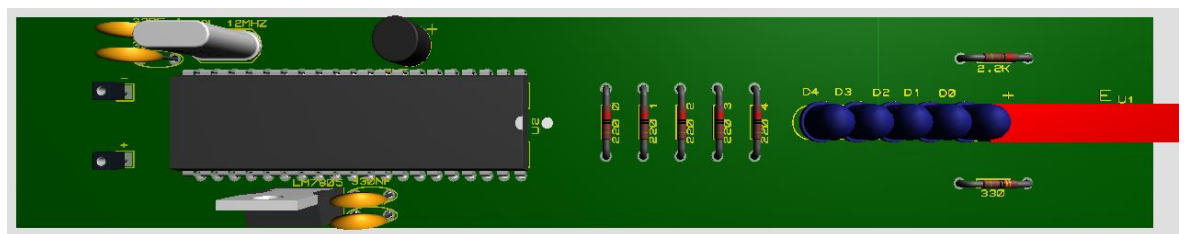
- Bảng mạch dài 17.15cm, rộng 3.15cm.

- Hình 3D mô phỏng: (chiếc lỗ ở ngay chính giữa bảng mạch là mình ước lượng và đánh dấu trọng tâm của bảng mạch)

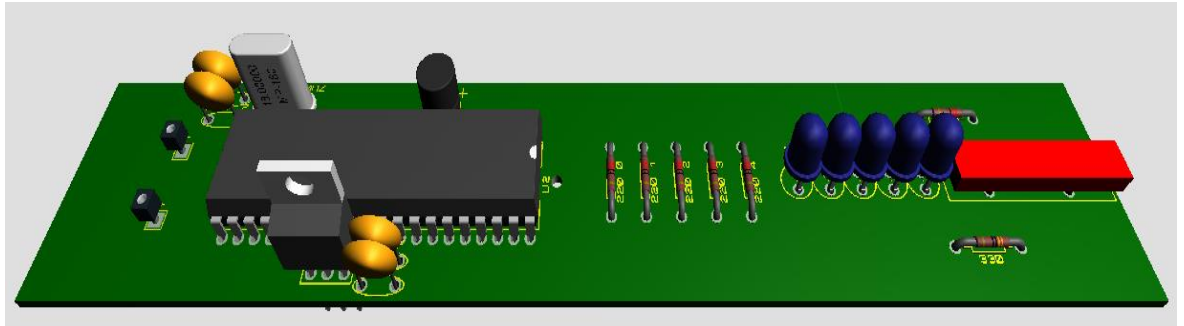
+ Mặt trên khi chưa cắm linh kiện.



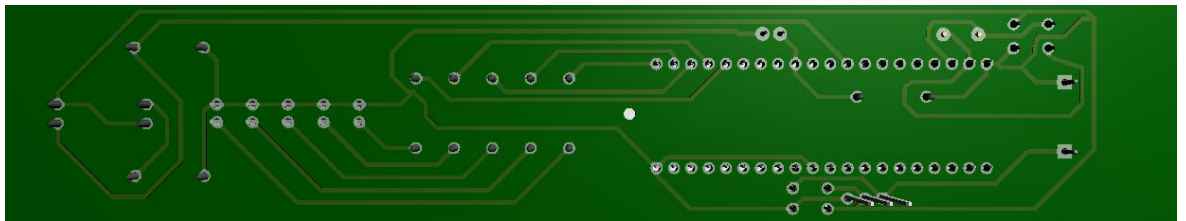
+ Mặt trên đã cắm linh kiện.



+ Góc nhìn khác.



+ Lật mạch theo chiều ngang.



- Nhược điểm lớn là mỗi lần thay đổi motor hoặc thay đổi bán kính đường tròn quay, ta phải tính lại chi tiết các thông số và điều chỉnh hàm tạo trễ, rồi sửa đổi kích thước mảng dữ liệu (nếu ai dùng giải thuật phần mềm hay hơn thì bỏ qua bước này):

6000 vòng : 60 giây

1 vòng : ? giây

Suy ra

1 vòng : $60 / 6000 = 0.01 \text{ (s)} = 10 \text{ (ms)}$

- Bán kính của vòng tròn quay (radius) : **r** = 5 (cm)

- Chu vi của vòng tròn quay (Perimeter) : **P** = $2\pi r = 2 \times 3,14 \times 5 = 31.4 \text{ (cm)}$

- Độ rộng của một đèn LED : **w** = 3 (mm) = 0.3 (cm)

- Độ rộng của một cột LED : **W** = $1 \times w = 1 \times 0.3 = 0.3 \text{ (cm)}$

- Số lượng cột LED có thể có : **x** = **P** / **W** = $31.4 / 0.3 = 104.67 = 103 \text{ (cột)}$

- Thời gian hiển thị của 103 cột : **T** = 10 (ms)

- Thời gian hiển thị của 1 cột : $t = T / x = 10 / 103 = 0.09708 \text{ (ms)}$
= 97 (μs)
- Thời gian hiển thị 5 cột LED liên tiếp : $t_{5 \text{ cột}} = 5 \times t = 5 \times 97 = 485 \text{ (}\mu\text{s)}$
- Độ rộng của 5 cột LED liên tiếp : $W_{5 \text{ cột}} = 5 \times W = 5 \times 0.3 = 1.5 \text{ (cm)}$
- Số lượng ký tự có thể hiển thị : $z = P / W_{5 \text{ cột}} = 31.4 / 1.5 = 20.93 =$
20 (ký tự)

- Tiến hành chế biến phần cứng. Đầu tiên ta phải thiết kế ý tưởng trước, đừng vội tìm mua các linh kiện ngay.

- Lắp ghép motor:

+ Nếu motor của bạn có trục ngắn thì cần tìm mua 1 khớp nối trục, cây khóa lục giác để vặn ốc cố định khớp, trục sắt để nối dài trục motor. Ở dưới là hình của khớp nối dài khoảng 2cm (viên sắt màu vàng), 4 ốc vặn cố định trục (4 ốc màu đen), cây lục giác (cầm trên tay sẽ hiểu). Đừng quên mua một trục nối dài. Khi mua, ta phải kiểm tra chất lượng gia công của khớp nối, trục nối dài ngay tại cửa hàng, xem có bị méo mó ở đâu không, một bộ như vậy mình đã mua với giá khoảng 25.000đ. (chiếc khớp đầu tiên mình mua có chất lượng kém, khi nối vào rồi cho motor quay thì trục bị rung mạnh)



- Bảng mạch:

- + In mạch thủ công và cố gắng đạt thẩm mỹ cao nhất có thể. Càng chính xác về kích thước bảng mạch thì càng tốt. (ai muốn đạt chất lượng tốt nhất thì đặt mạch cũng được). Nên dùng để cắm 40 chân hàn vào bảng mạch thay vì hàn trực tiếp chip vi điều khiển, mục đích là tránh trường hợp chip nóng quá mức dẫn đến chết chip trong khi hàn. (và tháo chip ra để nạp Code mới)
- + Dùng khoan lỗ trọng tâm vội, ta phải xác định trọng tâm lại một lần nữa. Mẹo nhỏ đơn giản để xác định trọng tâm của một vật thì mọi người tự tìm hiểu. (bước này nếu ai tìm được cách đặt bảng mạch lên trục motor hay hơn thì không cần khoan lỗ trọng tâm trên bảng mạch)
- + Đảm bảo những mối hàn phải chắc chắn, kiểm tra kỹ lưỡng hoạt động của cảm biến.

- Ý tưởng cấp nguồn cho bảng mạch sử dụng bạc đạn (đây là phần mình gặp khó khăn nhất vì đã không nghĩ ra cách dùng bạc đạn):

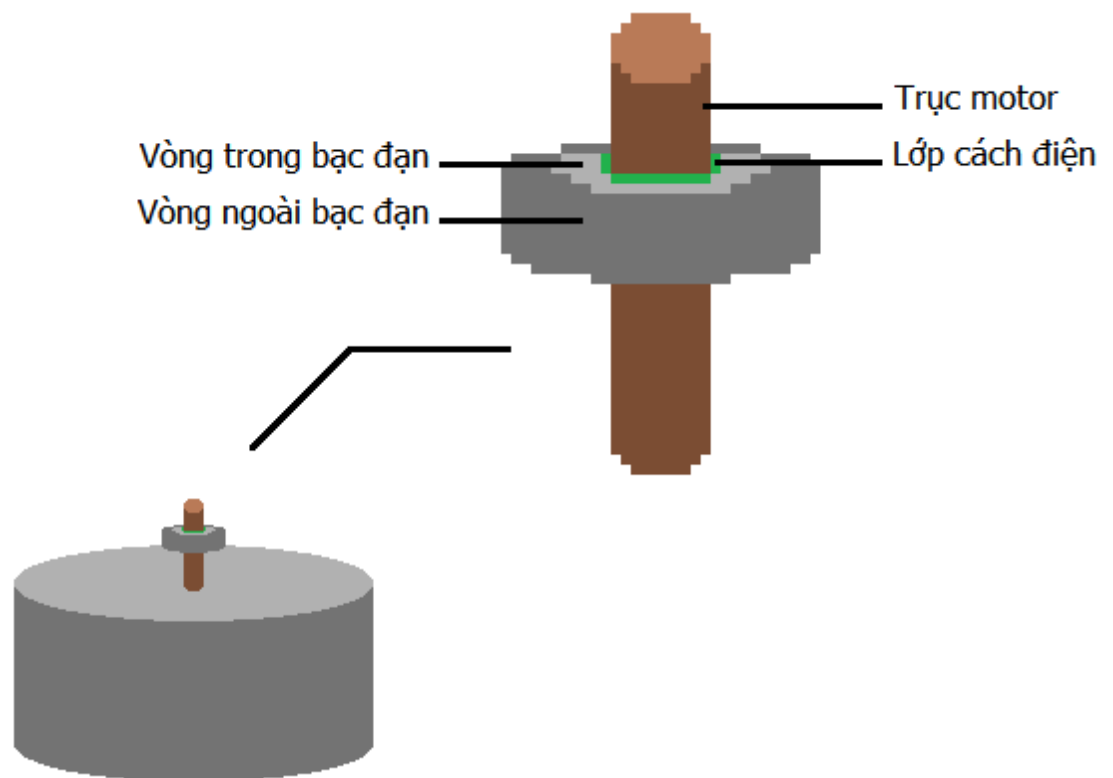
- + Tìm hiểu nguyên lý hoạt động và mua bạc đạn (ổ bi, tiếng Anh là Ball Bearing), nếu ai ở Tp.HCM thì có thể tìm mua ở cửa hàng Nam Bạc Đạn trên đường Tạ Uyên, địa chỉ 54 Tạ Uyên, Phường 15, Quận 5, TP. Hồ Chí Minh (có bán cả khớp nối, trục, nhiều thứ bổ trợ cho motor). Những linh kiện như bạc đạn là những thứ đồ chuyên dụng, sẽ khó tìm mua cho những ai mới tiếp cận. Bạc đạn là cốt lõi của phương pháp cấp nguồn cho bảng mạch mà không cần dùng PIN riêng. Thông thường những nguồn trên mạng, họ dùng trực tiếp nguồn PIN cấp riêng cho mạch và đặt ngay lên bảng mạch, ưu điểm là dễ thiết kế, dễ hiểu, nhược điểm là kích cỡ bảng mạch cũng như thiết bị LED quay khi hoàn thành sẽ lớn, chưa kể là độ nặng có thể làm giảm tốc độ của motor.
- + Hình ảnh bạc đạn:



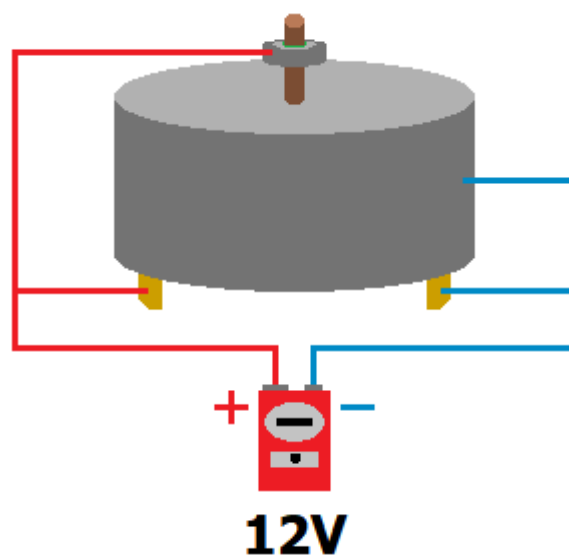
+ Bạn phải cầm trên tay một chiếc bạc đạn, sau đó dùng đồng hồ đo vòng ngoài và vòng trong của bạc đạn xem chúng có nối với nhau hay không, nhớ phải ép mạnh hai đầu kim vào các vòng của bạc đạn (chạm nhẹ thì đồng hồ không reo, sẽ bị làm tưởng vòng ngoài không dẫn điện vào vòng trong), nếu đồng hồ reo tức là điện cấp cho vòng ngoài sẽ được truyền vào vòng trong.

+ Mình mô phỏng ý tưởng cấp nguồn như sau:

* Lồng một lớp cách điện quanh trục motor, lớp cách điện có thể là ống nhựa/cao su, băng keo đen, Phải kiểm tra và chắc chắn rằng vòng trong bạc đạn hoàn toàn cách điện với trục motor, vì mục đích chính là không để cực dương nguồn tiếp xúc trực tiếp cực âm nguồn (đọc tiếp sẽ rõ cách đi dây). Tiếp đến gắn bạc đạn vào trục motor (có thể chọn loại bạc đạn to hơn và gắn vào khớp nối vẫn được, nhớ cách điện khớp nối với vòng trong bạc đạn). Hình vẽ:



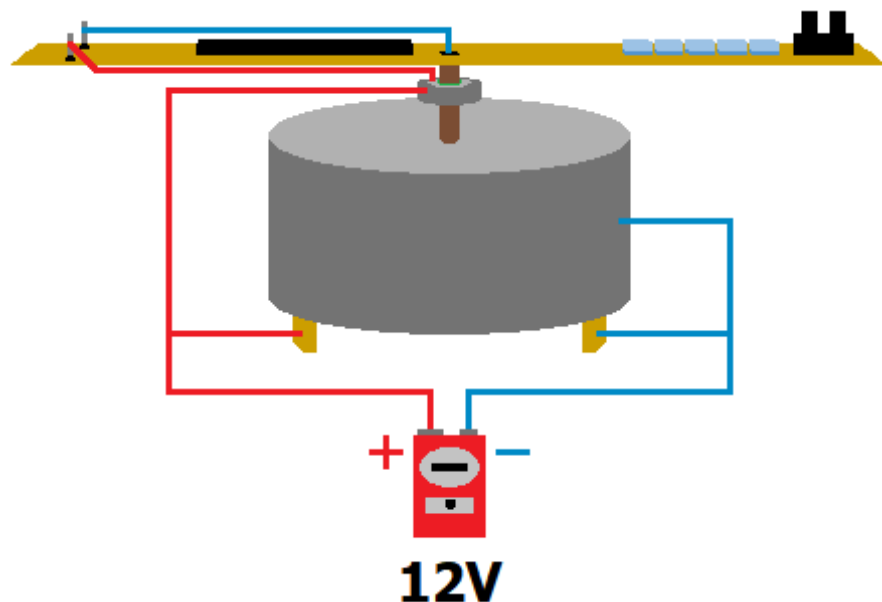
* Cực dương nguồn nối với vòng ngoài bạc đạn và một chân của motor (nối làm sao để motor quay theo chiều mong muốn), cực âm nguồn nối với vỏ motor và một chân khác của motor (vỏ sẽ dẫn điện vào trục motor, kiểm tra bằng đồng hồ sẽ thấy). Hình vẽ:



* Cực dương của bảng mạch nối với vòng trong bạc đạn, cực âm của bảng mạch nối với trục motor. Sau đó lắp bảng mạch vào. Hai bên cánh phải cân bằng trọng lượng để thiết bị không bị lắc khi quay.

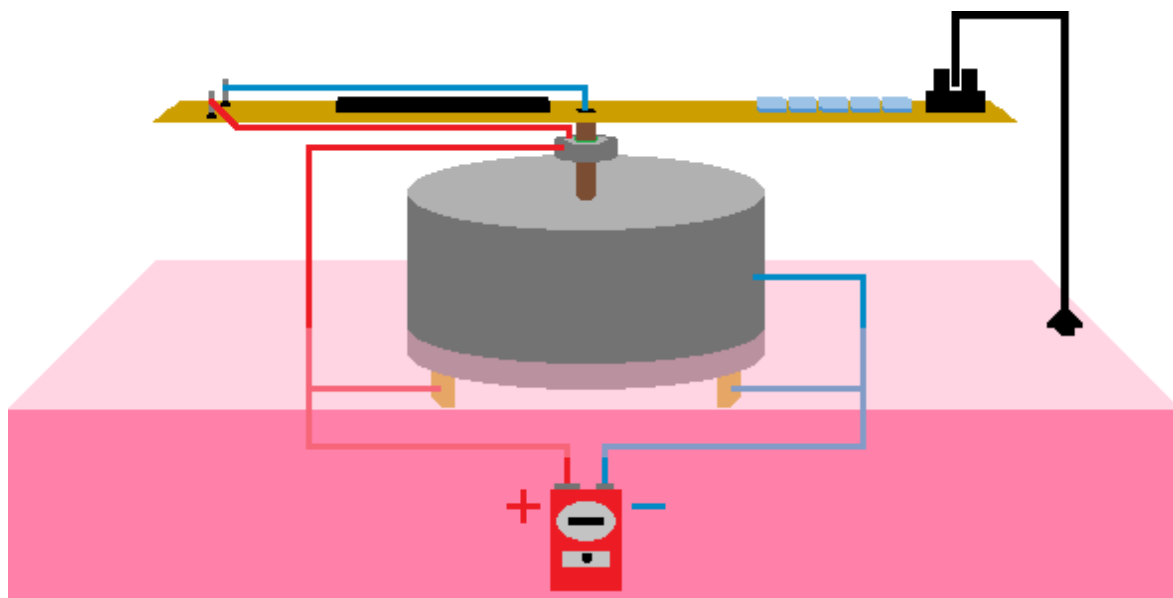
Nguyên lý: khi trục motor quay, vòng trong bạc đạn quay theo, vòng ngoài đứng yên, điện cấp cho vòng ngoài được truyền vào vòng trong rồi truyền lên bảng mạch, vì vậy sẽ không phải lo dây dẫn bị xoắn lại khi quay. Cuối cùng ta đã hoàn thành việc cấp nguồn cho hệ thống.

Lưu ý là phải tính toán kỹ cường độ dòng điện cho từng nhánh, vì dòng sẽ bị chia nhánh khi ta mắc song song nhiều thiết bị. Hình vẽ: (bảng mạch màu vàng là do không phủ xanh)



- Ý tưởng để đỡ cho motor và kèm theo một vật dùng để ngắt cảm biến:

+ Bước này đơn giản, tùy khả năng chế biến và sáng tạo của mỗi người, vật dùng để ngắt cảm biến có thể là bất cứ thứ gì không cho tia hồng ngoại xuyên qua. Sẵn mình vẽ mô phỏng luôn:



+ Nối thêm một công tắc để bật tắt thiết bị là hoàn tất.

- Sau khi đã mô phỏng ý tưởng, **mình tìm mua và đo đạc chuẩn xác những thứ cần mua, nếu không đo đạc kỹ các thông số như vòng trong, vòng ngoài bạc đạn, độ dày của lớp cách điện, độ dày của dây dẫn,... sẽ khiến bạn mất thời gian - chi phí mua lại.**

- Đến đây ta phải cẩn thận việc nối dây, cách điện, đặc biệt chú ý sự dẫn điện từ ngoài vào trong của bạc đạn, nếu hỏng một điểm nhỏ có thể khiến bạn làm lại từ đầu toàn bộ quá trình.

- Cách lắp ráp của mình có hơi khác mô phỏng ở trên, cụ thể là lúc dùng bạc đạn thì mình lồng với khớp nối chứ không lồng trực tiếp với trục 2mm (lý do là vì không tìm được kích cỡ phù hợp cho ống cách điện)

- Dưới đây là các linh kiện kèm thông số mình sử dụng:

+ Motor 6000RPM 12V, trục ngắn 2mm.



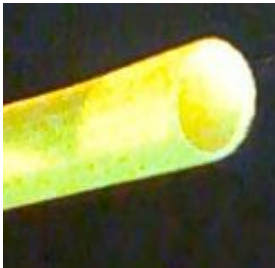
+ Khớp nối trục 2mm – 2mm (tức 2mm nối với 2mm), bán kính vòng ngoài 9mm.



+ Trục 2mm, dài khoảng 2cm.



+ Ống nhựa đường kính trong 9mm, dày khoảng 0.5mm, dùng làm lớp cách điện giữa khớp nối và bạc đạn. (nếu ai tìm được ống silicon 2mm thì hãy nên dùng nó, một phần vì thẩm mỹ, một phần vì sự chắc chắn)



+ Dây đồng hút chì, rộng 3mm, dày khoảng 0.3mm, dùng làm dây dẫn nối giữa vòng trong bạc đạn và cực dương bảng mạch.



- + Bạc đạn đường kính vòng trong 10mm. không nhất thiết phải quan tâm đường kính vòng ngoài.
- + Vật dùng để ngắt mình chọn một mảnh bìa cứng.
- + Bộ công nghệ lớp 5 hồi tiểu học và một ít thanh sắt hình chữ L, dùng làm đế đỡ motor và trục đỡ vật ngắt.

Chương 4. KẾT QUẢ THỰC NGHIỆM

-

Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

- Quá trình thực hành đồ án bắt đầu từ ngày 15/10/2017. Hoàn tất ngày .
- Code chỉnh sửa nhiều lần giá trị hàm tạo trễ, vì khi lắp ghép phần cứng gặp không ít trục trặc. Cụ thể là trục trặc ở mạch ngắt cảm biến, cách lựa chọn motor, giải thuật phần mềm, kích thước bảng mạch, xác định trọng tâm của bản mạch, đặc biệt hao tốn nhiều thời gian cho việc tìm tòi và chạy đi mua những linh kiện chuyên dụng cần thiết cho motor, mất nhiều thời gian cho các công đoạn tự làm đi làm lại mạch in thủ công.
- Qua đồ án này, mình có thêm kinh nghiệm sử dụng các vật dụng chuyên dụng cho motor, cách cấp nguồn độc đáo sử dụng bạc đạn, hiểu được một trong nhiều ứng dụng của vi điều khiển, hiểu sâu hơn về PIN.

5.2. Hướng phát triển

- Ta có thể sử dụng các dòng chip vi điều khiển khác như AVR, PIC, ATmega, ARM để lập trình, sử dụng các linh kiện dán cho bảng mạch gọn nhẹ.
- Thiết kế LED quay điều khiển hình ảnh hiển thị từ xa.
- Dưới đây là Video công nghệ LED quay hiển thị hình ảnh 3D:

https://www.youtube.com/watch?v=Ey-eO-H2_Mk

TÀI LIỆU THAM KHẢO

Theo chuẩn IEEE