

ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC NÔNG LÂM HUẾ

ĐỀ CƯƠNG
KHÓA LUẬN TỐT NGHIỆP

TÊN ĐỀ TÀI:

Họ và tên sinh viên : Trần Nguyên Danh

Ngành: Kỹ thuật cơ – điện tử

Giáo viên hướng dẫn: Ts. Nguyễn Tiến Long

Thời gian thực hiện: từ tháng 10/2019

Địa điểm thực tập:

HUẾ - 2019

ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC NÔNG LÂM HUẾ

ĐỀ CƯƠNG
KHÓA LUẬN TỐT NGHIỆP

TÊN ĐỀ TÀI:

Họ và tên sinh viên : Trần Nguyên Danh

Ngành: Kỹ thuật cơ – điện tử

Giáo viên hướng dẫn: Ts. Nguyễn Tiến Long

Thời gian thực hiện: từ tháng 10/2019

Địa điểm thực tập:

HUẾ - 2019

LỜI NÓI ĐẦU

Ngày nay, nền kinh tế của nước ta đang trên đà phát triển mạnh mẽ, các ngành công nghiệp ngày càng đòi hỏi trình độ khoa học kỹ thuật cao. Nhu cầu của nhà sản xuất cũng như nhà tiêu dùng là tăng không ngừng. Đây là cơ hội và cũng là thách thức cho ngành cơ điện tử, với việc ứng dụng các thành tựu nhân loại để phục vụ nhu cầu xã hội.

Các loại máy móc đã và đang trở thành công cụ lao động thông minh, từng bước thay thế con người trong hoạt động sản xuất. Nhờ đó mà năng suất và chất lượng lao động ngày càng được cải thiện và tiệm cận sự hoàn hảo. Trong học phần này, chúng em thực hiện đề tài: *Thiết kế và chế tạo Hệ thống đèn giao thông tự động*. Với mục tiêu tạo ra một sản phẩm có khả năng ứng dụng thực tế mang lại hiệu quả kinh tế cao. Các nhiệm vụ cần thực hiện trong đề tài này bao gồm:

- Xác định phương án thiết kế.
- Nghiên cứu và điều khiển các đèn thành phần.
- Chạy thử nghiệm và xử lý lỗi phát sinh.

Để đề án này đạt kết quả tốt đẹp, chúng tôi đã nhận được sự hỗ trợ, giúp đỡ của nhiều cơ quan, cá nhân. Với tình cảm sâu sắc cho phép chúng tôi được bày tỏ lòng biết ơn sâu sắc đến tất cả các cá nhân và cơ quan đã tạo điều kiện giúp đỡ trong suốt quá trình học tập và nghiên cứu đề tài.

Trước hết chúng tôi gửi tới các thầy Khoa Cơ khí trường Đại học Nông Lâm Huế lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, chỉ dạy và truyền đạt kiến thức để chúng tôi hoàn thành đề tài này.

Đặc biệt chúng tôi gửi lời cảm ơn chân thành nhất tới thầy giáo – *TS. Nguyễn Tiến Long* đã quan tâm giúp đỡ, hướng dẫn hoàn thành tốt đề tài này trong thời gian qua. Không thể không nhắc tới sự hỗ trợ truyền đạt kinh nghiệm của các bạn trong tập thể lớp **CDT-49**.

Tuy nhiên do kiến thức và kinh nghiệm còn nhiều hạn chế nên khó tránh khỏi những thiếu sót. Chúng tôi rất mong nhận được sự thông cảm và đóng góp ý kiến của các thầy và các bạn sinh viên để đề án của chúng tôi được hoàn chỉnh hơn. Xin chân thành cảm ơn.

Huế, ngày 12 tháng 12 năm 2019

Người thực hiện đề tài:

Trần Nguyên Danh

MỤC LỤC

{Đề 2 dòng trống tại đây}

| | |
|--|-----------|
| Tóm tắt | |
| Nhiệm vụ đồ án | |
| Lời nói đầu và cảm ơn | i |
| Lời cam đoan liên chính học thuật | ii |
| Mục lục | iii |
| Danh sách các bảng biểu, hình vẽ và sơ đồ | v |
| Danh sách các cụm từ viết tắt | vii |
| CHƯƠNG 1. ĐẶT VẤN ĐỀ | 7 |
| 1.1. Tính cấp thiết của đề tài: | 7 |
| 1.2. Mục tiêu của đề tài: | 7 |
| 1.3. Nội dung báo cáo: | 8 |
| CHƯƠNG 2: TỔNG QUAN VỀ HỆ THỐNG ĐÈN GIAO THÔNG | 9 |
| 2.1. Yêu cầu của hệ thống | 9 |
| 2.2. Lựa chọn phương án thiết kế | 9 |
| 2.3. Tổng quan về hệ thống | 9 |
| CHƯƠNG 3: THIẾT KẾ, CHẾ TẠO HỆ THỐNG ĐÈN GIAO THÔNG | 11 |
| 3.1. Các thành phần cơ bản..... | 11 |
| 3.1.1. Arduino Mega..... | 11 |
| 3.1.2. IC ghi dịch 74hc595..... | 12 |
| 3.1.3. Led 7 thanh..... | 14 |
| 3.1.4. Module Nguồn LM2596 | 17 |
| 3.2. Sơ đồ mạch điện:..... | 18 |

| | |
|-------------------------------------|----|
| 3.3. Chương trình điều khiển: | 20 |
| 3.3.1. Khởi khai báo | 20 |
| 3.3.2. Khởi cài đặt | 22 |
| 3.3.3. Các chương trình con | 23 |
| 3.3.4. Chương trình chính..... | 31 |
| 3.4. Kết quả thử nghiệm..... | 31 |

CHƯƠNG 4: ĐÁNH GIÁ ĐỀ TÀI VÀ HƯỚNG PHÁT TRIỂN **33**

| | |
|------------------------------------|----|
| 4.1. Kết quả đạt được..... | 33 |
| 4.2. Hạn chế còn tồn tại | 33 |
| 4.3. Hướng phát triển đề tài | 33 |

| | |
|---------------------------------|-----------|
| Tài liệu tham khảo | 33 |
|---------------------------------|-----------|

| | |
|----------------------|-----------|
| Phụ lục | 33 |
|----------------------|-----------|

CHƯƠNG 1. ĐẶT VẤN ĐỀ

1.1. Tính cấp thiết của đề tài:

Trong hệ thống giao thông hiện nay ở nước ta, vấn đề về an toàn giao thông và tránh ùn tắc tại các đô thị và thành phố lớn là một trong những vấn đề hết sức cấp bách và được toàn xã hội quan tâm. Vì vậy các phương tiện hướng dẫn giao thông đóng vai trò rất quan trọng, nó góp phần hạn chế những xung đột xảy ra khi tham gia giao thông. Tại các đô thị thì hệ thống đèn điều khiển giao thông là rất quan trọng. Hệ thống đèn điều khiển giao thông không những có tác dụng hạn chế những xung đột trong giao thông thành phố mà còn là công cụ điều khiển các luồng giao thông nhằm hạn chế ùn tắc - một vấn đề nan giải tại các thành phố lớn.

Hiện nay với sự phát triển nhanh chóng của nền kinh tế nước nhà và lượng dân cư đông đúc. Nhu cầu đi lại, vận chuyển cũng tăng lên nhanh chóng đi kèm theo nó là bài toán tắc nghẽn ùn tắc giao thông. Ùn tắc giao thông gây thiệt hại không nhỏ cho sự phát triển kinh tế quốc gia, giảm hiệu suất lao động và tăng các chi phí không cần thiết trong quá trình sản xuất. Trong bối cảnh kinh tế giảm phát và khó khăn như hiện nay, lãng phí trong giao thông lại đặt thêm một gánh nặng đối với đời sống kinh tế của người dân. Nguyên nhân của vấn đề này một phần là do cơ sở hạ tầng chưa đáp ứng được nhu cầu lưu thông hiện nay, một phần là do việc phân chia, định thời gian của hệ thống đèn giao thông hiện tại ở các nút giao thông chưa hợp lý khiến cho số lượng phương tiện giao thông bị ùn tắc theo thời gian rồi dẫn đến tắc nghẽn giao thông. Hiện tại có rất nhiều giải pháp để nâng cao hiệu quả phân chia thời gian các làn đường ở các nút giao thông trọng điểm. Như lập trình bộ điều khiển tín hiệu theo khung giờ cao điểm, hay trực tiếp phân luồng bởi con người. Tuy vậy các giải pháp chỉ mang tính tương đối và tốn tài nguyên về con người.

Vì lý do trên, chúng em quyết định chọn đề tài “Thiết kế hệ thống đèn giao thông tại ngã tư có giao với đường sắt” Nhằm góp phần vào việc xây dựng một phương pháp điều khiển đèn tín hiệu giao thông một cách hợp lý hơn, giúp tình trạng giao thông tại các ngã tư ở nước ta được ổn định hơn.

1.2. Mục tiêu của đề tài:

Mật độ giao thông phát triển mạnh mẽ ở các đô thị lớn đòi hỏi hệ thống điều khiển giao thông phải chính xác, linh hoạt, đơn giản nhưng hiệu quả cao. Hệ thống đèn giao thông hai pha phân nào đã đáp ứng được những yêu cầu trên.

Mỗi pha gồm các đèn: Xanh - Đỏ - Vàng điều khiển các phương tiện cơ giới.

- Hoạt động của từng pha ở chế độ điều khiển tự động như sau:
- Đèn xanh: trong 20 giây, hướng đi ứng với pha này được phép đi.
- Đèn vàng: trong 3 giây, thông báo cho các phương tiện ứng với pha này giảm tốc độ, chuẩn bị dừng lại.
- Đèn đỏ: trong 27 giây, các phương tiện dừng lại.
- Khi đèn xanh của làn đường 1 sáng thì đèn xanh của người đi bộ ở làn đường kia cũng sang và ngược lại.
- Khi có tín hiệu báo tàu hỏa sắp đến, đèn đỏ sẽ được bật ở hai chiều vuông góc với tuyến đường sắt để đảm bảo an toàn cho tuyến đường sắt.

1.3. Nội dung báo cáo:

Đồ án tốt nghiệp “ Thiết kế và chế tạo Hệ thống đèn giao thông tại ngã tư có giao với đường sắt” được chúng em trình bày trong 4 chương với bố cục như sau:

- Chương 1: Đặt vấn đề
Chương này nêu ra những vấn đề mà xã hội đang gặp phải, lý do chọn đề tài, mục tiêu và bố cục đề tài.
- Chương 2: Tổng quan về Hệ thống đèn giao thông.
Chương này giới thiệu tổng quan về hệ thống đèn giao thông, các giải pháp được đưa ra để giải quyết vấn đề và lựa chọn phương án tối ưu nhất.
- Chương 3: Thiết kế, chế tạo Hệ thống đèn giao thông.
Chương này giới thiệu về các thành phần của hệ thống cũng như tính toán các linh kiện cho phù hợp. Xây dựng chương trình điều khiển hoàn chỉnh, chạy thử nghiệm hệ thống và kiểm tra lỗi.
- Chương 4: Kết quả đạt được và hướng phát triển
Tóm tắt những kết quả đạt được , những hạn chế và nêu lên các hướng phát triển trong tương lai.

CHƯƠNG 2: TỔNG QUAN VỀ HỆ THỐNG ĐÈN GIAO THÔNG

2.1. Yêu cầu của hệ thống

Hệ thống được thiết kế để đặt tại ngã tư có giao với đường sắt, hệ thống gồm có hai pha và có những yêu cầu như sau:

Mỗi pha gồm các đèn: Xanh - Đỏ - Vàng điều khiển các phương tiện cơ giới.

- Hoạt động của từng pha ở chế độ điều khiển tự động như sau:
- Đèn xanh: trong 20 giây, hướng đi ứng với pha này được phép đi.
- Đèn vàng: trong 3 giây, thông báo cho các phương tiện ứng với pha này giảm tốc độ, chuẩn bị dừng lại.
- Đèn đỏ: trong 27 giây, các phương tiện dừng lại.
- Khi đèn xanh của làn đường 1 sáng thì đèn xanh của người đi bộ ở làn đường kia cũng sang và ngược lại.
- Khi có tín hiệu báo tàu hỏa sắp đến, đèn đỏ sẽ được bật ở hai chiều vuông góc với tuyến đường sắt để đảm bảo an toàn cho tuyến đường sắt.

2.2. Lựa chọn phương án thiết kế

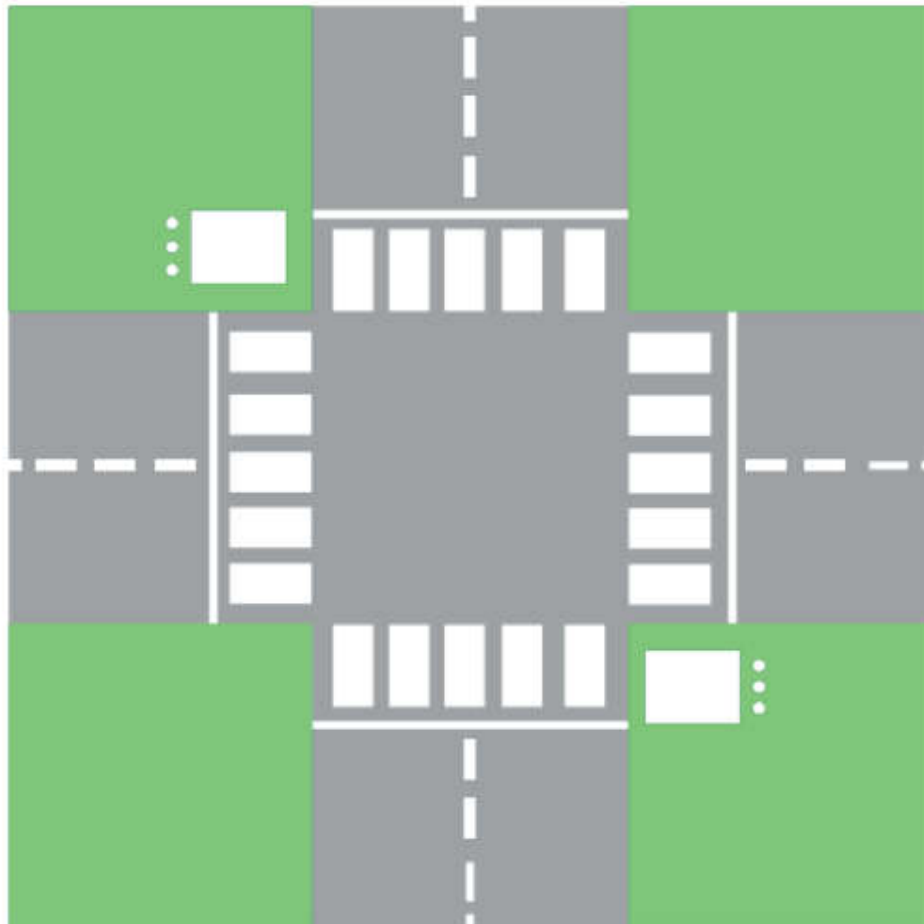
Hiện nay việc sử dụng các mạch số kết hợp với chip vi điều khiển trong các hệ thống điều khiển tự động đã trở nên rất phổ biến vì những ưu việt của nó như: độ chính xác, khả năng lập trình được, tốc độ điều khiển nhanh, sử dụng đơn giản,... Mặt khác kỹ thuật số, vi xử lý, vi điều khiển là lĩnh vực đang phát triển mạnh mẽ và có ứng dụng trong rất nhiều ngành sản xuất. Vì vậy, ta sẽ thiết kế một hệ thống điều khiển giao thông đơn giản, chỉ sử dụng bộ vi điều khiển. Tất cả các tín hiệu điều khiển đều được đưa đến khối hiển thị trực tiếp từ các cổng của bộ vi điều khiển thông qua một IC ghi-dịch 74HC595.

Phương án này có đặc điểm là mạch gọn nhẹ, không quá phức tạp, cách thức bố trí linh kiện dễ dàng, lập trình đơn giản, dễ chỉnh sửa.

2.3. Tổng quan về hệ thống

Hệ thống sử dụng một vi điều khiển chính là Arduino Nano, điều khiển các IC ghi-dịch và các bóng Led của Led 7 thanh để hiển thị thời gian còn lại của mỗi pha, đồng thời điều khiển đèn tín hiệu của 4 trụ đèn nhờ hệ thống rơ le DC.

Ngoài ra hệ thống có sử dụng một bộ điều khiển dành cho người gác đường tàu, có thể bật chế độ có tàu hỏa để vi điều khiển trung tâm điều chỉnh chế độ phù hợp với giao thông, đồng thời hạ barrie và bật đèn cảnh báo có tàu hỏa.



CHƯƠNG 3: THIẾT KẾ, CHẾ TẠO HỆ THỐNG ĐÈN GIAO THÔNG

3.1. Các thành phần cơ bản

3.1.1. Arduino Mega

Arduino là một board mạch có mã nguồn mở, được sinh ra tại thị trấn Ivrea ở Ý vào năm 2005. Cái tên arduino được đặt theo tên một quán bar tại Ivrea, nơi mà các nhà sáng lập thường xuyên gặp mặt. Bản thân cái tên của quán bar này cũng đc đặt theo tên của Bá tước vùng Ivrea và là vua của Italy từ 1002 – 1014.

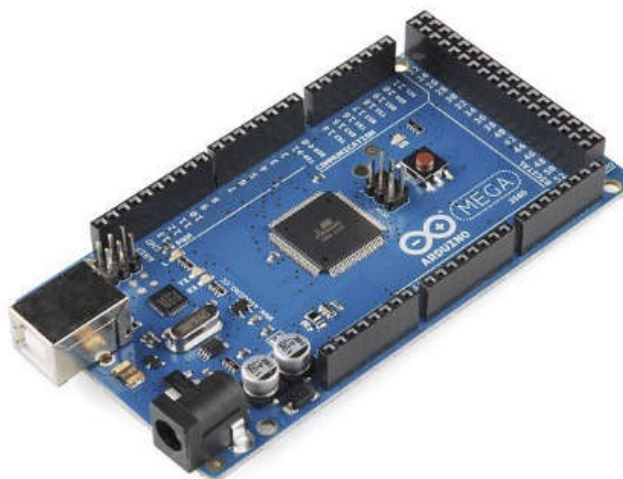
Cùng với board mạch thì môi trường phát triển thích hợp (IDE) cũng được ngày càng nâng cấp, chạy được trên các máy tính các nhân thông thường và cho phép người dùng có thể viết các chương trình cho arduino bằng ngôn ngữ C và C++ ngoài ra cũng có thể nhúng thêm ASM hay HTML trong một vài ứng dụng đặc biệt

Arduino có nhiều loại như Uno R3, Promini, Nano,... trong đồ án này chúng em sử dụng arduino Mega2560 R3.

Arduino Mega2560 sử dụng vi điều khiển Atmega2560 thuộc họ AVR của hãng Atmel có các thông số cơ bản như sau:

| STT | Đặc điểm | Thông số |
|-----|-------------------------------|---|
| 1 | Vi xử lý | ATmega2560 |
| 2 | IC nạp và giao tiếp UART | ATmega16U2 |
| 3 | Điện áp hoạt động | 5V |
| 4 | Điện áp đầu vào | 7-12V |
| 5 | Dòng tối đa trên mỗi chân I/O | 20mA |
| 6 | Chân vào/ra số | 54 (15 chân có khả năng xuất PWM) |
| 7 | Chân vào tương tự | 16 chân (10 bit ADC) |
| 8 | Dòng điện mỗi chân vào/ra | 40 mA |
| 9 | Bộ nhớ Flash | 256 KB trong đó 8 KB sử dụng cho bootloader |
| 10 | SRAM | 8KB |
| 11 | EEPROM | 4 KB |

| | | |
|----|------------|------------------|
| 12 | UART | 4 |
| 13 | Timer | 6 |
| 14 | Kích thước | 101.52 x 53.3 mm |



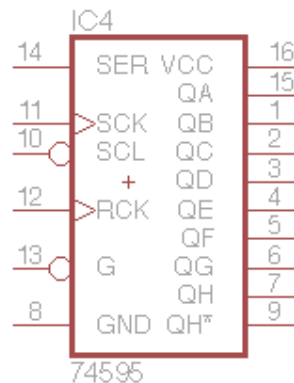
Arduino Mega2560 R3

3.1.2. IC ghi dịch 74hc595

74HC595 thuộc dòng 74xx595 (xx là ký hiệu có thể khác nhau giữa các hãng) là IC ghi dịch 8bit chốt dữ liệu, đầu vào nối tiếp, đầu ra song song. Dùng để mở rộng cổng ra của MCU (Vi điều khiển) sử dụng giao tiếp SPI. Chúng ta có thể mắc nối tiếp các IC 595 với nhau để mở rộng tùy thích



IC ghi-dịch 74HC595



Sơ đồ chân của IC 74HC595

Giao tiếp SPI với IC 595 gồm 3 đường:

DATA: (14) – Dữ liệu từ MCU sẽ đưa vào chân này, dữ liệu được đưa vào theo kiểu nối tiếp. Để dễ hình dung, bạn có thể hiểu giống như bạn đưa các đồng xu vào ống vậy. Tức dữ liệu sẽ được đẩy dần vào các ngăn nhớ của 595.

CLK: (11) – Đây là tín hiệu đồng bộ quá trình giao tiếp giữa MCU và IC 595. Cứ mỗi bit dữ liệu được gửi đi đồng nghĩa với việc gửi đi 1 xung clock. Tức khi có 1 xung tích cực ở sườn dương (từ 0 lên 1) thì 1 bit sẽ được dịch vào IC.

SCK: (12) – Là chân Shift Clock, dùng để chốt dữ liệu. Khi có 1 xung tích cực ở sườn dương thì dữ liệu ở trong IC 595 sẽ được đẩy ra các chân Output tương ứng, và bạn có thể đẩy dữ liệu ra bất cứ lúc nào.

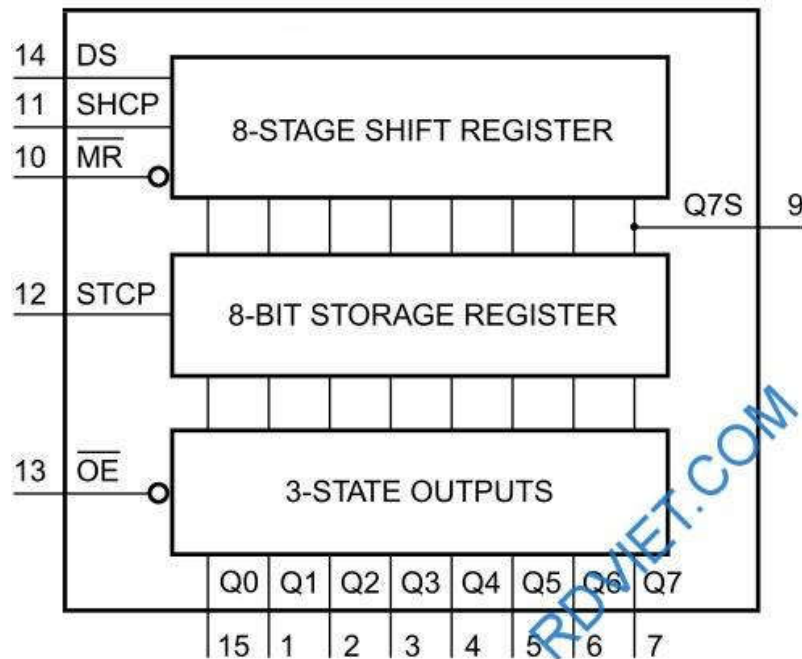
VD: Bạn dịch 3bits vào IC, khi có tín hiệu chốt, dữ liệu sẽ được đưa ra ở Qa – Qb – Qc. Chiều dịch dữ liệu là từ Qa đến Qh. Vì vậy nên cần chú ý chiều dịch dữ liệu.

Một số chân chức năng khác trên IC 595:

OE (13): Output Enable – Dữ liệu được phép xuất ra khi chân này ở mức tích cực thấp.

MR (10): Memory Clear – Khi chân này ở mức tích cực thấp sẽ xóa dữ liệu đã đưa vào IC.

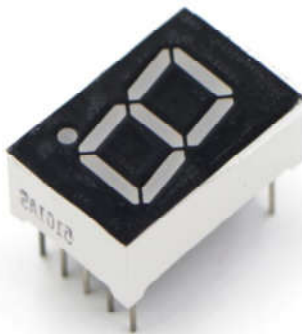
Q7' (9): Chân dữ liệu nối tiếp, chân này sẽ nối với chân DATA của IC 595 tiếp theo trong trường hợp mắc nối tiếp nhiều IC. Dữ liệu sẽ được dịch qua chân này khi đã dịch đủ 8bits cho IC trước đó.



Sơ đồ khối của IC 74HC595

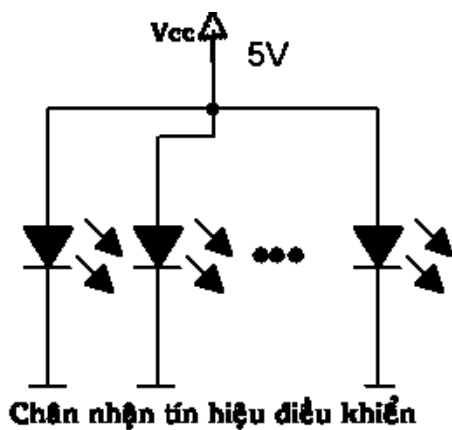
3.1.3. Led 7 thanh

Trong các thiết bị, để báo trạng thái hoạt động của thiết bị đó cho người sử dụng với thông số chỉ là các dãy số đơn thuần, thường người ta sử dụng "led 7 đoạn". Led 7 đoạn được sử dụng khi các dãy số không đòi hỏi quá phức tạp, chỉ cần hiển thị số là đủ, chẳng hạn led 7 đoạn được dùng để hiển thị nhiệt độ phòng, trong các đồng hồ treo tường bằng điện tử, hiển thị số lượng sản phẩm được kiểm tra sau một công đoạn nào đó...

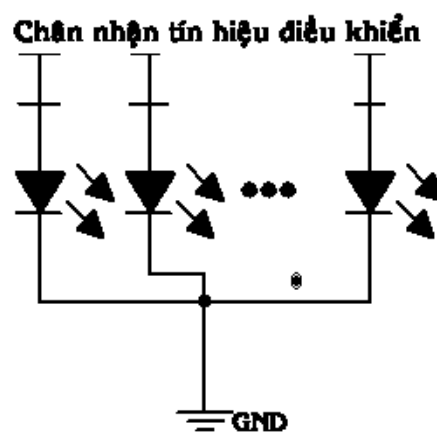


Led 7 đoạn có cấu tạo bao gồm 7 led đơn có dạng thanh xếp theo hình dạng số 8 và có thêm một led đơn hình tròn nhỏ thể hiện dấu chấm tròn ở góc dưới, bên phải của led 7 đoạn 8 led đơn trên led 7 đoạn có Anode(cực +) hoặc Cathode(cực -) được nối

chung với nhau vào một điểm, được đưa chân ra ngoài để kết nối với mạch điện. 8 cực còn lại trên mỗi led đơn được đưa thành 8 chân riêng, cũng được đưa ra ngoài để kết nối với mạch điện. Nếu led 7 đoạn có Anode(cực +) chung, đầu chung này được nối với +Vcc, các chân còn lại dùng để điều khiển trạng thái sáng tắt của các led đơn, led chỉ sáng khi tín hiệu đặt vào các chân này ở mức 0. Nếu led 7 đoạn có Cathode(cực -) chung, đầu chung này được nối xuống Ground (hay Mass), các chân còn lại dùng để điều khiển trạng thái sáng tắt của các led đơn, led chỉ sáng khi tín hiệu đặt vào các chân này ở mức 1



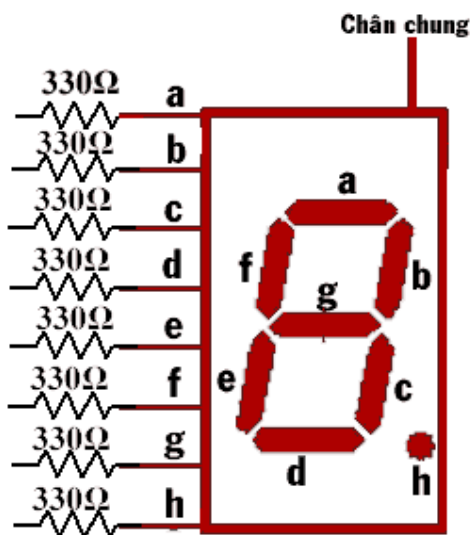
Anode(cực +) chung



Cathode(cực -) chung

Vì led 7 đoạn chứa bên trong nó các led đơn, do đó khi kết nối cần đảm bảo dòng qua mỗi led đơn trong khoảng 10mA-20mA để bảo vệ led. Nếu kết nối với nguồn 5V có thể hạn dòng bằng điện trở 330Ω trước các chân nhận tín hiệu điều khiển.

Sơ đồ vị trí các led được trình bày như hình dưới:



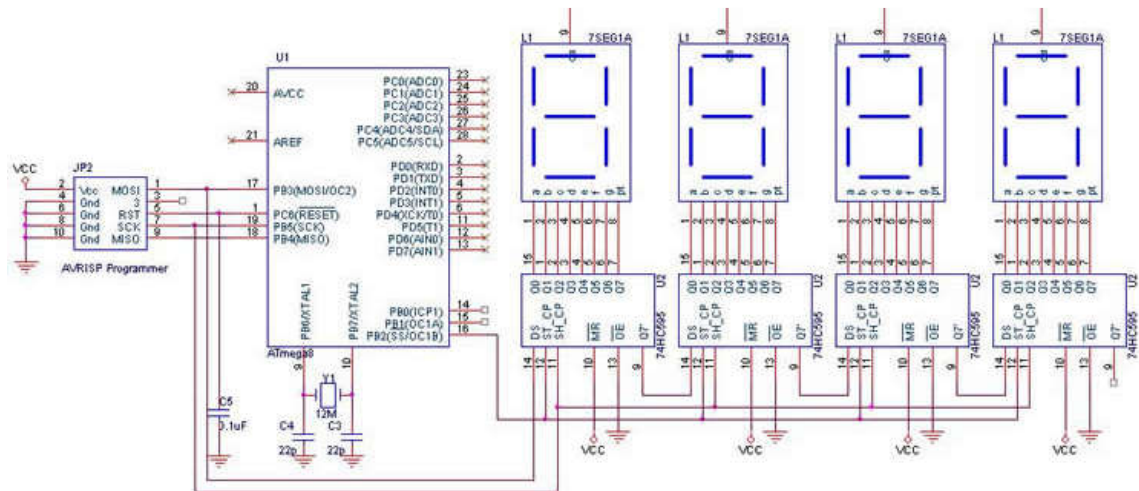
Các điện trở 330Ω là các điện trở bên ngoài được kết nối để giới hạn dòng điện qua led nếu led 7 đoạn được nối với nguồn 5V. Chân nhận tín hiệu a điều khiển led a sáng tắt, ngõ vào b để điều khiển led b. Tương tự với các chân và các led còn lại.

Ngõ nhận tín hiệu điều khiển của led 7 đoạn có 8 đường, vì vậy có thể dùng 1 Port nào đó của Vi điều khiển để điều khiển led 7 đoạn. Như vậy led 7 đoạn nhận một dữ liệu 8 bit từ Vi điều khiển để điều khiển hoạt động sáng tắt của từng led đơn trong nó, dữ liệu được xuất ra điều khiển led 7 đoạn thường được gọi là "mã hiển thị led 7 đoạn". Có hai kiểu mã hiển thị led 7 đoạn: mã dành cho led 7 đoạn có Anode(cực +) chung và mã dành cho led 7 đoạn có Cathode(cực -) chung. Chẳng hạn, để hiển thị số 1 cần làm cho các led ở vị trí b và c sáng, nếu sử dụng led 7 đoạn có Anode chung thì phải đặt vào hai chân b và c điện áp là 0V(mức 0) các chân còn lại được đặt điện áp là 5V(mức 1), nếu sử dụng led 7 đoạn có Cathode chung thì điện áp(hay mức logic) hoàn toàn ngược lại, tức là phải đặt vào chân b và c điện áp là 5V(mức 1).

Nếu kết nối mỗi một Port của Vi điều khiển với 1 led 7 đoạn thì tối đa kết nối được 4 led 7 đoạn. Mặt khác nếu kết nối như trên sẽ hạn chế khả năng thực hiện các công việc khác của Vi điều khiển. Cho nên cần phải kết nối, điều khiển nhiều led 7 đoạn với số lượng chân điều khiển từ Vi điều khiển càng ít càng tốt. Có hai giải pháp: một là sử dụng các IC chuyên dụng cho việc hiển thị led 7 đoạn, hai là kết nối nhiều led 7 đoạn vào cùng một đường xuất tín hiệu hiển thị. Nội phần này sẽ đề cập đến cách kết nối nhiều led 7 đoạn theo giải pháp thứ 2.

Mắt người có đặc điểm sinh lí là chỉ thu nhận 24 hình/giây để tổng hợp các hình ảnh về thế giới xung quanh. Nếu một tín hiệu ánh sáng có chu kì sáng tắt hơn 24 lần trong 1 giây, mắt người luôn cảm nhận đó là một nguồn sáng liên tục. Để minh họa cho điều này, bạn hãy lấy các chương trình đã thực hiện với led đơn và làm ngắn thời gian delay lại, đến một giá trị nào đó bạn sẽ thấy các led đều sáng liên tục.

Để kết nối nhiều led 7 đoạn vào vi điều khiển thực hiện như sau: nối tất cả các chân nhận tín hiệu của tất cả các led 7 đoạn (chân abcdefgh) cần sử dụng vào cùng 1 Port, trong ví dụ, 8 led 7 đoạn có các chân nhận tín hiệu cùng được nối với P0. Dùng các ngõ ra còn lại của Vi điều khiển điều khiển on/off cho led 7 đoạn, mỗi ngõ ra điều khiển ON/OFF cho 1 led 7 đoạn,(ON: led 7 đoạn được cấp nguồn để hiển thị, OFF: led 7 đoạn bị ngắt nguồn nên không hiển thị được).

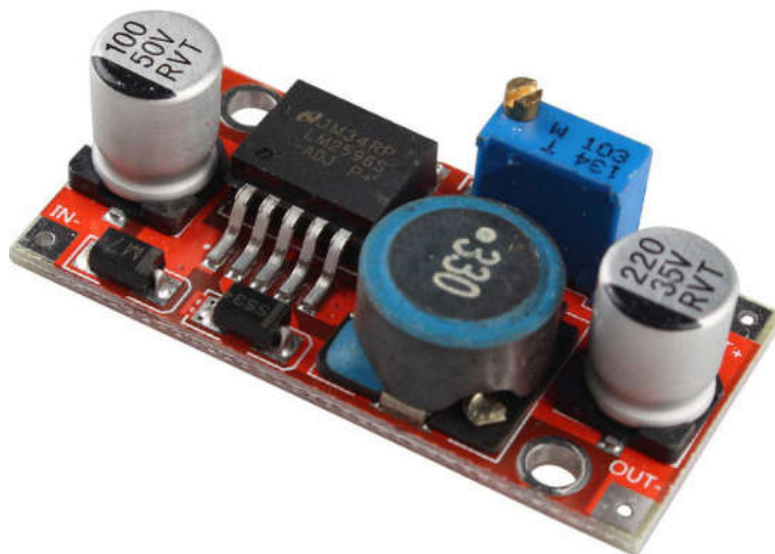


Kết nối Led 7 thanh với IC 74HC595

3.1.4. Module Nguồn LM2596

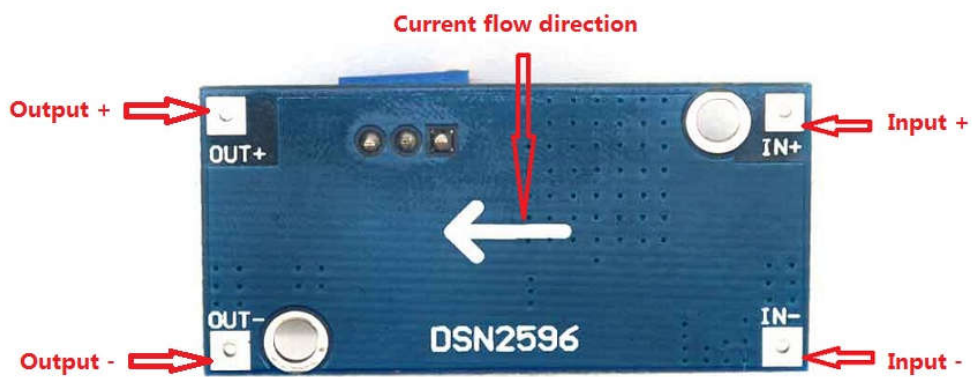
Mạch Giảm Áp LM2596 là module giảm áp có khả năng điều chỉnh được dòng ra đến 3A. **LM2596** là IC nguồn tích hợp đầy đủ bên trong. Tức là khi cấp nguồn 9v vào module, sau khi giảm áp ta có thể lắp được nguồn $3A < 9v$...như 5V hay 3.3V.

- Module nguồn không sử dụng cách ly
- Nguồn đầu vào từ 4V - 35V.
- Nguồn đầu ra: 1V - 30V.
- Dòng ra Max: 3A
- Kích thước mạch: 53mm x 26mm
- Đầu vào: INPUT +, INPUT-
- Đầu ra: OUTPUT+, OUTPUT-



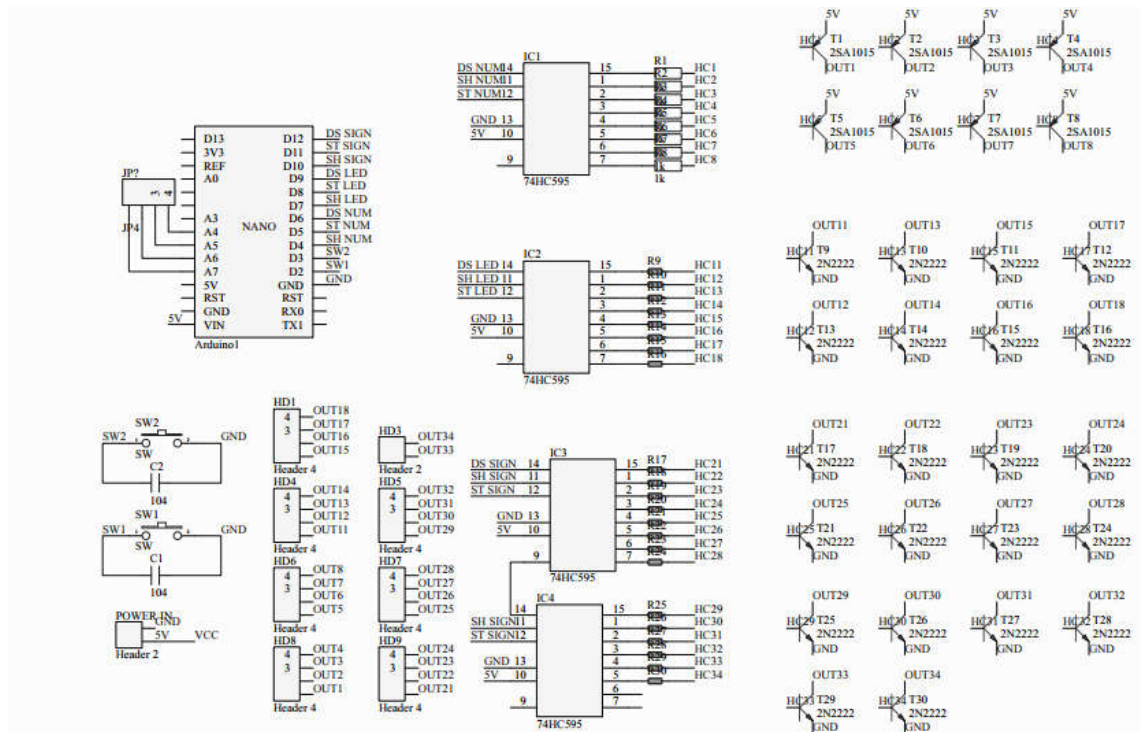
Module LM2596

Chúng ta chỉ cần cấp nguồn thô vào chân INPUT+, INPUT- rồi nhận nguồn ra từ chân OUTPUT+, OUTPUT-

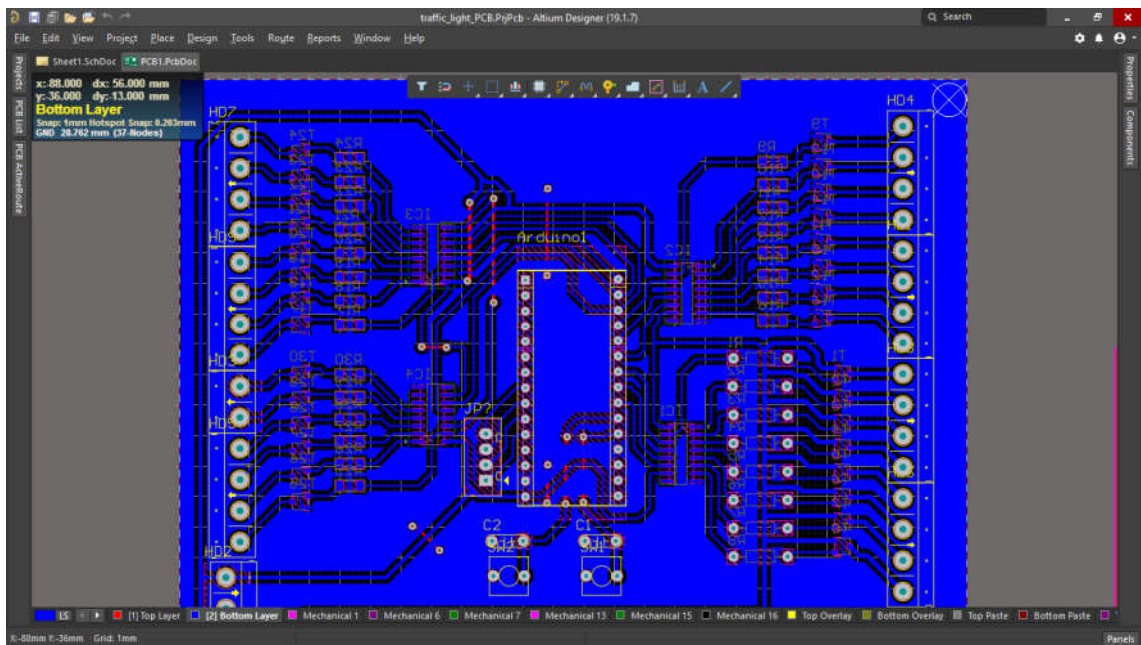


Hướng dẫn nối dây cho LM2596

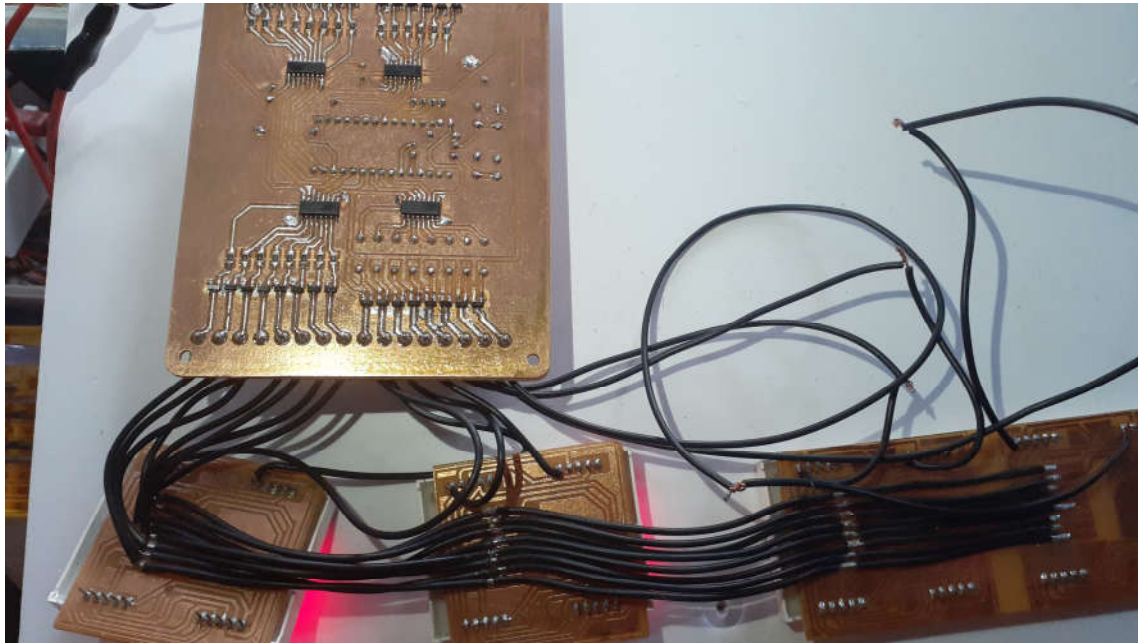
3.2. Sơ đồ mạch điện:



Sơ đồ mạch điện được thiết kế trên phần mềm Altium



Mạch sau khi được Layout



Mạch thực tế sau khi hàn, nối dây

3.3. Chương trình điều khiển:

Chương trình điều khiển được thiết kế trên phần mềm Visual Studio Code, với những tính năng vượt trội như mã nguồn mở, được hỗ trợ bởi cộng đồng Microsoft, phần mềm này đáp ứng tốt những nhu cầu về thiết kế chương trình điều khiển.

3.3.1. Khởi khai báo

```
#include <Arduino.h>
#include <TimerOne.h>

#define NUMBER_SHCP    4
#define NUMBER_STCP    5
#define NUMBER_DS      6
#define NUMBERDATA_SHCP 7
#define NUMBERDATA_STCP 8
#define NUMBERDATA_DS   9
#define LEDSIGN_SHCP    10
#define LEDSIGN_STCP    11
#define LEDSIGN_DS      12
```

```

#define btn1      2
#define btn2      3
#define YELLOW_DURATION  4
#define GREEN_DURATION  17
#define RED_DURATION  21
enum colors { RED, GREEN, YELLOW, BLACK };
class Lights {
public:
    colors color = YELLOW;
    int secondsRemaining = 0;
    Lights() {}
    Lights( colors color, int seconds) :color(color), secondsRemaining(seconds) {}
};
Lights LightA, LightC, LightB, LightD;
Lights  LightRed(RED,  RED_DURATION),  LightYellow(  YELLOW,
YELLOW_DURATION)
    , LightGreen(GREEN, GREEN_DURATION),LightBlack(BLACK,10) ;
    unsigned          char          font[11]          =          {
0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F,0x80 };
    byte no[8] = {0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f};
    byte aaa[10] = { 1,3,5,7,2,4,6,8};
    bool isStart = 0;
    bool isTrain = 0;
    uint16_t timetemp =1;
    uint32_t timetemp2 =1;
    uint32_t timetemp3 =1;
    uint16_t counting = 1;
    uint8_t turnred = 0;
    uint8_t turngreen = 0;
    uint16_t leddata = 0xFF;

```

```

uint8_t numberdigit = 16;
uint8_t numberdata = 16;
uint8_t digitData[8] ;

int led13 = 0;
void display();
void calculate();
void serialprint();
void wait();
void start();
void train();
inline void updateledturn(uint8_t r, uint8_t g);
void shiftoutdigit(uint8_t *a);//
int c=1;

```

3.3.2. Khởi cài đặt

```

void setup(){
    Serial.begin(9600);
    pinMode(btn1,INPUT_PULLUP);
    pinMode(btn2,INPUT_PULLUP);
    attachInterrupt(0,start,RISING);
    attachInterrupt(1,train,RISING);
    for(int i = 4 ; i < 18 ; i++){
        pinMode(i,OUTPUT);
    }
    digitalWrite(4,0);
    digitalWrite(5,0);
    digitalWrite(6,0);
    digitalWrite(7,1);
    digitalWrite(8,1);

```

```

    digitalWrite(9,1);
}

```

3.3.3. Các chương trình con

```

void start()
{
    while(!btn1){}
    delay(50);
    while(!btn1){}
    if(!isStart)
    {
        LightA = LightGreen;
        LightB = LightGreen;
        LightC = LightRed;
        LightD = LightRed;
    }
    isStart ^= 1 ;
    led13 ^=1;
}

void train()
{
    isTrain ^= 1 ;
    led13 ^=1;
}

void wait()
{

    if(isTrain)
    {

```

```

updateledturn(1,0);
LightD = LightRed;
LightC = LightRed;
if((millis()-timetemp)>500)
{
    digitalWrite(16,counting%2);
    digitalWrite(17,!counting%2);
    counting++;
}
}else
{
    //digitoff
    updateledturn(0,0);
    if((millis()-timetemp)>500)
    {
        if(counting%2)
        {
            LightA = LightYellow;
            LightB = LightYellow;
            LightC = LightYellow;
            LightD = LightYellow;
        }
        else
        {
            LightA = LightBlack;
            LightB = LightBlack;
            LightC = LightBlack;
            LightD = LightBlack;
        }
    }
}

```



```

        counting++;
    }
}
LightC.secondsRemaining = 88;
LightA.secondsRemaining = 88;

}

void calculate()
{
    if(isTrain)
    {
        //digit show --
        updateledturn(1,0);
        LightA = LightGreen;
        LightB = LightGreen;
        LightC = LightRed;
        LightD = LightRed;
        if((millis()-timetemp)>500)
        {
            digitalWrite(16,counting%2);
            digitalWrite(17,!counting%2);
            counting++;
        }
        LightC.secondsRemaining = 88;
        LightA.secondsRemaining = 88;
    }else
    {
        if (LightA.secondsRemaining == 0)
        {

```

```

switch (LightA.color)
{
case RED:
    LightA = LightGreen;
    LightB = LightGreen;
    LightD = LightRed;
    LightC = LightRed;
    updateledturn(0,1);
    break;
case YELLOW:
    LightA = LightRed;
    LightC = LightGreen;
    LightB = LightRed;
    LightD = LightGreen;
    updateledturn(1,0);
    break;
case GREEN:
    LightA = LightYellow;
    LightB = LightYellow;
    updateledturn(1,0);
    break;
case BLACK:
    break;
}
}
if (LightC.secondsRemaining == 0)
{
    switch (LightC.color)
    {

```

```

    case RED:
        LightC = LightGreen;
        LightA = LightRed;
        LightD = LightGreen;
        LightB = LightRed;
        break;
    case YELLOW:
        LightC = LightRed;
        LightA = LightGreen;
        LightD = LightRed;
        LightB = LightGreen;
        break;
    case GREEN:
        LightC = LightYellow;
        LightD = LightYellow;
        break;
    case BLACK:
        break;
    }
}

LightA.secondsRemaining--;
LightC.secondsRemaining--;
LightB.secondsRemaining--;
LightD.secondsRemaining--;
}
}

void display()
{

```

```
// AAAB BBCC CDDD RG
```

```
switch (LightA.color)
{
    case RED:
        leddata |= 0x40;
        leddata &=~ 0xB0;
        break;
    case YELLOW:
        leddata |= 0x20;
        leddata &=~ 0xD0;
        break;
    case GREEN:
        leddata |= 0x90;
        leddata &=~ 0x60;
        break;
    case BLACK:
        leddata &=~ 0xFF;
        break;
}
```

```
switch (LightC.color)
{
    case RED:
        leddata |= 0x02;
        leddata &=~ 0x05;
        break;
    case YELLOW:
        leddata |= 0x01;
        leddata &=~ 0x06;
        break;
}
```

```

    case GREEN:
        leddata |= 0x04;
        leddata &= ~ 0x03;
        break;
    case BLACK:
        break;
}

digitalWrite(LED_SIGN_STCP, 1);
shiftOut(LED_SIGN_DS, LED_SIGN_SHCP, LSBFIRST, leddata);
shiftOut(LED_SIGN_DS, LED_SIGN_SHCP, LSBFIRST, leddata << 8);
digitalWrite(LED_SIGN_STCP, 1);

//display 8 digit
digitData[0]    = LightA.secondsRemaining / 10;
digitData[1]    = LightA.secondsRemaining % 10;
digitData[2]    = LightC.secondsRemaining / 10;
digitData[3]    = LightC.secondsRemaining % 10;
digitData[4]    = LightA.secondsRemaining / 10;
digitData[5]    = LightA.secondsRemaining % 10;
digitData[6]    = LightC.secondsRemaining / 10;
digitData[7]    = LightC.secondsRemaining % 10;
}

inline void updateLedTurn(uint8_t r, uint8_t g)
{
    r?leddata |= (1<<13) : leddata &= ~ (1<<13) ;
    g?leddata |= (1<<14) : leddata &= ~ (1<<14) ;
}

void shiftOutDigit(uint8_t digit[])

```

```

{
    // digit = 0 -> 9
    //font
    byte fontt[10];
    for(int i = 0; i < 10; i++)
    {
        fontt[i] = font[digit[i]];
    }
    for(int i = 0 ; i < 8; i++)
    {
        digitalWrite(NUMBERDATA_STCP, 1);
        shiftOut(NUMBERDATA_DS, NUMBERDATA_SHCP, LSBFIRST, fontt[i]);
        digitalWrite(NUMBERDATA_STCP, 0);

        digitalWrite(NUMBER_STCP, 1);
        shiftOut(NUMBER_DS, NUMBER_SHCP, MSBFIRST, 0xff);
        digitalWrite(NUMBER_STCP, 0);
        delayMicroseconds(2000);

        digitalWrite(NUMBER_STCP, 1);
        shiftOut(NUMBER_DS, NUMBER_SHCP, MSBFIRST, no[i]);
        digitalWrite(NUMBER_STCP, 0);
    }
}

void serialprint()
{
    Serial.print(LightA.secondsRemaining);
    Serial.print("    ");
    Serial.print(LightA.color);
}

```

```

Serial.print("  ");
Serial.print(LightC.secondsRemaining);
Serial.println("  ");
Serial.print(LightA.color);
Serial.print("  ");

}

```

3.3.4. Chương trình chính

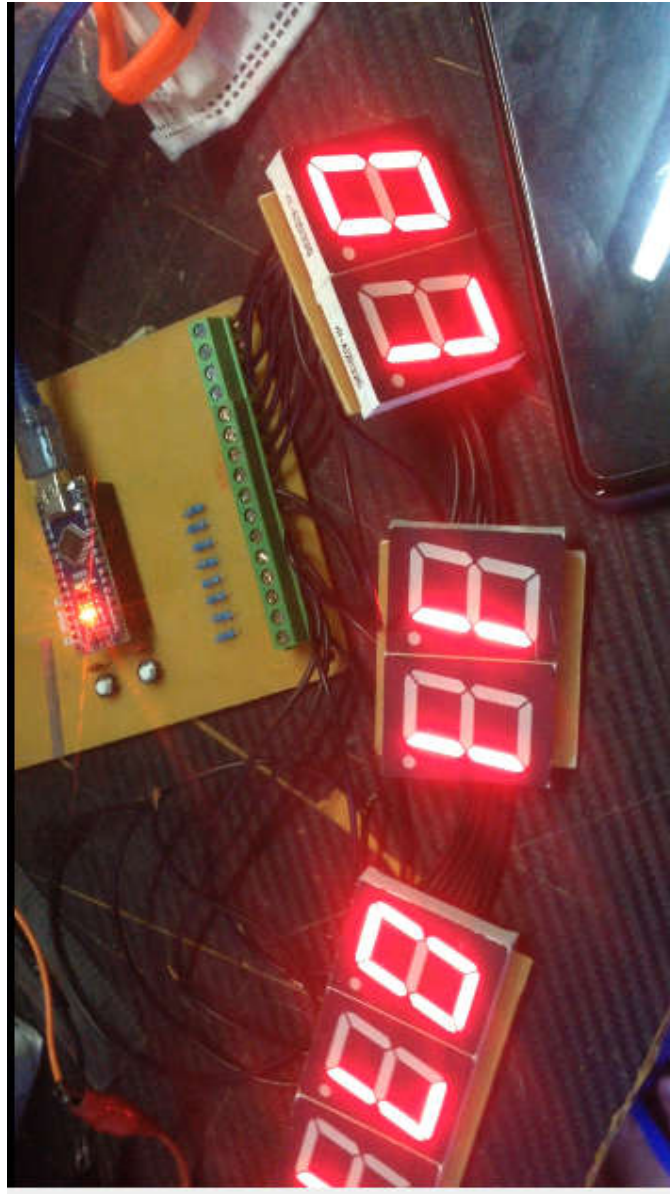
```

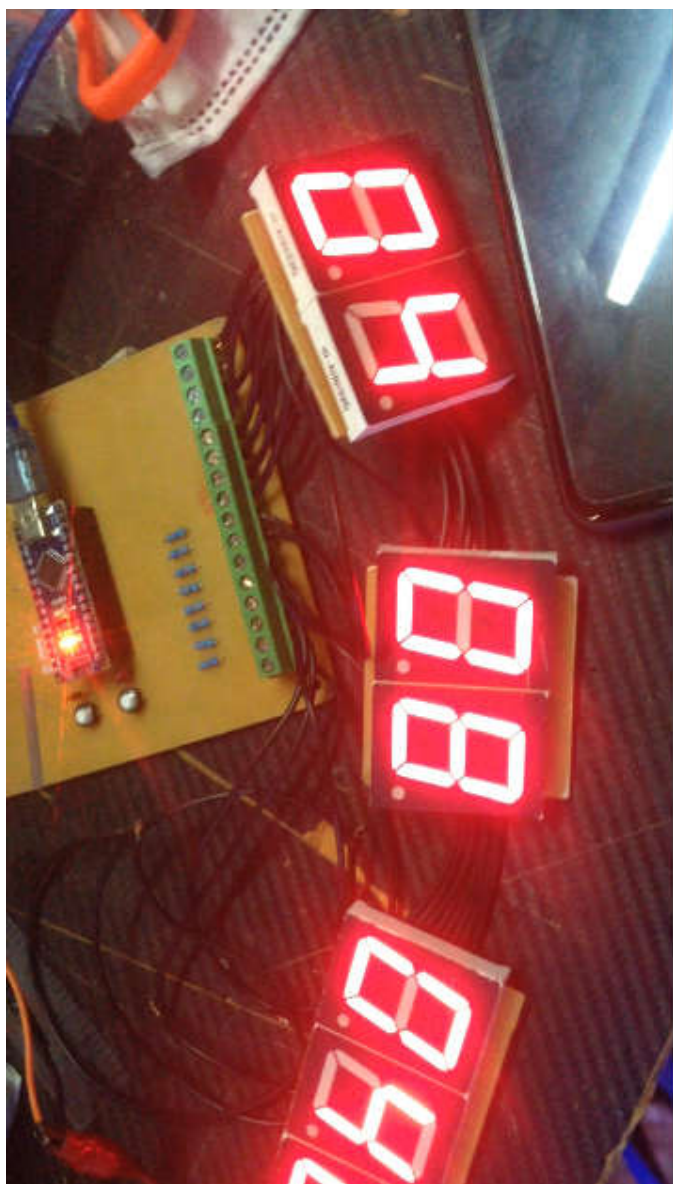
void loop()
{
    shiftoutdigit(digitData);
    if(isStart) //
    {
        if((millis()-timetemp2)>500)
        {
            timetemp2 = millis();
            calculate();
            serialprint();
        }
        display();
    }else
    {
        wait();
        display();
        serialprint();
    }
}

```

3.4. Kết quả thử nghiệm

Sau khi đổ chương trình vào Arduino, nhóm chúng em thu được thành quả như sau:





CHƯƠNG 4: ĐÁNH GIÁ ĐỀ TÀI VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết quả đạt được

4.2. Hạn chế còn tồn tại

4.3. Hướng phát triển đề tài

Tài liệu tham khảo

Phụ lục

