

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KỸ THUẬT MÁY TÍNH

----- ○  ○ -----



BÁO CÁO ĐỒ ÁN CUỐI KỲ
MÔN: VI XỬ LÝ – VI ĐIỀU KHIỂN

HỆ THỐNG ĐÈN GIAO THÔNG TẠI NGÃ TƯ SỬ DỤNG AT89S52

Giảng viên hướng dẫn: **Ths. Trần Hoàng Lộc**

Lớp học phần: CE103.O24

Nhóm sinh viên thực hiện : Nhóm 12

Bùi Văn Sinh - 22521245

Nguyễn Ngọc Minh Quốc - 22521214

Trịnh Thu Thanh - 22521369

TP. Hồ Chí Minh, tháng 06 năm 2024

LỜI MỞ ĐẦU

Ngày nay, với sự tiến bộ của khoa học kỹ thuật, con người đã tìm ra nhiều giải pháp để giải phóng sức lao động và nâng cao đời sống vật chất lẫn tinh thần. Một trong những giải pháp đó là công nghệ tự động hóa dựa trên các ứng dụng kỹ thuật điện, kỹ thuật điều khiển vào trong máy móc và các thiết bị phục vụ đời sống như tivi, tủ lạnh, máy giặt, ô tô, ... Trong các thiết bị thông minh đó một thành phần không thể thiếu chính là vi điều khiển.

Vi điều khiển ra đời từ năm 1976 do hãng Intel phát triển, cho đến nay nó đã trở nên vô cùng phổ biến trong các thiết bị máy móc với rất nhiều chủng loại và được nhiều công ty khác nhau trên thế giới sản xuất. Nổi bật trong đó có vi điều khiển họ 8051 của Intel, vi điều khiển ARM của công ty cùng tên ARM, vi điều khiển PIC của Microchip, vi điều khiển AVR của Atmel. Vi điều khiển có thể được xem như là một máy tính nhỏ gọn bao gồm vi xử lý, các thành phần bộ nhớ (ROM, RAM), các thiết bị ngoại vi khác nhằm hỗ trợ cho vi điều khiển giao tiếp với các thiết bị trong việc nhận tín hiệu và xuất các tín hiệu điều khiển.

Để có thể áp dụng và củng cố các kiến thức đã học vào thực tế, nhóm chúng em đã lựa chọn và thực hiện đề tài **“Hệ thống đèn giao thông tại ngã tư đèn AT89S52”** trong 3 đề tài đồ án của môn học Vi xử lý – Vi điều khiển. Trong quá trình hoàn thành đồ án, nhóm chúng em đã học được và trải nghiệm rất nhiều trong nhiều quá trình khác nhau, song kiến thức và kinh nghiệm còn có hạn nên sẽ không tránh khỏi những sai sót. Chúng em rất mong nhận được sự giúp đỡ và tham khảo ý kiến từ thầy cùng các bạn để có thể hoàn thiện đồ án tốt hơn nữa.

LỜI CẢM ƠN

Nhóm chúng em xin gửi lời tri ân sâu sắc đến thầy Trần Hoàng Lộc đã tận tình hướng dẫn, truyền đạt kiến thức và hỗ trợ chúng em trong suốt quá trình thực hiện đồ án "Đèn giao thông". Nhờ sự nhiệt huyết, tâm huyết và những lời khuyên quý giá của thầy, chúng em đã hoàn thành được đồ án một cách tốt đẹp nhất.

Đồ án của chúng em không chỉ là một bài kiểm tra kiến thức mà còn là cơ hội thực hành và ứng dụng những kiến thức đã học vào một sản phẩm thực tế. Biết rõ hơn Vi xử lý – Vi điều khiển có vai trò quan trọng như thế nào đối với đời sống con người, được trải nghiệm làm một dự án nhỏ khi còn ở trên ghế nhà trường.

Chúng em ý thức được rằng, trong bài báo cáo này vẫn còn tồn tại một số thiếu sót và hạn chế. Do vậy, chúng em rất mong nhận được những góp ý quý báu từ thầy để hoàn thiện bài báo cáo hơn nữa. Một lần nữa, nhóm chúng em xin chân thành cảm ơn sự hỗ trợ quý báu của thầy trong quá trình học tập và nghiên cứu.

Xin trân trọng cảm ơn!

Thành phố Hồ Chí Minh, ngày 30 tháng 6 năm 2024

Nhóm sinh viên thực hiện

MỤC LỤC

LỜI MỞ ĐẦU	2
LỜI CẢM ƠN	3
MỤC LỤC	4
DANH MỤC HÌNH ẢNH	6
DANH MỤC BẢNG	7
CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI	8
1.1. Mô tả đề tài	8
1.2. Mục tiêu đề tài	8
1.3. Phương pháp thực hiện đề tài	9
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	10
2.1. Phương thức quét Led	10
2.2. Bộ định thời/Bộ đếm trong 8051	10
2.3. Hoạt động ngắt trong 8051	13
2.4. Giao tiếp với bàn phím 4x4	15
2.5. Thông tin về linh kiện trong đề tài	15
2.5.1. Vi điều khiển AT89S52	15
2.5.2. Mạch Led giao thông 5V	18
2.5.3. Led 7 đoạn 2 số 0.36 inch anode chung	19
2.5.4. Keyboard matrix phím 4x4	19
2.5.5. Thạch anh	20
CHƯƠNG 3. HIỆN THỰC ĐỀ TÀI MẠCH ĐIỀU KHIỂN HỆ THỐNG ĐÈN GIAO THÔNG	21
3.1. Sơ đồ khối hệ thống	21
3.2. Mô phỏng đề tài trên phần mềm Proteus	22
3.2.1. Giới thiệu phần mềm Protues	22
3.2.2. Kết nối linh kiện trên phần mềm	22
3.2.2.1. Tạo project trên phần mềm Proteus	22
3.2.2.2. Chọn linh kiện nối dây	24

3.3. Hiện thực đề tài trên phần cứng.....	26
3.4. Viết chương trình nạp cho vi xử lý AT89S52.....	29
3.5. Nguyên lý hoạt động	38
Chương 4. TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN	40
4.1. Tổng kết	40
4.1.1. Kết quả	40
4.1.2. Đánh giá.....	40
4.2. Hướng phát triển	40
TÀI LIỆU THAM KHẢO	41

DANH MỤC HÌNH ẢNH

Hình 1: Chuyển đổi trạng thái chương trình khi có ngắt.....	14
Hình 2: Sơ đồ ngắt trong 8051	14
Hình 3: AT89S52	17
Hình 4: Sơ đồ chân AT89S52	17
Hình 5: Mạch Led giao thông 5V.....	19
Hình 6: Led 7 đoạn 2 số anode chung.....	19
Hình 7: Keypad 4x4 mềm	20
Hình 8: Thạch anh 12 MHz.....	20
Hình 9: Sơ đồ khối hệ thống.....	21
Hình 10: Giao diện đặt tên và chọn đường dẫn project.....	23
Hình 11: Tạo firmware project.....	23
Hình 12: Giao diện Schematic Capture ban đầu	24
Hình 13: Giao diện source code ban đầu.....	24
Hình 14: Led đôi 7 đoạn hiển thị thời gian và đèn hiển thị màu xanh đỏ vàng.....	25
Hình 15: Chip 8051 xử lý của mạch đèn giao thông.....	25
Hình 16: Keypad 4x4 dùng để cài đặt thời gian hiển thị đèn	25
Hình 17: Schematic hoàn chỉnh của mạch.....	26
Hình 18: Chạy thử mạch	26
Hình 19: PCB Layout và vị trí hiển thị thiết bị trong mô phỏng.....	28
Hình 20: Mạch xem trước trong giao diện mô phỏng.....	28

DANH MỤC BẢNG

<i>Bảng 2.2.1 Các thanh ghi sử dụng trong việc định thời của 8051.....</i>	<i>10</i>
<i>Bảng 2.2.2 Các bit chức năng trong thanh ghi điều khiển TCON.....</i>	<i>11</i>
<i>Bảng 2.2.3 Các bit chức năng trong thanh ghi điều khiển TMOD.....</i>	<i>11</i>
<i>Bảng 2.2.4 Các mode định thời trong 8051</i>	<i>12</i>

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Mô tả đề tài

Trong đời sống ngày càng phát triển, việc di chuyển đi lại là điều cần thiết. Đèn giao thông là thứ tất yếu để đảm bảo vấn đề di chuyển ổn định, hạn chế xảy ra va chạm, tai nạn. Đồng thời, người điều khiển có thể sử dụng thiết bị đếm thời gian, cài đặt hẹn giờ để điều chỉnh phù hợp với mật độ giao thông. Đề tài **“Hệ thống đèn giao thông tại ngã tư dùng AT89S52”** tập trung vào việc phát triển một hệ thống đèn giao thông tại một ngã tư, đồng thời điều chỉnh được thời gian của từng đèn xanh, đỏ, vàng. Hệ thống này được vận hành bởi vi điều khiển AT89S52, một vi điều khiển 8-bit thuộc họ 8051 của Atmel. Ngoài ra, hệ thống còn sử dụng mạch led giao thông 5V để tăng tính thẩm mỹ và trực quan sinh động cho đề tài.

1.2. Mục tiêu đề tài

Như chúng ta đã biết, tín hiệu đèn giao thông là thứ không thể thiếu trong cuộc sống đi lại giữa đường phố, đảm bảo trật tự an toàn giao thông. Nhờ có đèn giao thông, các tuyến đường sẽ được lưu thông ổn định, đảm bảo mật độ phương tiện đi lại tránh ùn tắc, gây tai nạn. Hệ thống đèn giao thông tại ngã tư dùng chip xử lý 8051 còn nhiều hạn chế bởi khả năng đáp ứng xử lý yêu cầu của con chip nên không thể ứng dụng ngoài thực tế một cách hiệu quả. Tuy nhiên, hệ thống có thể được ứng dụng trong việc nghiên cứu học tập, là bước đệm đầu tiên trong việc tìm hiểu các con vi xử lý, thiết bị điện tử phục vụ việc học tập các môn học, định hướng phát triển nghiên cứu chuyên sâu.

Ứng dụng các kiến thức đã học trong môn Vi xử lý – Vi điều khiển, từ đó xây dựng được các cột đèn giao thông tại ngã tư. Cụ thể:

- Led hiển thị thời gian hiển thị được số giây từ 0 đến 99.
- Có đèn màu xanh – vàng – đỏ trực quan theo bộ đếm thời gian.
- Đèn có thời gian được cài đặt sẵn chạy tự động.
- Người điều khiển có thể sử dụng bàn phím cài đặt lại thời gian phù hợp yêu cầu.
- Có thể trực tiếp reset trạng thái bởi người điều khiển.

1.3. Phương pháp thực hiện đề tài

Sử dụng Proteus 8 Professional 8.12: Proteus 8 Professional 8.12 cung cấp môi trường lập trình assembly trên 8051, vẽ Schematic với các linh kiện có sẵn khá đầy đủ. Phiên bản 8.12 cp1 các công cụ hỗ trợ được tích hợp để lập trình, biên dịch. Việc lập trình, phát triển chương trình điều khiển cho vi điều khiển AT89C52 được thực hiện hoàn toàn trên phần mềm Proteus 8 Professional 8.12.

Lựa chọn phần cứng: Phần cứng trung tâm của đồ án là vi điều khiển AT89S52. AT89S52 là bộ vi điều khiển 8 bit, công suất thấp, hiệu suất cao, dựa trên công nghệ CMOS với dải tần hoạt động từ 0 đến 33MHz. Con chip này có thể hoạt động ở hai chế độ lựa chọn nguồn điện hoạt động. AT89S52 tương thích rất tốt với Kit phát triển 8051 - KIT AT89S52 SOCKET. KIT AT89S52 SOCKET là một Kit nạp code có giá thành hợp lý, dễ sử dụng, những yếu tố trên khiến nhóm quyết định chọn vi điều khiển này để thực hiện đồ án.

Lập trình: Chương trình được viết hoàn toàn bằng ngôn ngữ Assembly dành cho 8051. Việc viết chương trình bằng ngôn ngữ C gây cho nhóm khá nhiều khó khăn trong quá trình phát triển và sửa lỗi. Cũng có khá ít tài liệu tham khảo về cách viết chương trình assembly 8051 hiệu quả trên mạng. Tuy vậy khi phát triển đồ án bằng Assembly 8051 giúp nhóm phải tính toán kỹ lưỡng, luôn luôn kiểm thử chương trình để tránh bị lỗi, từ đó giúp khả năng lập trình tốt hơn.

Thiết kế: Mạch được thiết kế cẩn thận và kiểm tra kỹ lưỡng trước khi in để đảm bảo chất lượng và tránh hư hỏng. PCB của đồ án được thiết kế trên Proteus 8 Professional 8.12, Proteus cung cấp giao diện trực quan và các tính năng vẽ PCB khá tốt. PCB Layout và 3D Visualizer trong Proteus đáp ứng tốt trong quá trình thực hiện đồ án của nhóm.

Thử nghiệm và hiệu chỉnh: Quá trình thử nghiệm và hiệu chỉnh là bước cuối cùng nhưng không kém phần quan trọng. Chương trình đã được chạy thử nghiệm trên test board để kiểm tra các trường hợp có thể gây ra lỗi, đảm bảo rằng sản phẩm cuối cùng sẽ hoạt động mượt mà và ổn định.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Phương thức quét Led

Nguyên lý quét LED (LED multiplexing) dựa trên việc bật và tắt các LED một cách tuần tự với tốc độ đủ nhanh để mắt người không phát hiện được sự nhấp nháy, hiện tượng này gọi là hiện tượng lưu ảnh ở mắt - là hiện tượng hình ảnh tiếp tục xuất hiện trong mắt sau một thời gian không tiếp xúc với hình ảnh gốc. Kỹ thuật này cho phép điều khiển nhiều LED bằng cách sử dụng ít chân I/O hơn so với số LED.

Nhóm sử dụng Led 7 đoạn 2 số nên việc quét Led cũng khá là dễ dàng, nhóm sử dụng Timer trong 8051 để delay 1 giây trong quá trình quét Led.

2.2. Bộ định thời/Bộ đếm trong 8051

Trong vi điều khiển 8051 có hai timer 16-bit: Timer 0 và Timer1; mỗi timer có bốn chế độ hoạt động. Hai timer này thường được dùng làm bộ định thời để tạo thời gian trễ (delay), làm bộ đếm các sự kiện xảy ra bên ngoài vi điều khiển hoặc dùng để tạo tốc độ baud cho cổng nối tiếp (UART).

SFR	Chức năng	Địa chỉ	Địa chỉ theo bit
TCON	Điều khiển	88H	Có
TMOD	Chọn chế độ	89H	Không
TL0	Byte thấp của Timer 0	8AH	Không
TL1	Byte thấp của Timer 1	8BH	Không
TH0	Byte cao của Timer0	8CH	Không
TH1	Byte cao của Timer1	8DH	Không

Bảng 2.2.1 Các thanh ghi sử dụng trong việc định thời của 8051

Thanh ghi TCON là một thanh ghi 8 bit và có thể định địa chỉ theo bit. Nó chứa các bit điều khiển và trạng thái của bộ định thời 0 và bộ định thời 1, và các bit dùng cho ngắt như mô tả trong Bảng 4-2. Trong đó, chỉ có bốn bit cao (TCON.7- TCON.4) được dùng để điều khiển các bộ định thời hoạt động hoặc chứa các cờ tràn. Bốn bit thấp của TCON (TCON.3 - TCON.0) được sử dụng để phát hiện và khởi động các ngắt.

BIT	Tên	Địa chỉ	Chức năng
7	TF1	8FH	Cờ báo tràn của Timer1, được lập bởi phần cứng khi có tràn Timer1, được xóa bởi phần mềm, hoặc bởi phần cứng khi phục vụ chương trình ngắt
6	TR1	8EH	Bit điều khiển hoạt động của Timer1, được lập hoặc xóa bởi phần mềm để điều khiển Timer1 hoạt động hay ngưng hoạt động
5	TF0	8DH	Cờ báo tràn của Timer0
4	TR0	8CH	Bit điều khiển hoạt động của Timer1
3	IE1	8BH	Được sử dụng trong phục vụ ngắt
2	IT1	8AH	
1	IE0	89H	
0	IT0	88H	

Bảng 2.2.2 Các bit chức năng trong thanh ghi điều khiển TCON

Thanh ghi TMOD dùng để thiết lập chế độ hoạt động cho bộ định thời 0 và bộ định thời 1. Thanh ghi này là thanh ghi 8 bit, không được định địa chỉ theo bit và được nạp một lần bởi phần mềm ở thời điểm bắt đầu của một chương trình để khởi động chế độ hoạt động của bộ định thời. Trong đó 4 bit thấp dành cho bộ Timer 0 và 4 bit cao dành cho bộ Timer 1, mỗi bộ gồm 2 bit dành cho việc thiết lập chế độ định thời, còn 2 bit dùng để chọn chức năng định thời/đếm và điều khiển cổng. Cụ thể chức năng của từng bit của thanh ghi TMOD được mô tả trong bảng 2.2.3 sau.

BIT	Tên	Timer	Chức năng
7	GATE	1	Bit điều khiển cổng. Khi GATE = 1, Timer chỉ hoạt động khi INT1 ở mức cao.
6	C/T'	1	Bit chọn chức năng đếm hoặc định thời: 1 – Đếm sự kiện; 0 – Định thời
5	M1	1	Bit chọn chế độ cho Timer 1
4	M0	1	Bit chọn chế độ cho Timer 1
3	GATE	0	Bit điều khiển cổng. Khi GATE = 1, Timer chỉ hoạt động khi INT1 ở mức cao.
2	C/T'	0	Bit chọn chức năng đếm hoặc định thời: 1 – Đếm sự kiện; 0 – Định thời
1	M1	0	Bit chọn chế độ cho Timer 0
0	M0	0	Bit chọn chế độ cho Timer 0

Bảng 2.2.3 Các bit chức năng trong thanh ghi điều khiển TMOD

Mỗi Timer có bốn chế độ hoạt động và chế độ hoạt động sẽ được chọn phụ thuộc vào giá trị của các bit M1 và M0 tương ứng với mỗi Timer theo bảng 2.2.4 sau.

M1	M0	Chế độ	Mô tả
0	0	0	Chế độ định thời 13 bit
0	1	1	Chế độ định thời 16 bit
1	0	2	Chế độ định thời tự động nạp lại 8 bit
1	1	3	Chế độ định thời chia tách: Timer 0: TL0 là bộ định thời 8 bit điều khiển bởi chế độ của Timer 0; TH0 là bộ định thời 8 bit điều khiển bởi chế độ của Timer 1 Timer 1: ngưng hoạt động

Bảng 2.2.4 Các mode định thời trong 8051

Chế độ định thời 0 là chế độ định thời 13 bit, chế độ này được thiết kế nhằm cung cấp khả năng tương thích với bộ vi điều khiển trước là 8048. Chế độ này hoạt động bằng cách ghép 5 bit thấp của thanh ghi TLx và thanh ghi THx để tạo thành bộ định thời 13 bit, 3 bit cao của thanh ghi TLx không được sử dụng.

Chế độ định thời 1 là chế độ định thời 16 bit, hoạt động tương tự như chế độ 0. Khi nhận được xung nhịp bộ định thời sẽ đếm lên từ giá trị chứa bên trong thanh ghi định thời, nếu thanh ghi định thời chưa được nạp thì bộ định thời sẽ đếm lên từ 0000H. Cờ tràn sẽ bật lên 1 khi bộ định thời chuyển trạng thái từ FFFFH về 0000H, và bộ định thời sẽ tiếp tục đếm. Cờ tràn là bit TFX trong thanh ghi TCON có thể đọc hoặc ghi bằng phần mềm. Bit có trọng số lớn nhất (MSB) của giá trị trong các thanh ghi định thời là bit 7 của THx và bit có trọng số thấp nhất (LSB) là bit 0 của thanh ghi TLx. Giá trị bắt đầu của bộ định thời được nạp vào các thanh ghi TLx và THx trước khi cho phép bộ định thời hoạt động. THx và TLx có thể đọc và ghi bằng phần mềm.

Chế độ định thời 2 là chế độ định thời tự nạp lại 8 bit, trong chế độ này TLx hoạt động như một timer 8 bit, còn THx vẫn giữ nguyên giá trị bắt đầu của đếm. Và sẽ nạp lại giá trị từ THx và TLx mỗi khi bộ định thời tràn từ FFH đến 00H, đồng thời bật cờ tràn TFX tại thời điểm này. Chế này được sử dụng thường xuyên bởi vì bộ định thời sẽ tự động hoạt động sau khi cài đặt các thanh ghi TMOD và THx mà không cần phải nạp giá trị vào các thanh ghi định thời mỗi khi cờ tràn bật lên.

Chế độ định thời 3 là chế độ định thời chia sẻ và trong chế độ này 2 bộ định thời trong 8051 có hoạt động khác nhau. Bộ định thời 0 ở chế độ 3 được chia thành 2 bộ định

thời 8 bit hoạt động độc lập với nhau TL0 có cờ báo tràn là TF0, còn TH0 có cờ báo tràn là TF1. Khi bộ định thời 0 ở chế độ 3 thì bộ định thời 1 không hoạt động định thời bởi vì cờ tràn TF1 đã được nối với TH0, nhưng có thể được khởi động bằng thiết lập bit TR1 bằng 1 trong chế độ khác chế độ 3.

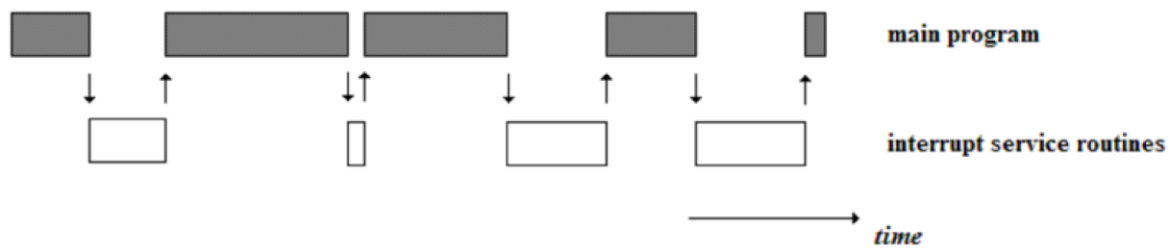
Chế độ định thời 3 nhằm cung cấp thêm một bộ định thời 8 bit nữa cho 8051. Khi đó 8051 có 3 bộ định thời, bộ định thời 0 ở chế độ 3 hoạt động như 2 bộ định thời, và bộ định thời 1 không hoạt động định thời nên có thể sử dụng để tạo tốc độ baud trong giao tiếp cổng nối tiếp (UART) hoặc sử dụng như một bộ đếm thông thường mà không yêu cầu ngắt.

2.3. Hoạt động ngắt trong 8051

Một ngắt là một sự kiện bên trong hoặc bên ngoài làm ngắt bộ vi điều khiển để báo cho nó biết rằng có sự kiện cần dịch vụ của nó.

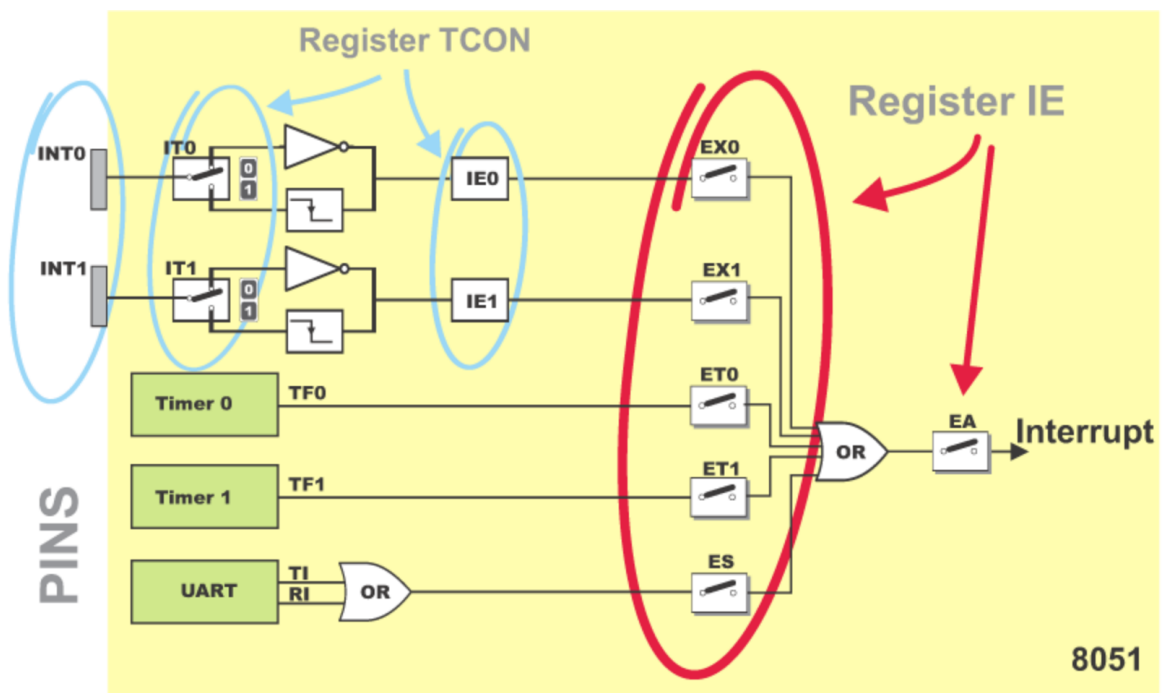
Trong phương pháp này thì mỗi khi có một thiết bị bất kỳ cần đến dịch vụ thì nó báo cho bộ vi điều khiển bằng cách gửi một tín hiệu ngắt. Khi nhận được tín hiệu ngắt thì bộ vi điều khiển có thể ngưng tạm thời việc thực thi chương trình nó đang thực hiện để chuyển sang phục vụ thiết bị. Và khi không có thiết bị nào cần phục vụ thì CPU sẽ nghỉ ngơi để giảm thiểu năng lượng tiêu thụ. Ngoài ra, trong phương pháp sử dụng ngắt chúng ta còn có thể cài đặt độ ưu tiên cho các thiết bị nào cần phục vụ trước, hoặc chúng ta có thể cài đặt để bỏ qua ngắt ở một số thiết bị nào không cần thiết.

Chương trình đi cùng với ngắt gọi là chương trình phục vụ ngắt (ISR) hoặc chương trình quản lý ngắt. Chương trình phục vụ ngắt được thực thi để đáp ứng một ngắt như bộ định thời, cổng giao tiếp nối tiếp, các ngắt ngoài. Chương trình phục vụ ngắt được kết thúc bằng lệnh “RETI”, khi đó chương trình chính sẽ tiếp tục chạy tại nơi nó bị dừng trước khi ngắt xảy ra. Chương trình chính thực hiện ở mức nền còn chương trình phục vụ ngắt thực thi ở mức ngắt như sơ đồ ở dưới.



Hình 1: Chuyển đổi trạng thái chương trình khi có ngắt

Vi điều khiển 8051 có 6 nguồn yêu cầu ngắt: 1 ngắt reset có mức ưu tiên cao nhất, 2 nguồn ngắt ngoài, 2 nguồn ngắt từ 2 timer và 1 nguồn ngắt từ cổng nối tiếp. Khi hệ thống reset thì tất cả các nguồn ngắt đều bị cấm (trừ ngắt reset), để cho phép các ngắt hoạt động cần lập trình nạp các giá trị phù hợp vào thanh ghi IE. Khi có nhiều nguồn ngắt yêu cầu ngắt đồng thời, hoặc một ngắt xảy ra trong khi một ngắt khác đang được phục vụ, vi điều khiển sẽ đáp ứng bằng cách xét theo vòng tuần tự (cố định thứ tự các ngắt) hoặc dựa vào mức ưu tiên các ngắt đã được lập trình trước đó để xác định ngắt nào cần thực hiện.



Hình 2: Sơ đồ ngắt trong 8051

Khi có nhiều nguồn ngắt yêu cầu ngắt đồng thời, hoặc một ngắt xảy ra trong khi một ngắt khác đang được phục vụ, vi điều khiển sẽ đáp ứng bằng cách xét theo vòng tuần tự (cố định thứ tự các ngắt) hoặc dựa vào mức ưu tiên các ngắt đã được lập trình trước đó để xác định ngắt nào cần thực hiện.

2.4. Giao tiếp với bàn phím 4x4

Bàn phím sử dụng nhiều trong các thiết bị trong cuộc sống từ những thiết bị đơn giản như máy tính bỏ túi, hay đến các thiết bị như điện thoại di động, nút bấm điều khiển đầu DVD. Đặc điểm của bàn phím 4x4 là các nút được nối với nhau theo cột chung và hàng chung, vì vậy khi lập trình thì người dùng có thể quét theo hàng hoặc theo cột tùy thích. Nếu như quét theo hàng thì sẽ đọc giá trị theo cột và ngược lại.

2.5. Thông tin về linh kiện trong đề tài

2.5.1. Vi điều khiển AT89S52

a. Thông tin chung

- AT89S52 là một trong những bộ vi điều khiển phổ biến của họ Atmel, vi điều khiển AT89S52 là vi điều khiển công nghệ CMOS 8 bit có bộ nhớ Flash 8kb và bộ nhớ RAM 256 byte.
- AT89S52 có thể hoạt động ở tần số tối đa là 33MHz bằng cách sử dụng bộ dao động bên ngoài. Giống như các bộ vi điều khiển khác, nó có các chân GPIO, ba bộ định thời 16 bit, một cổng giao tiếp UART song công (truyền tín hiệu theo cả 2 hướng), ba bộ định thời 16 bit, bộ tạo dao động trên chip.
- Thiết bị này cung cấp 32 chân GPIO có thể được sử dụng làm chân đầu vào - đầu ra digital hoặc cũng có thể sử dụng cùng một chân cho các chức năng thay thế.

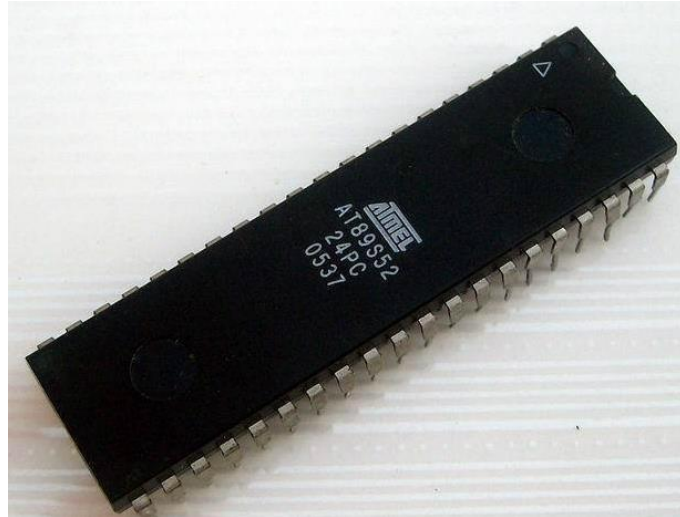
b. Thông số kỹ thuật.

- Kiến trúc vi xử lý: PIC 8 bit
- Số lượng chân: 40
- RAM: 256 byte
- EEPROM/HEF: Không

- Bộ nhớ chương trình: 8 KBytes
- Tốc độ CPU: 33 MHz
- Bộ tạo dao động bên trong: Không
- Số bộ so sánh: 2
- ADC và DAC: Không
- Số lượng chân I O có thể lập trình: 32
- Bộ hẹn giờ Watchdog: Không
- Tần số dao động bên ngoài: 23 MHz (tối đa)
- Độ phân giải của PWM: Không
- Số bộ hẹn giờ 16 bit: 3
- Loại bộ nhớ chương trình: Flash
- Module UART: 1
- Cờ ngắt nguồn: Có
- Cặp con trỏ dữ liệu : Có
- Điện áp hoạt động: 4V - 5,5 V
- Nhiệt độ hoạt động: -55 0 C - 125 0 C
- Một số tính năng chi tiết nổi bật được liệt kê dưới đây:
 - o Là bộ vi điều khiển công nghệ CMOS hiệu suất cao tích hợp công nghệ Flash.
 - o Hoạt động ở dải điện áp rộng 4 - 5.5V, vì vậy nó là một IC công suất thấp.
 - o Thiết bị hỗ trợ lập trình bên trong ở cả chế độ page và byte của bộ nhớ Flash.
 - o Tần số hoạt động lên đến 33MHz nhưng có thể thay đổi để tiết kiệm năng lượng.
 - o Module có thời gian lập trình nhanh với 10.000 chu kỳ đọc/ghi.
 - o Bộ nhớ RAM 256×8 bit.
 - o Giao tiếp nối tiếp thông qua module UART song công.
 - o Nó có một chân reset, ba bộ định thời 16 bit và tám bộ ngắt.
 - o AT89S52 có hai chế độ nguồn. Đầu tiên là chế độ nhàn rỗi, trong đó thiết bị xử lý dừng hoạt động trong khi ngoại vi vẫn tiếp tục

hoạt động. Thứ hai là chế độ tắt nguồn sẽ tạm dừng bộ dao động và các chức năng khác và lưu nội dung RAM.

- Bộ đếm thời gian Watchdog để hoạt động khởi động thiết bị từ chế độ ngủ và có thể được kích hoạt hoặc hủy kích hoạt thông qua lập trình.



Hình 3: AT89S52

c. Sơ đồ chân

(T2) P1.0	1	40	VCC
(T2 EX) P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
P1.4	5	36	P0.3 (AD3)
(MOSI) P1.5	6	35	P0.4 (AD4)
(MISO) P1.6	7	34	P0.5 (AD5)
(SCK) P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	$\overline{\text{EA}}/\text{VPP}$
(TXD) P3.1	11	30	ALE/PROG
(INT0) P3.2	12	29	PSEN
(INT1) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (A14)
(T1) P3.5	15	26	P2.5 (A13)
$\overline{\text{WR}}$ P3.6	16	25	P2.4 (A12)
$\overline{\text{RD}}$ P3.7	17	24	P2.3 (A11)
XTAL2	18	23	P2.2 (A10)
XTAL1	19	22	P2.1 (A9)
GND	20	21	P2.0 (A8)

Hình 4: Sơ đồ chân AT89S52

- Bộ vi điều khiển 8-bit AT89S52 có 3 package. Nhưng sơ đồ sơ đồ chân là giống nhau cho cả ba. Chi tiết cấu hình chân trong bảng được đề cập dưới đây:

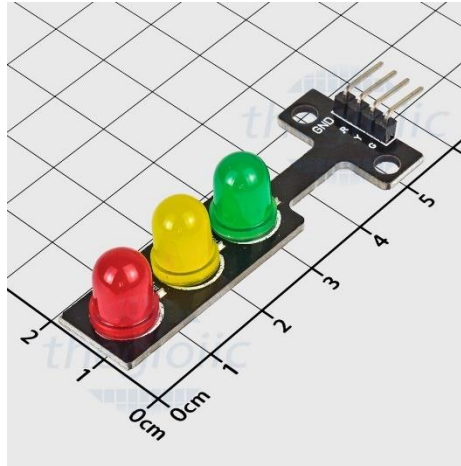
Số chân	Tên chân	Đặc điểm
32 - 39	Port 0	8 chân Địa chỉ và Dữ liệu / GPIO
1 - 8	Port 1	8 chân GPIO
21 - 28	Port 2	8 chân GPIO
10 - 17	Port 3	8 chân GPIO
9	RST	Chân Reset
18	XTAL2	Chân đầu ra của bộ tạo dao động bên ngoài
19	XTAL1	Chân đầu vào bộ tạo dao động bên ngoài
20	GND	Chân nối đất
40	VCC	Chân cấp điện
31	EA/VPP	Kích hoạt truy xuất bên ngoài / chân cấp nguồn kích hoạt Flash
30	ALE/PROG	Chân chốt địa chỉ / Chân lập trình flash
29	PSEN	Chân cho phép lưu chương trình

Bảng 3.2.3 Đặc điểm chân của AT89S52

2.5.2. Mạch Led giao thông 5V

a. Thông số kỹ thuật

- Kích thước: 56 * 21 * 11 nm.
- Màu sắc: Đỏ - Vàng – Xanh
- LED: 3 Led đục – Đường kính bóng led 8nm.
- Điện áp: 5V.
- Trọng lượng: 25 gram.
- 4 chân nối: GND Red Yellow Green.



Hình 5: Mạch Led giao thông 5V

2.5.3. Led 7 đoạn 2 số 0.36 inch anode chung

a. Thông số kỹ thuật

- Màu đỏ.
- Anode chung.
- 2 Dot, 10 chân.
- Điện áp rơi trên LED là 2.2V.
- Dòng tối đa chạy qua mỗi LED là 25mA.
- Dòng chạy bình thường: 10mA. Nếu nguồn 5V thì mỗi Led phải nối với 1 điện trở 220R (dòng chạy qua mỗi led 13mA).



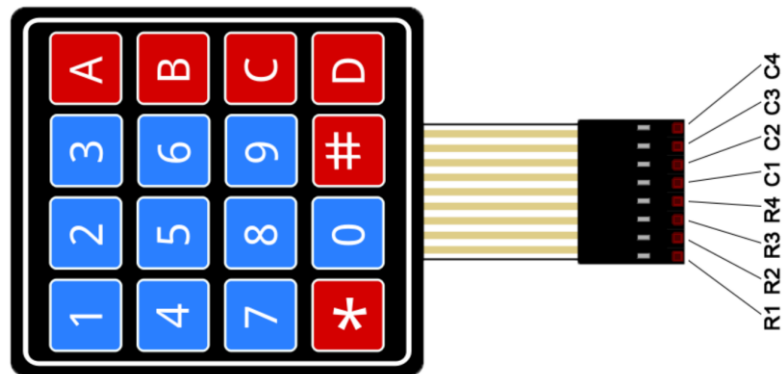
Hình 6: Led 7 đoạn 2 số anode chung

2.5.4. Keyboard matrix phím 4x4

a. Thông số kỹ thuật

- Module bàn phím ma trận 4x4 loại phím mềm.
- Độ dài cáp: 88mm.

- Nhiệt độ hoạt động 0 ~ 70oC.
- Đầu nối ra 8 chân.
- Kích thước bàn phím 77 x 69 mm.



Hình 7: Keypad 4x4 mềm

2.5.5. Thạch anh

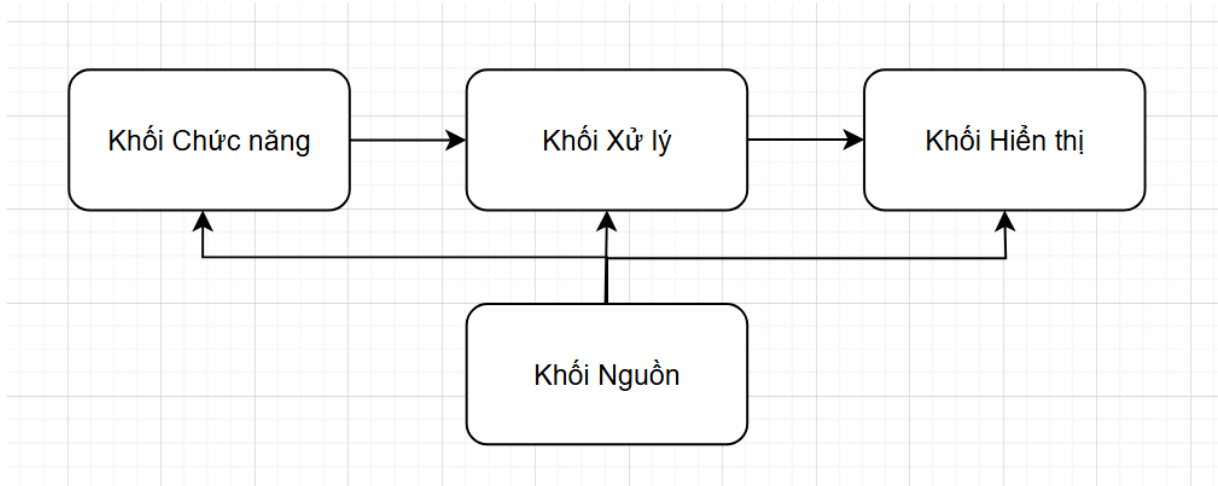
- Thông số kỹ thuật
 - Tần số: 12MHz
 - Dung sai tần số: $\pm 20\text{ppm}$
 - ESR (Equivalent series resistance): 30 Ohm
 - Nhiệt độ hoạt động: -20oC ~ 70oC



Hình 8: Thạch anh 12 MHz

CHƯƠNG 3. HIỆN THỰC ĐỀ TÀI MẠCH ĐIỀU KHIỂN HỆ THỐNG ĐÈN GIAO THÔNG

3.1. Sơ đồ khối hệ thống



Hình 9: Sơ đồ khối hệ thống

Mạch được thiết kế gồm 4 khối, bao gồm:

- **Khối nguồn:** Để mạch hoạt động tốt, cần sử dụng khối nguồn để cấp điện. Do mạch cần dùng nguồn điện áp 5V DC cấp điện nên khi thực thi sản phẩm ta cần dùng bộ chuyển USB – 5V để đảm bảo nguồn điện không bị quá tải.
- **Khối xử lý:** Sử dụng vi điều khiển AT89S52 thuộc họ vi điều khiển 8051 để làm trung tâm điều khiển hoạt động toàn mạch đi cùng với khối reset, bao gồm các nhiệm vụ: nhận tín hiệu từ khối chức năng, xử lý tín hiệu, giải mã để cho kết quả vào khối hiển thị.
- **Khối hiển thị:** Dùng để đưa ra thông tin hiển thị đèn giao thông, bao gồm thiết bị led đôi anode 7 đoạn và đèn hiển thị màu xanh – đỏ - vàng. Thiết bị led đôi 7 đoạn được cài đặt thời gian hiển thị đếm giây thông qua bộ đếm từ khối chức năng đưa vào khối xử lý cho ra kết quả.
- **Khối chức năng:** Do mạch tương tự như mạch đếm, khối này có chức năng tạo mạch dao động bên trong chip 8051 được ghép với thạch anh bên ngoài ở hai chân XTAL1 và XTAL2 (chân 18 và 19). Tần số danh định của thạch anh là 12 MHz cho hầu hết các chip của họ MCS-51. Đồng thời cài đặt thời gian hiển thị đèn thông qua keypad 4x4.

3.2. Mô phỏng đề tài trên phần mềm Proteus

3.2.1. Giới thiệu phần mềm Protues

Phần mềm Proteus là phần mềm vẽ và mô phỏng mạch điện tử được phát triển bởi Labcenter Electronics. Proteus hỗ trợ mô phỏng nhiều loại vi điều khiển phổ biến như 8051, PIC, AVR, ARM, và Arduino. Phần mềm này rất mạnh mẽ và được sử dụng rộng rãi trong việc thiết kế và kiểm tra các mạch điện tử trước khi triển khai thực tế.

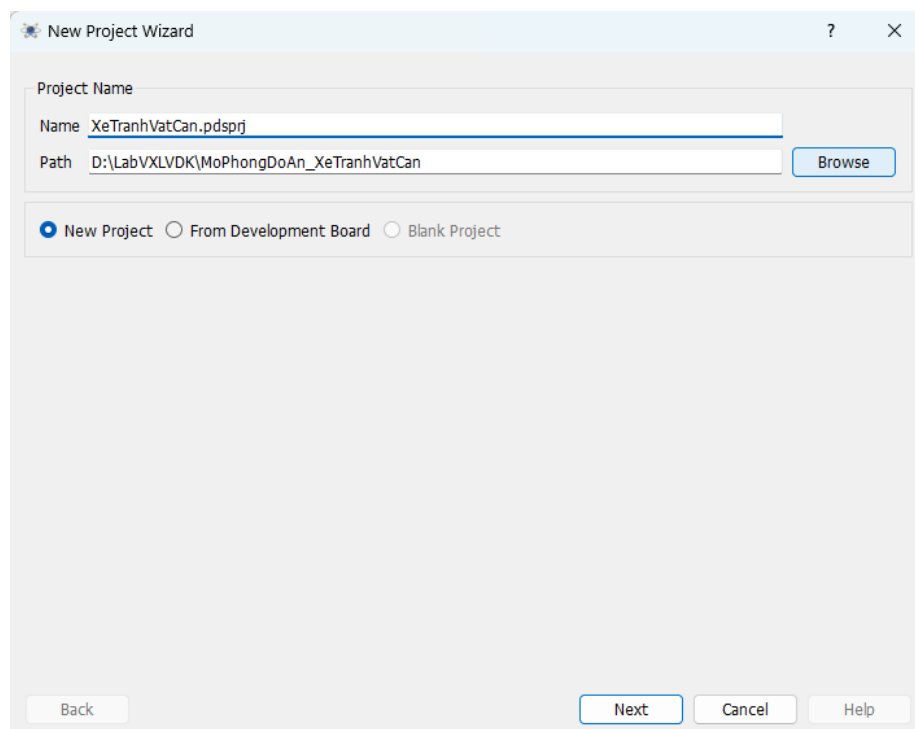
Các bước sử dụng Proteus:

- Bước 1: Khởi động chương trình Proteus Professional.
- Bước 2: Mở chương trình ISIS Professional.
- Bước 3: Lấy các linh kiện từ thư viện của Proteus.
- Bước 4: Đưa linh kiện ra ngoài màn hình thiết kế.
- Bước 5: Thay đổi thông số kỹ thuật của linh kiện điện tử cho phù hợp.
- Bước 6: Bố trí, sắp xếp lại linh kiện cho hợp lý.
- Bước 7: Nối dây.
- Bước 8: Kiểm tra sơ đồ mạch nguyên lý.
- Bước 9: Mô phỏng.

3.2.2. Kết nối linh kiện trên phần mềm

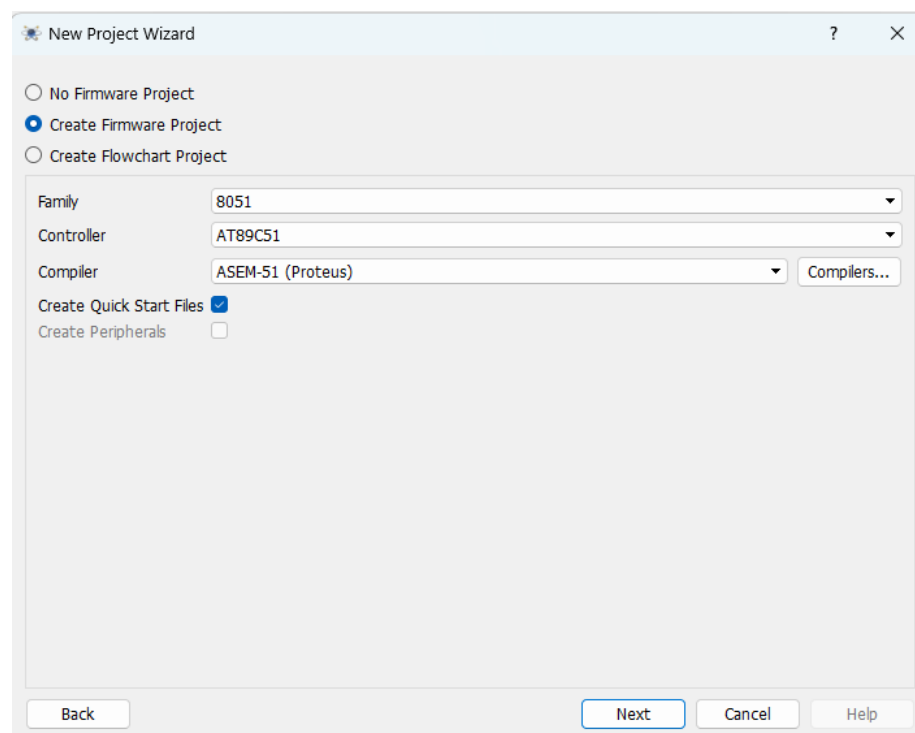
3.2.2.1. Tạo project trên phần mềm Proteus

- Bước 1: Ở màn hình giao diện “Home Page”, chọn “New Project” để tạo project mới.
- Bước 2: Đặt tên và chọn đường dẫn cho project.



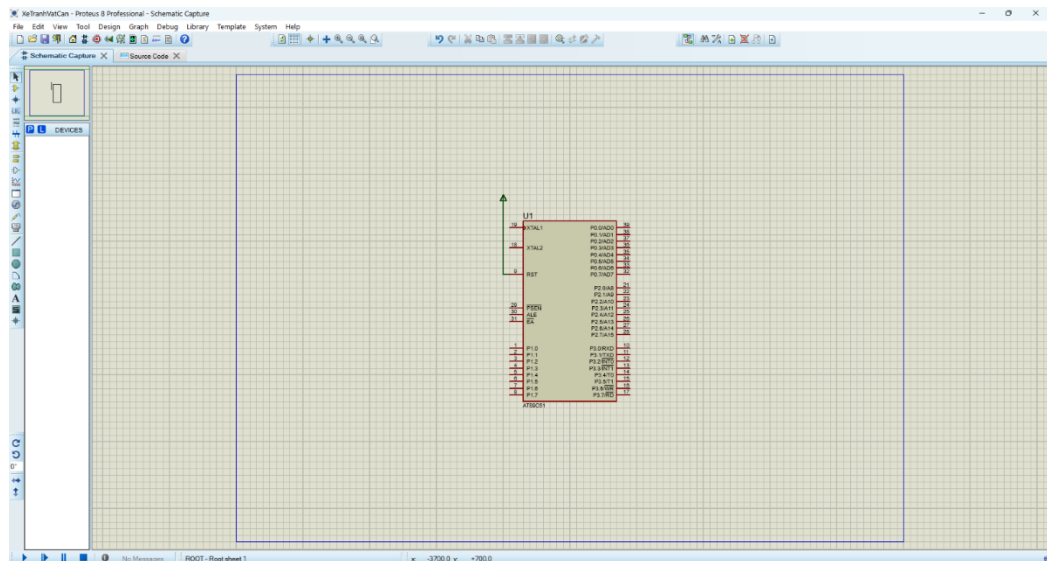
Hình 10: Giao diện đặt tên và chọn đường dẫn project

- Bước 3: Nhấn Next liên tục để chọn các thông số mặc định cho đến màn hình chọn Firmware. Chọn Family 8051, controller AT89C51 và compiler là ASEM-51.

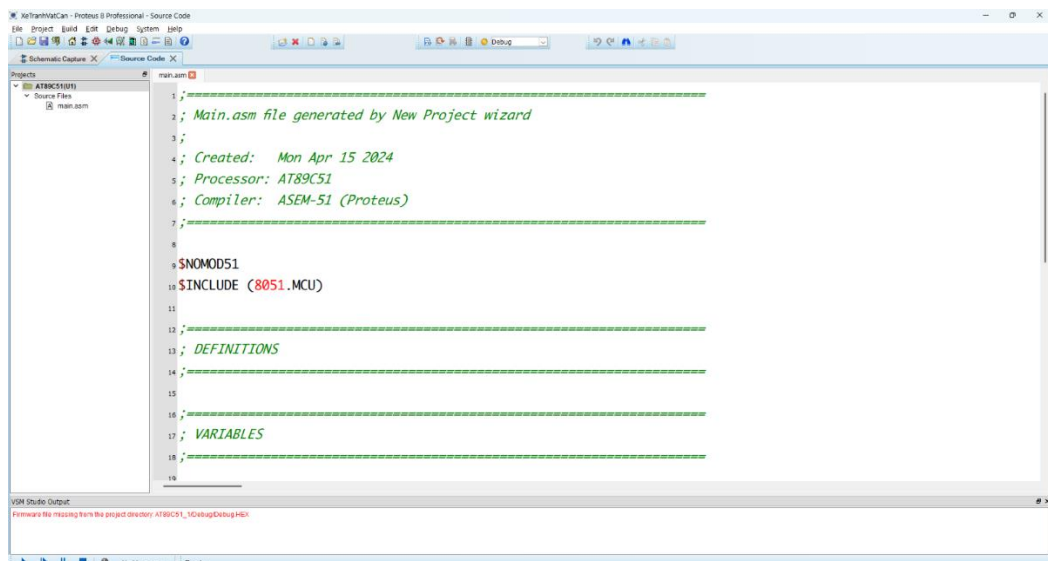


Hình 11: Tạo firmware project

- Bước 4: Tiếp tục nhấn Next. Sau đó nhấn Finish để vào giao diện Schematic Capture. Ở đây, Protues sẽ tự tạo thêm phần source code, trong đó bao gồm file main.asm.



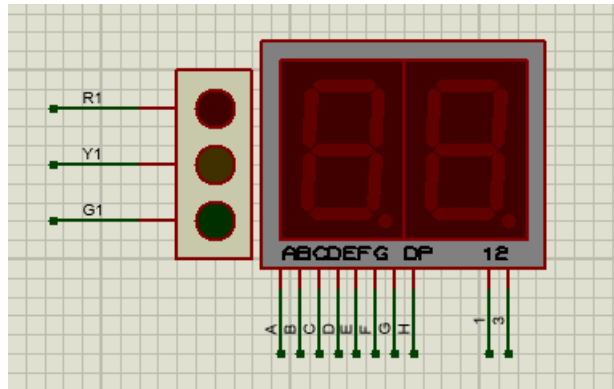
Hình 12: Giao diện Schematic Capture ban đầu



Hình 13: Giao diện source code ban đầu

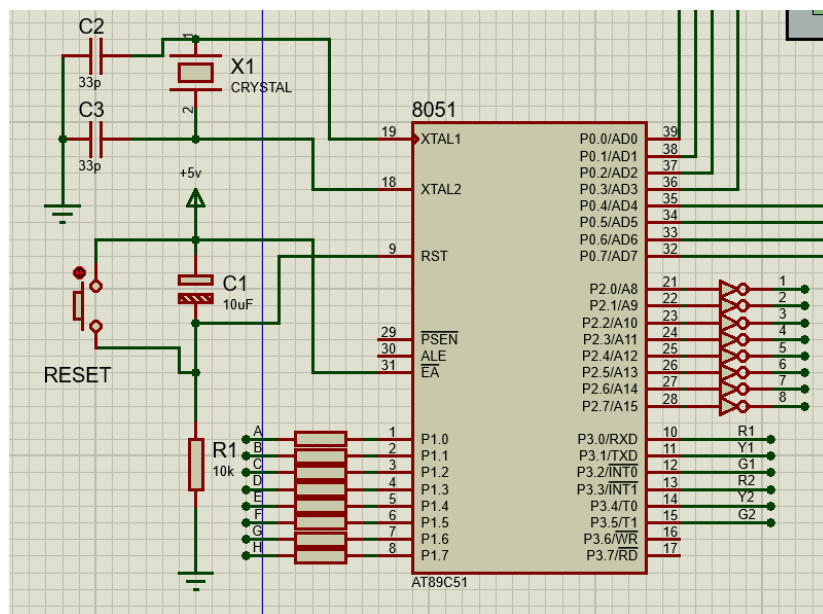
3.2.2.2. Chọn linh kiện nối dây

- Sử dụng nút P trong phần Devices để mở cửa sổ Pick Devices. Ở đây ta có thể chọn các linh kiện từ thư viện để mô phỏng.
- Tìm kiếm các linh kiện cần thiết hỗ trợ cho việc hoàn thành Đèn giao thông.

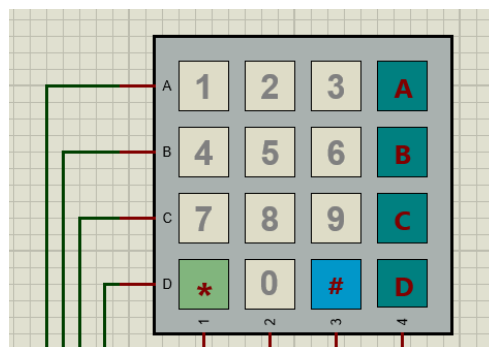


Hình 14: Led đôi 7 đoạn hiển thị thời gian và đèn hiển thị màu xanh đỏ vàng

- Trong mô phỏng thực hiện, chúng ta có thể không cần thiết phải sử dụng linh kiện thạch anh. Tuy nhiên, để tạo ra sản phẩm hoàn thiện, thạch anh là thứ cần thiết trong quá trình thiết kế mạch. Vì vậy, để đảm bảo mạch hoạt động hiệu quả, thạch anh tạo xung sẽ được đưa vào mô phỏng.

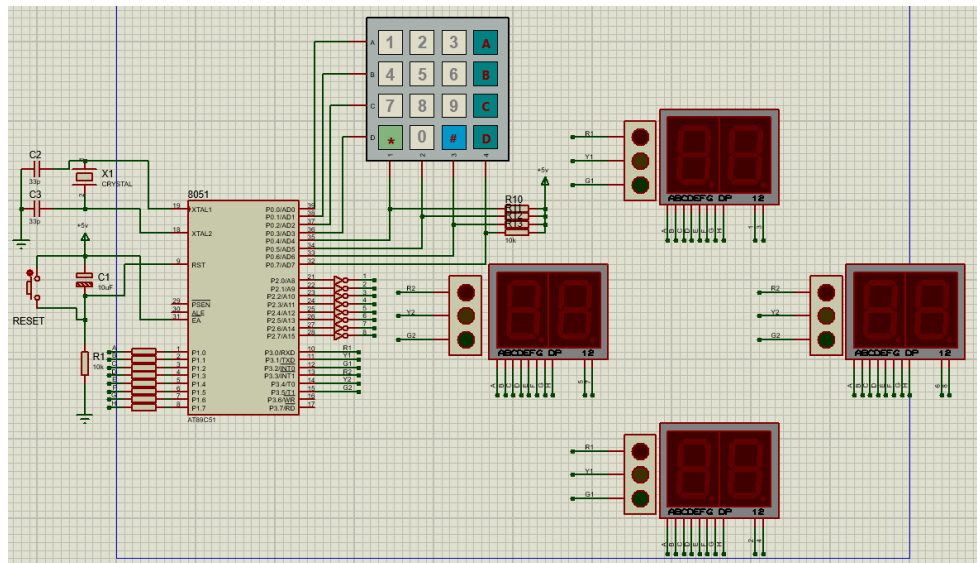


Hình 15: Chip 8051 xử lý của mạch đèn giao thông



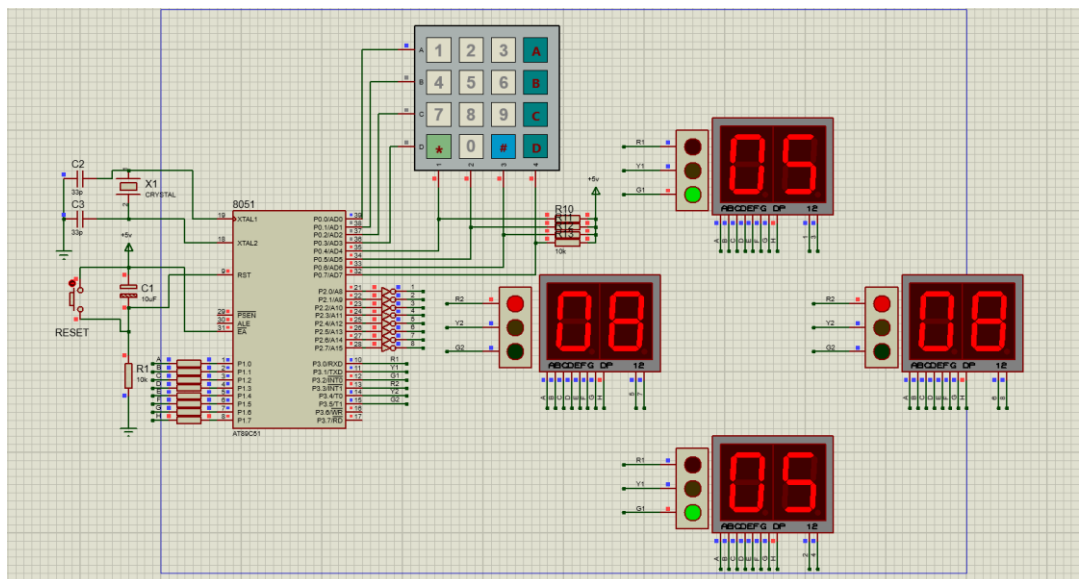
Hình 16: Keypad 4x4 dùng để cài đặt thời gian hiển thị đèn

- Bố trí, sắp xếp và nối các linh kiện lại với nhau.



Hình 17: Schematic hoàn chỉnh của mạch

- Sau khi cài đặt chương trình và chạy thử

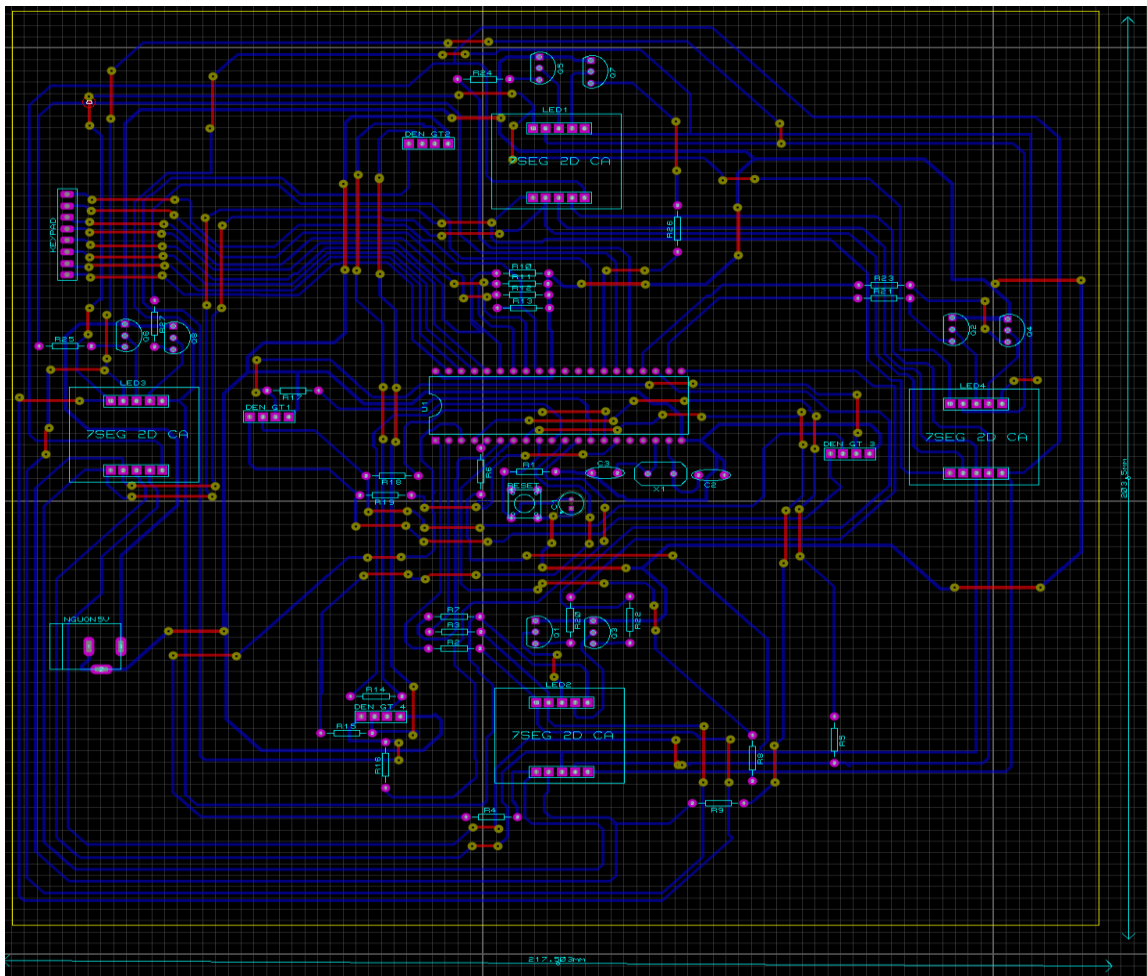


Hình 18: Chạy thử mạch

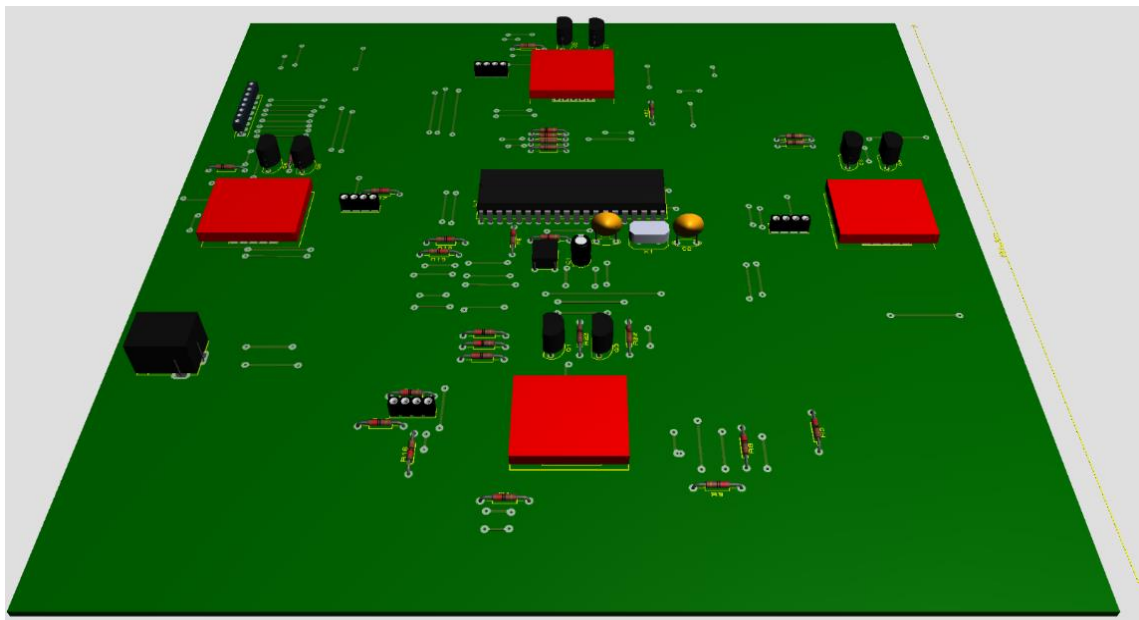
3.3. Hiện thực đề tài trên phần cứng

- **Bước 1:** Vẽ sơ đồ mạch nguyên lý.
 - o Trước tiên, cần vẽ sơ đồ nguyên lý của mạch trong Proteus, sử dụng các linh kiện có sẵn và kết nối chúng với nhau.

- **Bước 2:** Chuyển sang mô phỏng PCB.
 - Sau khi vẽ sơ đồ nguyên lý, chọn "Tools" → "Switch to PCB Layout" trong menu để chuyển sang môi trường thiết kế mạch in.
- **Bước 3:** Định vị linh kiện trên PCB.
 - Các linh kiện từ sơ đồ nguyên lý sẽ được tự động xuất hiện trên trang thiết kế PCB. Bạn cần định vị và sắp xếp chúng sao cho phù hợp trên bảng mạch in.
 - Sử dụng các công cụ như "Move" (Di chuyển) và "Rotate" (Xoay) để đặt linh kiện ở vị trí mong muốn.
- **Bước 4:** Kết nối linh kiện trên PCB.
 - Sử dụng công cụ "Route" (kết nối) để tạo các đường dẫn kết nối giữa các chân linh kiện. Kéo đường từ chân linh kiện này đến chân linh kiện khác để tạo các liên kết trên PCB.
 - Cố gắng tạo các đường dẫn ngắn và đơn giản để giảm thiểu tiềm năng gây nhiễu và giảm khả năng xảy ra lỗi.
- **Bước 5:** Thêm via và các chi tiết khác.
 - Sử dụng công cụ "Via" để thêm các Via (lỗ thông qua) nếu cần thiết để kết nối các lớp khác nhau trên PCB.
 - Thêm các chi tiết khác như đường dẫn nguồn điện, chân kết nối, logo hoặc lớp định danh để làm cho mạch in hoàn chỉnh và dễ đọc.
- **Bước 6:** Kiểm tra và xác nhận thiết kế.
 - Kiểm tra kỹ thiết kế PCB để đảm bảo không có lỗi nào, chẳng hạn như các đường dẫn không kết nối hoặc đung độ.
 - Xác nhận rằng các thông số kỹ thuật của mạch in như kích thước, độ dày, và các yêu cầu khác đều đúng.
- **Bước 7:** Xuất file Gerber.
 - Sau khi hoàn tất thiết kế, xuất file Gerber để gửi đi sản xuất mạch in. Chọn "File" -> "Export" -> "Export Gerber" và thiết lập các tùy chọn xuất file.



Hình 19: PCB Layout và vị trí hiển thị thiết bị trong mô phỏng



Hình 20: Mạch xem trước trong giao diện mô phỏng

3.4. Viết chương trình nạp cho vi xử lý AT89S52

LEDDATA EQU P1

COT1 EQU P0.4

COT2 EQU P0.5

COT3 EQU P0.6

COT4 EQU P0.7

HANG1 EQU P0.0

HANG2 EQU P0.1

HANG3 EQU P0.2

HANG4 EQU P0.3

CHUC EQU 30H

DONVI EQU 31H

DEMLED1 EQU R0

DEMLED2 EQU R1

LEDR1 EQU P3.0

LEDY1 EQU P3.1

LEDG1 EQU P3.2

LEDR2 EQU P3.3

LEDY2 EQU P3.4

LEDG2 EQU P3.5

LED1 EQU P2.0

LED2 EQU P2.1

LED3 EQU P2.2

LED4 EQU P2.3

LED5 EQU P2.4

LED6 EQU P2.5

LED7 EQU P2.6

LED8 EQU P2.7

THOIGIANXANH EQU 33h

THOIGIANVANG EQU 34h

THOIGIANDO EQU 35h

KETQUAPHIM EQU 36h

CHEDODEN EQU 37h

DENHIENTAI EQU R2

DEMTG EQU R7

ORG 000h

LJMP MAIN

```

ORG 000Bh
CALL DEMTHOIGIAN
RETI

ORG 001Bh
CALL QUETLED
RETI

ORG 0040h
MAIN:
MOV P0, #0FFh
CALL KHOITAOTIMER

CALL TATLED
SETB LED1
SETB LED2
SETB LED3
SETB LED4
SETB LED5
SETB LED6
SETB LED7
SETB LED8

MOV DMTG, #0
MOV CHEDODEN, #0

MOV THOIGIANXANH, #8
MOV THOIGIANVANG, #3
MOV THOIGIANDO, #10

MOV KETQUAPHIM, #16

MOV DENHIENTAI, #0
MOV DEMLED1, THOIGIANXANH
MOV A, DEMLED1
ADD A, THOIGIANVANG
MOV DEMLED2, A

SETB LEDG1
SETB LEDR2

LAP:
CALL QUETPHIM
MOV R3, KETQUAPHIM
CJNE R3, #16, KIEMTRAPHIM
JMP LAP

KIEMTRAPHIM:
CLR TR0

```

```

MOV R3, KETQUAPHIM
CJNE R3, #10, KT2

MOV CHEDODEN, #1
SETB LEDG1
CLR LEDY1
CLR LEDR1
SETB LEDG2
CLR LEDY2
CLR LEDR2

MOV DEMLED1, #0
MOV DEMLED2, #0

JMP THOATKT

KT2:
MOV R3, KETQUAPHIM
CJNE R3, #11, KT3
MOV CHEDODEN, #2
CLR LEDG1
SETB LEDY1
CLR LEDR1
CLR LEDG2
SETB LEDY2
CLR LEDR2

MOV DEMLED1, #0
MOV DEMLED2, #0

JMP THOATKT

KT3:
MOV R3, KETQUAPHIM
CJNE R3, #12, KT4
MOV CHEDODEN, #3
CLR LEDG1
CLR LEDY1
SETB LEDR1
CLR LEDG2
CLR LEDY2
SETB LEDR2

MOV DEMLED1, #0
MOV DEMLED2, #0
JMP THOATKT

KT4:
MOV R3, KETQUAPHIM
CJNE R3, #13, KTSO
MOV R3, CHEDODEN

```

```
CJNE R3, #1, CAITGVANG
MOV THOIGIANXANH, DEMLED1
JMP BATDAUCHAY
```

```
CAITGVANG:
CJNE R3, #2, CAITGDO
MOV THOIGIANVANG, DEMLED1
JMP BATDAUCHAY
```

```
CAITGDO:
CJNE R3, #3, BATDAUCHAY
MOV THOIGIANDO, DEMLED1
JMP BATDAUCHAY
```

```
BATDAUCHAY:
MOV CHEDODEN, #0
MOV DENHIENTAI, #0
```

```
MOV DEMLED1, THOIGIANXANH
MOV A, DEMLED1
ADD A, THOIGIANVANG
MOV DEMLED2, A
```

```
MOV P3, #0
SETB LEDG1
SETB LEDR2
SETB TR0
```

```
JMP THOATKT
```

```
KTSO:
```

```
CLR TR1
MOV A, DEMLED1
MOV B, #10
MUL AB
ADD A, KETQUAPHIM
MOV DEMLED1, A
SETB TR1
```

```
THOATKT:
MOV KETQUAPHIM, #16
```

```
JMP LAP
```

```
KHOITAOTIMER:
    MOV TMOD, #11h
    CLR TF1
    SETB ET1
    MOV TH1, #0BFH
    MOV TL1, #0
```



```

        CLR TF0
        SETB ET0
        MOV TH0, #3CH
        MOV TL0, #0B0H
        SETB TR1
        SETB TR0
        SETB EA
RET

QUETPHIM:

MOV R3, KETQUAPHIM
CJNE R3, #16, NHAYTHOATQUETPHIM

CLR HANG1
JNB COT1, PHIM11
JNB COT2, PHIM12
JNB COT3, PHIM13
JNB COT4, PHIM14
JMP CHECKHANG2

PHIM11:
JNB COT1, $
MOV KETQUAPHIM, #1
JMP THOATQUETPHIM

PHIM12:
JNB COT2, $
MOV KETQUAPHIM, #2
JMP THOATQUETPHIM

PHIM13:
JNB COT3, $
MOV KETQUAPHIM, #3
JMP THOATQUETPHIM

PHIM14:
JNB COT4, $
MOV KETQUAPHIM, #10
JMP THOATQUETPHIM

CHECKHANG2:
SETB HANG1
CLR HANG2
JNB COT1, PHIM21
JNB COT2, PHIM22
JNB COT3, PHIM23
JNB COT4, PHIM24
JMP CHECKHANG3

```

```

PHIM21:
JNB COT1, $
MOV KETQUAPHIM, #4
JMP THOATQUETPHIM

NHAYTHOATQUETPHIM:
JMP THOATQUETPHIM

PHIM22:
JNB COT2, $
MOV KETQUAPHIM, #5
JMP THOATQUETPHIM

PHIM23:
JNB COT3, $
MOV KETQUAPHIM, #6
JMP THOATQUETPHIM

PHIM24:
JNB COT4, $
MOV KETQUAPHIM, #11
JMP THOATQUETPHIM

CHECKHANG3:
SETB HANG2
CLR HANG3
JNB COT1, PHIM31
JNB COT2, PHIM32
JNB COT3, PHIM33
JNB COT4, PHIM34
JMP CHECKHANG4

PHIM31:
JNB COT1, $
MOV KETQUAPHIM, #7
JMP THOATQUETPHIM

PHIM32:
JNB COT2, $
MOV KETQUAPHIM, #8
JMP THOATQUETPHIM

PHIM33:
JNB COT3, $
MOV KETQUAPHIM, #9
JMP THOATQUETPHIM

PHIM34:
JNB COT4, $
MOV KETQUAPHIM, #12
JMP THOATQUETPHIM

```

```

CHECKHANG4:
SETB HANG3
CLR HANG4
JNB COT1, PHIM41
JNB COT2, PHIM42
JNB COT3, PHIM43
JNB COT4, PHIM44
JMP THOATQUETPHIM

PHIM41:
MOV KETQUAPHIM, #0
JNB COT1, $
JMP THOATQUETPHIM

PHIM42:
MOV KETQUAPHIM, #0
JNB COT2, $
JMP THOATQUETPHIM

PHIM43:
MOV KETQUAPHIM, #0
JNB COT3, $
JMP THOATQUETPHIM

PHIM44:
MOV KETQUAPHIM, #13
JNB COT4, $

THOATQUETPHIM:
MOV P0, #0FFH
RET

QUETLED:
    CLR TR1
    CLR TF1
    MOV TH1, #0BFH
    MOV TL1, #0

    MOV A, DEMLED1
    MOV B, #10
    DIV AB
    MOV CHUC, A
    MOV DONVI, B
    MOV DPTR, #MALED

    MOV A, CHUC
    MOVC A, @A+DPTR
    MOV LEDDATA, A
    CLR LED1

```

```

CALL DELAY
SETB LED1
CLR LED2
CALL DELAY
SETB LED2

MOV A, DONVI
MOVC A, @A+DPTR
MOV LEDDATA, A
CLR LED3
CALL DELAY
SETB LED3
CLR LED4
CALL DELAY
SETB LED4

MOV A, DEMLED2
MOV B, #10
DIV AB
MOV CHUC, A
MOV DONVI, B
MOV DPTR, #MALED

MOV A, CHUC
MOVC A, @A+DPTR
MOV LEDDATA, A
CLR LED5
CALL DELAY
SETB LED5
CLR LED6
CALL DELAY
SETB LED6

MOV A, DONVI
MOVC A, @A+DPTR
MOV LEDDATA, A
CLR LED7
CALL DELAY
SETB LED7
CLR LED8
CALL DELAY
SETB LED8

SETB TR1
JMP EXITQUET
TATLED7DOAN:
SETB TR1
MOV LEDDATA, #0FFH
EXITQUET:
RET

```

DEMTHOIGIAN:

CLR TF0

MOV TH0, #3CH

MOV TL0, #0B0H

MOV R4, CHEDODEN

CJNE R4, #0, EXITDEMTG

INC DEMTG

CJNE DEMTG, #20, EXITDEMTG

MOV DEMTG, #0

DEC DEMLED1

DEC DEMLED2

CJNE DEMLED1, #0, CHECKLED2

CJNE DENHIENTAI, #0, CHECKDENVANG

MOV DENHIENTAI, #1

MOV DEMLED1, THOIGIANVANG

CLR LEDG1

SETB LEDY1

JMP EXITDEMTG

CHECKDENVANG:

CJNE DENHIENTAI, #1, CHECKDENDO

MOV DENHIENTAI, #2

MOV DEMLED1, THOIGIANDO

MOV A, DEMLED1

SUBB A, #3

MOV DEMLED2, A

CLR LEDY1

SETB LEDR1

CLR LEDR2

SETB LEDG2

JMP EXITDEMTG

CHECKDENDO:

MOV DENHIENTAI, #0

MOV DEMLED1, THOIGIANXANH

MOV A, DEMLED1

ADD A, #3

MOV DEMLED2, A

CLR LEDY2

SETB LEDR2

CLR LEDR1

SETB LEDG1

JMP EXITDEMTG

CHECKLED2:

CJNE DEMLED2, #0, EXITDEMTG

MOV DEMLED2, THOIGIANVANG

CLR LEDG2

```

    SETB LEDY2
    JMP EXITDEMTG
EXITDEMTG:

RET

TATLED:
    CLR LEDG1
    CLR LEDY1
    CLR LEDR1
    CLR LEDG2
    CLR LEDY2
    CLR LEDR2

RET

DELAY1MS:
    MOV R4, #200
WAIT21: MOV R2, #250
WAIT22: MOV R3, #250
        DJNZ R3, $
        DJNZ R2, WAIT22
        DJNZ R4, WAIT21

RET

DELAY: MOV R5, #5
WAIT2: MOV R6, #100
        DJNZ R6, $
        DJNZ R5, WAIT2

RET

MALED: DB 0C0H, 0F9H, 0A4H, 0B0H, 099H, 092H, 082H, 0F8H, 080H, 090H, 0BFH
END

```

3.5. Nguyên lý hoạt động

Các mạch hiển thị được vận hành dựa trên việc điều khiển các bit vào/ra của các cổng trên IC AT89S52. Vi điều khiển được lập trình để điều khiển luồng công việc sau:

- Chuyển mức tín hiệu của 4 cột đèn giao thông đồng thời hiển thị số giây đếm ngược của mỗi đèn tại ngã tư. Việc hiển thị đèn sẽ thông qua 6 bit truyền tín hiệu, trong đó các cột đèn đối diện nhau sẽ dùng chung 1 cổng.

Làn đường thứ nhất	Làn đường thứ hai
P3.0 Đèn đỏ	P3.3 Đèn đỏ
P3.1 Đèn vàng	P3.4 Đèn vàng
P3.2 Đèn xanh	P3.5 Đèn xanh

- Đèn hiển thị đều là LED đôi nối chung anode. Đèn sáng tương ứng đang ở trạng thái mức logic thấp. Hai đèn đối diện nhau sẽ hiển thị giống nhau vì cùng làn đường. Khi đèn đỏ ở làn thứ nhất thì sẽ là làn xanh và vàng ở làn thứ 2. Việc chuyển tín hiệu đèn được hiển thị cùng với những bộ đếm hiển thị số giây đếm ngược, kết thúc mỗi bộ đếm sẽ chuyển trạng thái sang tín hiệu đèn tiếp theo và thực hiện bộ đếm mới. Bộ đếm được hiển thị trên LED đôi 7 đoạn.
- Trong thiết lập cài đặt ban đầu của chương trình, thời gian hiển thị đèn xanh là 8s, đèn vàng 3s, đèn đỏ 11s. Bên cạnh việc thiết lập sẵn, người dùng có thể cài đặt theo ý muốn thời gian hiển thị của từng đèn dựa trên phím nhấn keypad 4x4. Việc cài đặt được thực hiện bằng cách chọn đèn và nhấn số giây mình cần cho mỗi đèn. Quy ước, người dùng có thể tự chỉnh số giây của 2 đèn và thời gian đèn còn lại sẽ được chương trình tính toán.

Cụ thể:

- Nhấn phím A để cài đặt thời gian cho đèn xanh.
- Nhấn phím B để cài đặt thời gian cho đèn vàng.
- Nhấn phím C để cài đặt thời gian cho đèn đỏ.

Sau khi nhấn số giây của từng đèn, nhấn phím D để hoàn thành việc xác nhận cài đặt số giây. Sau khi cài đặt xong, các đèn và LED hiển thị sẽ hiển thị theo số giây được cài đặt.

Đối với nút nhấn reset, khi người dùng muốn quay trở về trạng thái cài đặt thời gian ban đầu được thiết lập, sử dụng nút reset để thực hiện.

Chương 4. TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

4.1. Tổng kết

4.1.1. Kết quả

Link github: [gundamdouble00/DoAn_VXL_VDK \(github.com\)](https://github.com/gundamdouble00/DoAn_VXL_VDK)

Link video: [Đèn giao thông.mp4 - Google Drive](#)

4.1.2. Đánh giá

- ✚ Mạch hoạt động đúng theo yêu cầu đề bài.
- ✚ Đèn hiển thị trực quan sinh động, tính hiệu đèn chuyển đổi mượt.
- ✚ Mạch chạy tương đối ổn định. Tuy nhiên do có chút trục trặc về việc đi dây trong PCB layout nên LED hiển thị của làn thứ nhất và làn thứ hai bị ngược nhau.
- ✚ Khá nhiều khó khăn trong việc thiết kế mạch in, quá trình kết nối dây linh kiện và chi phí in mạch, mua linh kiện khá cao.

4.2. Hướng phát triển

Ngoài việc hiển thị như đèn giao thông tại các ngã tư, Hệ thống đèn giao thông tại ngã tư có thể phát triển trở thành hệ thống đèn giao thông tại ngã năm, đèn giao thông tại các vòng xoay lớn. Bên cạnh việc sử dụng keypad để cài đặt thời gian, có thể phát triển theo hướng cài đặt thời gian từ xa thông qua thiết bị có khả năng kết nối không dây để xử lý tín hiệu từ xa truyền vào chip để xử lý.

TÀI LIỆU THAM KHẢO

- [1] <https://www.elprocus.com/8051-assembly-language-programming/>
- [2] Giáo trình Vi điều khiển – Trường Đại học Công nghệ Thông tin
- [3] <https://www.alldatasheet.com/>