

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



BÁO CÁO ĐỒ ÁN
VI XỬ LÝ-VI ĐIỀU KHIỂN

ĐỀ TÀI

MẠCH ĐIỀU KHIỂN ĐÈN GIAO THÔNG

Giảng viên hướng dẫn: ThS. Trần Hoàng Lộc

Sinh viên thực hiện:

Hồ Văn Việt Hoàng	22520461
Nguyễn Phúc Hiếu	22520441
Trần Minh Huy	22520576
Nguyễn Việt Thịnh	21522633

TP. Hồ Chí Minh, tháng 06 năm 2024

LỜI NÓI ĐẦU

Trong thời đại công nghệ số hiện nay, vi xử lý và vi điều khiển đã trở thành những thành phần không thể thiếu trong các hệ thống điện tử và công nghiệp. Từ những ứng dụng nhỏ gọn trong các thiết bị gia dụng, hệ thống điều khiển công nghiệp đến các hệ thống phức tạp trong ô tô, hàng không vũ trụ và các thiết bị y tế, vi xử lý và vi điều khiển đã chứng minh được vai trò quan trọng của mình trong việc nâng cao hiệu suất và tính tự động hóa.

Sự phát triển của vi xử lý và vi điều khiển đã tạo ra những bước đột phá đáng kể, không chỉ trong công nghệ mà còn trong xã hội và cuộc sống hàng ngày. Với khả năng tính toán mạnh mẽ, tính linh hoạt cao và chi phí ngày càng giảm, vi xử lý và vi điều khiển đang dần thay đổi cách chúng ta tương tác với thế giới xung quanh. Từ việc điều khiển các thiết bị thông minh trong nhà, quản lý hệ thống giao thông đô thị đến các ứng dụng trong chăm sóc sức khỏe, những công nghệ này đang góp phần xây dựng một xã hội thông minh và bền vững hơn.

Để góp phần làm sáng tỏ những hiệu quả của những ứng dụng thực tế từ môn học vi xử lý-vi điều khiển, trong bài báo cáo này chúng em sẽ giới thiệu một mạch mô phỏng đèn giao thông - một ứng dụng điển hình của vi điều khiển trong quản lý giao thông đô thị. Mạch mô phỏng đèn giao thông không chỉ giúp hiểu rõ hơn về cách vi điều khiển hoạt động mà còn là cơ hội để áp dụng những kiến thức đã học vào thực tế, nhằm giải quyết các vấn đề giao thông một cách hiệu quả và an toàn hơn. Chúng ta sẽ cùng khám phá cấu trúc, nguyên lý hoạt động và lợi ích của mạch mô phỏng đèn giao thông, từ đó thấy được tầm quan trọng của vi xử lý và vi điều khiển trong việc cải thiện chất lượng cuộc sống và thúc đẩy sự phát triển công nghệ.

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy Trần Hoàng Lộc, giảng viên môn Vi xử lý - Vi điều khiển, vì những đóng góp to lớn và quý báu của thầy trong suốt quá trình học tập và thực hiện đồ án môn học. Sự tận tâm, nhiệt huyết trong giảng dạy của thầy đã mang đến cho chúng em những kiến thức chuyên sâu và toàn diện về vi xử lý và vi điều khiển, giúp chúng em không chỉ hiểu rõ các nguyên lý cơ bản mà còn áp dụng chúng vào thực tiễn một cách hiệu quả.

Thầy không chỉ là một người thầy với kiến thức rộng lớn mà còn là nguồn cảm hứng mạnh mẽ cho chúng em. Những bài giảng của thầy luôn được trình bày một cách sinh động, dễ hiểu và hấp dẫn, tạo điều kiện thuận lợi để chúng em tiếp thu kiến thức một cách sâu sắc. Thầy đã giúp chúng em nhìn thấy tầm quan trọng và ứng dụng rộng rãi của vi xử lý và vi điều khiển trong cuộc sống hiện đại, từ những thiết bị điện tử hàng ngày đến các hệ thống tự động hóa phức tạp.

Trong suốt quá trình thực hiện đồ án, chúng em đã nhận được sự hỗ trợ và hướng dẫn tận tình từ thầy. Thầy luôn sẵn sàng giải đáp mọi thắc mắc, giúp chúng em vượt qua những khó khăn và thách thức. Những góp ý và chỉ dẫn quý báu của thầy đã giúp chúng em hoàn thiện đồ án một cách tốt nhất, nâng cao kỹ năng thực hành và tư duy sáng tạo.

Ngoài ra, thầy còn truyền cho chúng em tinh thần học hỏi không ngừng, khuyến khích chúng em tìm tòi, nghiên cứu và phát triển thêm các ý tưởng mới. Chính nhờ sự khích lệ và động viên của thầy, chúng em đã tự tin hơn trong việc khám phá và khai thác tiềm năng của mình.

Một lần nữa, chúng em xin gửi lời cảm ơn sâu sắc đến thầy Trần Hoàng Lộc. Chúng em cảm thấy vô cùng may mắn và vinh dự khi được học tập dưới sự dẫn dắt của thầy. Chúng em sẽ luôn ghi nhớ những bài học quý báu và những kỷ niệm đẹp trong quá trình học tập môn Vi xử lý - Vi điều khiển. Chúng em xin chúc thầy sức khỏe, hạnh phúc và tiếp tục thành công trong sự nghiệp giảng dạy và nghiên cứu khoa học.

MỤC LỤC

CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI.....	6
1.1. Thành phần chính của hệ thống đèn giao thông dùng vi điều khiển 8051.....	6
1.2. Nguyên lý hoạt động	6
<i>Hình 1.1: Sơ đồ khối của hệ thống</i>	7
1.3. Phần mềm thực hiện lập trình và ngôn ngữ lập trình	7
1.3.1. Phần mềm	7
1.3.2. Ngôn ngữ lập trình.....	7
CHƯƠNG II: KHẢO SÁT VI ĐIỀU KHIỂN AT89C51	8
2.1. Giới thiệu	8
<i>Hình 2.1: AT89C51 của Microchip</i>	8
<i>Hình 2.2: Sơ đồ chân AT89C51</i>	9
2.2. Chi tiết chân của AT89C51	9
2.2.1. Vcc	10
2.2.2. GND.....	10
2.2.3. Các Port	10
<i>Bảng 2.1: Chức năng của các chân trong Port 3</i>	11
2.2.4. PSEN	11
2.2.5. ALE/PROG.....	11
2.2.6. EA/Vpp.....	11
2.2.7. RESETS (RST).....	12
2.2.8. XTALI & XTAL2.....	12
2.3. Tổ chức bộ nhớ.....	12
2.3.1. Vùng RAM đa mục đích.....	12
<i>Bảng 2.2: Cấu trúc bộ nhớ RAM bên trong AT89C51</i>	13
2.3.2. Vùng RAM định địa chỉ bit.....	13
2.3.3. Các dây thanh ghi	13
2.4. Bộ nhớ ngoài.....	14
2.4.1. Truy xuất bộ nhớ chương trình ngoài.....	14
2.4.2. Truy xuất bộ nhớ dữ liệu ngoài	14
2.4.3. Giải mã địa chỉ	14
2.5. Giao tiếp ngoại vi	14
2.5.1. Led 7 thanh.....	14

<i>Hình 2.3: Dạng led 7 thanh anode chung</i>	14
<i>Bảng 2.3: Bảng mã cho led 7 thanh Anode chung</i>	15
2.6. Thanh ghi điều khiển bộ định thời/bo đếm TCON (Timer/Counter Control Register).....	15
2.7. Thanh ghi TMOD (Timer/Counter Mode Control Register).....	16
2.8. Hệ các thanh ghi phục vụ định thời.....	17
<i>Bảng 2.4: Tóm tắt thanh ghi IP</i>	19
<i>Bảng 2.5: Tóm tắt thanh ghi IE</i>	20
CHƯƠNG 3: PHÂN TÍCH MẠCH ĐIỀU KHIỂN HỆ THỐNG ĐÈN GIAO THÔNG	20
3.1. Sơ đồ khối	20
<i>Hình 3.1: Sơ đồ khối</i>	20
3.2. Thiết kế phần cứng	21
3.2.1. Mạch dao động và Reset.....	21
<i>Hình 3.2: Sơ đồ nguyên lý của mạch dao động và Reset</i>	21
<i>Bảng 3.1: Trạng thái của các thanh ghi khi Reset</i>	22
3.2.2. Mạch hiển thị đếm ngược LED 7 đoạn	22
3.2.3. Mạch hiển thị LED đơn	24
3.2.4. Keypad	25
<i>Hình 3.3: Keypad và sơ đồ nguyên lý hoạt động</i>	25
- Nguyên lý hoạt động của Keypad.....	25
1. Cấu trúc của Keypad:.....	25
3.2.5. Sơ đồ nguyên lý của hệ thống.....	27
<i>Hình 3.4: Sơ đồ nguyên lý của hệ thống</i>	27
3.2.6. Sơ đồ thiết kế	27
<i>Hình 3.5: Sơ đồ thiết kế của mạch</i>	27
3.2.7. Sơ đồ PCB.....	28
<i>Hình 3.6: Sơ đồ PCB của mạch</i>	28
3.3. Xây dựng, thiết kế chương trình.....	28
3.3.1. Lưu đồ thuật toán	29
3.3.2. Chương trình Assembly cho 89C51.....	30
CHƯƠNG 4: KẾT LUẬN	35
TÀI LIỆU THAM KHẢO	37

CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI

Ngày nay cùng với sự phát triển của khoa học kỹ thuật, xã hội ngày càng văn minh và hiện đại, các đô thị ngày một đi lên. Nhu cầu về giao thông ngày càng trở nên cấp thiết, nhất là trong các khu vực thành thị. Do nhu cầu của đời sống con người, đặc biệt là nhu cầu đi lại, các loại phương tiện giao thông đã tăng một cách chóng mặt. Riêng tại Việt Nam, số lượng xe máy trong những năm qua tăng một cách đột biến, mật độ xe lưu thông trên đường ngày một nhiều, trong khi đó hệ thống đường xá tại Việt Nam còn quá nhiều hạn chế nên thường gây ra các hiện tượng như kẹt xe, ách tắc giao thông, đặc biệt là tai nạn giao thông ngày càng phổ biến trở thành mối hiểm họa cho nhiều người.

Vì lý do đó các luật giao thông lần lượt ra đời và được đưa vào sử dụng một cách lặt vặt rồi dần trở nên phổ biến như hiện nay. Trong đó hệ thống đèn giao thông là công cụ điều khiển giao thông công cộng thực tế và hiệu quả có vai trò rất lớn trong việc đảm bảo an toàn và giảm thiểu tai nạn giao thông. Từ thực tế đó nhóm chúng em thực hiện đề tài “Thiết kế mạch đèn giao thông dùng vi điều khiển 8051” nhằm giúp cho mọi người nhất là tầng lớp sinh viên có ý thức hơn trong việc chấp hành luật lệ an toàn giao thông.

Với yêu cầu khi hệ thống chạy, 3 đèn LED xanh, vàng và đỏ sáng trong thời gian mặc định được hiển thị ra các thanh LED 7 đoạn. Ngoài ra giá trị hiển thị có thể được nhập vào bằng tay và điều chỉnh bằng keypad.

1.1. Thành phần chính của hệ thống đèn giao thông dùng vi điều khiển 8051

Hệ thống đèn giao thông gồm 4 phần chủ yếu sau:

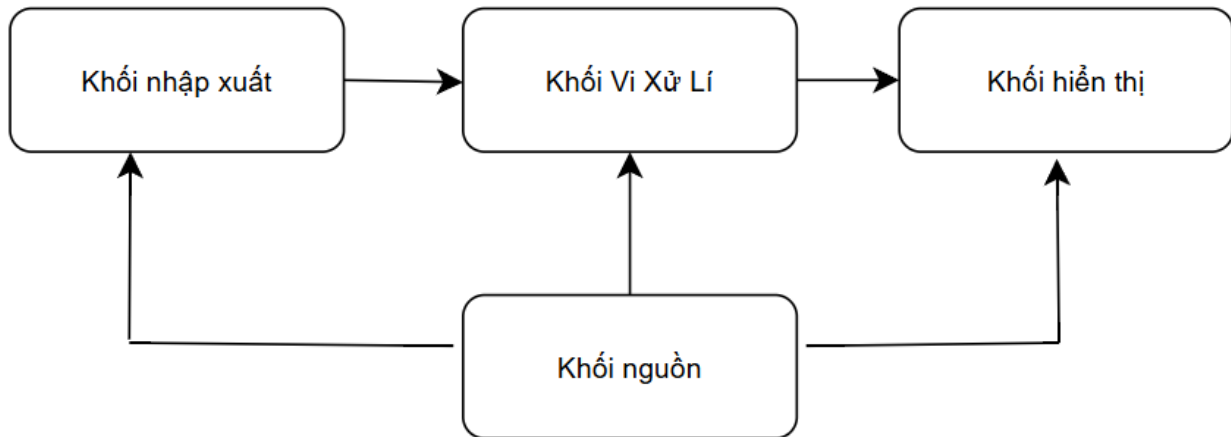
- Mạch điều khiển trung tâm dùng vi điều khiển AT89C51.
- Mạch dao động, reset.
- Mạch hiển thị thời gian, trạng thái đèn.
- Mạch nhập giá trị.

1.2. Nguyên lý hoạt động

Mạch đèn giao thông hoạt động dựa trên nội dung đã lập trình cho AT89C51, khi có tác động từ các giá trị nhập vào từ các nút của Keypad, giá trị mới sẽ được cập nhật với thời gian yêu cầu. AT89C51 đưa dữ liệu đến các LED xanh, đỏ, vàng để điều khiển các LED này đóng, mở. Ngoài ra, nó còn xuất dữ liệu đến các BIT để điều khiển các LED 7 đoạn. LED 7 đoạn còn nhận dữ liệu từ vi điều khiển trung tâm để thực hiện

việc đếm lùi thời gian.

Như vậy mỗi khi mạch bắt đầu thực hiện đếm lùi, nếu trục lộ bên này đèn xanh hoặc vàng sáng thì trục lộ bên kia đèn đỏ sáng và ngược lại. AT89C51 sẽ xuất ra các cổng I/O những xung ở mức cao hoặc mức thấp để điều khiển các BJT từ đó điều khiển các đèn hiển thị.



Hình 1.1: Sơ đồ khối của hệ thống

1.3. Phần mềm thực hiện lập trình và ngôn ngữ lập trình

1.3.1. Phần mềm

- Phần mềm Proteus 8.16, EasyDEA

1.3.2. Ngôn ngữ lập trình

- Sử dụng ngôn ngữ lập trình Assembly

CHƯƠNG II: KHẢO SÁT VI ĐIỀU KHIỂN AT89C51

2.1. Giới thiệu



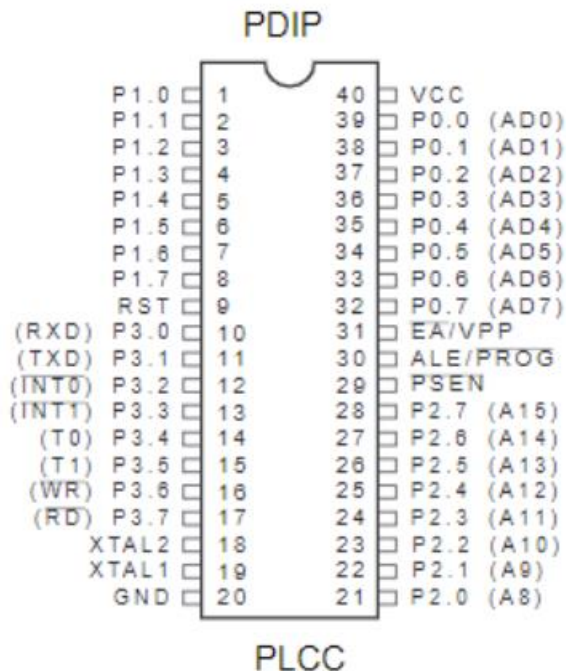
Hình 2.1: AT89C51 của Microchip

AT89C51 là một hệ vi tính 8-bit đơn chip CMOS có hiệu suất cao, công suất nguồn tiêu thụ thấp và có 4KB bộ nhớ ROM Flash xóa được/lập trình được. Chip này được sản xuất dựa vào công nghệ bộ nhớ không mất nội dung có độ tích hợp cao của Atmel.

Chip AT89C51 cũng tương thích với tập lệnh và các chân ra của chuẩn công nghiệp MCS-51. Flash trên chip này cho phép bộ nhớ chương trình được lập trình lại trên hệ thống. Kết hợp một CPU linh hoạt 8 bit với Flash trên một chip đơn thể, Atmel 89C51 là một hệ vi tính 8 bit đơn chip mạnh cho ta một giải pháp có hiệu quả về chi phí và rất linh hoạt đối với các ứng dụng điều khiển.

AT89C51 có các đặc trưng chuẩn sau: 4KN Flash, 128 byte RAM, 32 đường xuất nhập, 2 bộ định thời/đếm 16 bit, một cấu trúc ngắt hai mức ưu tiên và 5 nguyên nhân ngắt, một port nối tiếp song công, mạch dao động và tạo xung clock trên chip. Ngoài ra AT89C51 được thiết kế với logic tĩnh cho hoạt động có tần số giảm xuống 0 và hỗ trợ hai chế độ tiết kiệm năng lượng được lựa chọn bằng phần mềm. Chế độ nghỉ dừng CPU trong khi vẫn cho phép RAM, các bộ định thời/đếm, port nối tiếp và hệ thống ngắt tiếp tục hoạt động.

Cấu hình chân của AT89C51:



Hình 2.2: Sơ đồ chân AT89C51

AT89C51 có tất cả 40 chân. Mỗi chân có chức năng như các đường I/O (xuất/nhập), trong đó 24 chân có công dụng kép: mỗi đường có thể hoạt động như một đường I/O hoặc như một đường điều khiển hoặc như thành phần của bus địa chỉ và bus dữ liệu.

2.2. Chi tiết chân của AT89C51

Như ta thấy trong hình 2.2, 32 trong số 40 chân của 89C51 có công dụng xuất/nhập, tuy nhiên 24 trong 32 đường này có 2 mục đích (công dụng). Mỗi một đường có thể hoạt động xuất/nhập hoặc hoạt động như một đường điều khiển hoặc hoạt động như một đường địa chỉ/dữ liệu của bus địa chỉ/dữ liệu đa hợp.

32 chân nêu trên hình thành 4 port 8 bit. Với các thiết kế yêu cầu một mức tối thiểu bộ nhớ ngoài hoặc các thành phần bên ngoài khác, ta có thể sử dụng các port này làm nhiệm vụ xuất/nhập. 8 đường cho mỗi port có thể được xử lý như một đơn vị giao tiếp với các thiết bị song song như máy in, bộ biến đổi D-A, v.v ... hoặc mỗi đường có thể hoạt động độc lập giao tiếp với một thiết bị đơn bit như chuyển mạch, LED, BJT, động cơ, loa, v.v ...

2.2.1. Vcc

Chân cung cấp điện (5V).

2.2.2. GND

Chân nối đất (0V).

2.2.3. Các Port

+ **Port 0:**

Port 0 (các chân từ 32 đến 39 trên 89C51) có hai công dụng. Trong các thiết kế có tối thiểu thành phần, port 0 được sử dụng làm nhiệm vụ xuất/nhập. Trong các thiết kế lớn hơn có bộ nhớ ngoài, port 0 trở thành bus địa chỉ và bus dữ liệu đa hợp.

+ **Port 1:**

Port 1 chỉ có công dụng là xuất/nhập (các chân từ 1 đến 8 trên 89C51). Các chân của port 1 được ký hiệu là P1.0, P1.1, ... , P1.7 và được dùng để giao tiếp với thiết bị bên ngoài khi có yêu cầu. Không có chức năng nào khác nữa gán cho các chân của port 1, nghĩa là chúng chỉ được sử dụng để giao tiếp với các thiết bị ngoại vi.

+ **Port 2:**

Port 2 (các chân từ 21 đến 28 trên 89C51) có hai công dụng, hoặc làm nhiệm vụ xuất/nhập hoặc là byte địa chỉ 16-bit cho các thiết kế có bộ nhớ chương trình ngoài hoặc các thiết kế có nhiều hơn 256 byte bộ nhớ dữ liệu ngoài.

+ **Port 3:**

Port 3 (các chân từ 10 đến 17 trên 89C51) có hai công dụng. Khi không hoạt động xuất/nhập, các chân của port 3 có nhiều chức năng riêng (mỗi chân có chức năng riêng liên quan đến các đặc trưng cụ thể của 89C51).

Chân của Port 3	Chức năng
P3.0	RxD(ngõ vào của port nối tiếp)
P3.1	TxD(ngõ ra của port nối tiếp)
P3.2	INT0(ngõ vào ngắt ngoài 0)
P3.3	INT(ngõ vào ngắt ngoài 1)
P3.4	T0(ngõ vào bên ngoài của bộ định thời 0)
P3.5	T1(ngõ vào bên ngoài của bộ định thời 1)
P3.6	WR(điều khiển ghi bộ nhớ dữ liệu ngoài)
P3.7	RD(điều khiển đọc bộ nhớ dữ liệu ngoài)

Bảng 2.1: Chức năng của các chân trong Port 3

2.2.4. PSEN

Chân cho phép bộ nhớ chương trình PSEN (program store enable) điều khiển truy xuất bộ nhớ chương trình ngoài. Khi AT89C51 đang thực thi chương trình trong bộ nhớ chương trình ngoài, PSEN tích cực hai lần cho mỗi chu kỳ máy, ngoại trừ trường hợp 2 tác động của PSEN bị bỏ qua cho mỗi lần truy xuất bộ nhớ dữ liệu ngoài.

2.2.5. ALE/PROG

Xung của ngõ ra cho phép chốt địa chỉ ALE (address latch enable) cho phép chốt byte thấp của địa chỉ trong thời gian truy suất bộ nhớ ngoài. Chân này cũng được dùng làm ngõ vào xung lập trình (PROG) trong thời gian lập trình cho Flash. Khi hoạt động bình thường, xung của ngõ ra ALE luôn luôn có tần số bằng 1/6 tần số của mạch dao động trên chip, có thể được sử dụng cho các mục đích định thời từ bên ngoài và tạo xung clock. Tuy nhiên cần lưu ý là một xung ALE sẽ bị bỏ qua trong mỗi một chu kỳ truy xuất bộ nhớ dữ liệu ngoài.

2.2.6. EA/Vpp

Chân cho phép truy xuất bộ nhớ ngoài EA (external access enable) phải được nối với GND để cho phép chip vi điều khiển tìm nạp lệnh từ các vị trí nhớ của bộ nhớ chương trình ngoài, bắt đầu từ địa chỉ 0000H cho đến FFFFH. Tuy nhiên cần lưu ý là nếu bit khóa 1 (lock bit 1) được lập trình, EA sẽ được

chốt bên trong khi reset. EA nên nối với Vcc để thực thi chương trình bên trong chip. Chân EA/Vpp còn nhận điện áp cho phép lập trình Vpp trong thời gian lập trình cho Flash, điện áp này cấp cho các bộ phận có yêu cầu điện áp 12V.

2.2.7. RESETS (RST)

Ngõ vào RST (chân 9). Mức cao trên chân này trong 2 chu kỳ máy trong khi bộ dao động đang hoạt động sẽ reset AT89C51.

2.2.8. XTAL1 & XTAL2

XTAL1 ngõ vào đến mạch khuếch đại đảo của mạch dao động và ngõ vào đến mạch tạo xung clock bên trong chip. XTAL2 ngõ ra từ mạch khuếch đại đảo của mạch dao động.

2.3. Tổ chức bộ nhớ

AT89C51 có không gian bộ nhớ riêng cho chương trình và dữ liệu. Cả hai bộ nhớ chương trình và dữ liệu đều đặt bên trong chip, tuy nhiên ta có thể mở rộng bộ nhớ chương trình và bộ nhớ dữ liệu bằng cách sử dụng các chip nhớ bên ngoài với dung lượng tối đa là 64KB cho bộ nhớ chương trình (hay bộ nhớ mã) và 64KB cho bộ nhớ dữ liệu.

Bộ nhớ nội trong chip bao gồm ROM và RAM. RAM trên chip bao gồm vùng RAM đa chức năng (nhiều công dụng), vùng RAM với từng bit được định địa chỉ (gọi tắt là vùng RAM định địa chỉ bit), các dãy (bank) thanh ghi và các thanh ghi chức năng đặc biệt SFR (Special Function Register).

2.3.1. Vùng RAM đa mục đích

Hình 2.3 trình bày vùng RAM đa mục đích có 80-byte đặt ở địa chỉ từ 30H đến 7FH, bên dưới vùng này từ địa chỉ 00H đến 2FH là vùng nhớ có thể được sử dụng tương tự (mặc dù các vị trí nhớ này có các mục đích khác).

Địa chỉ byte	Địa chỉ bit								Địa chỉ byte	Địa chỉ bit								Ký hiệu (tên)	
7F	RAM đa dụng								FF										
									F0	F7	F6	F5	F4	F3	F2	F1	F0	B	
									E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC	
									D0	D7	D6	D5	D4	D3	D2	-	D0	PSW	
30									B8	-	-	-	BC	BB	BA	B9	B8	IP	
2F	7F	7E	7D	7C	7B	7A	79	78		B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
2E	77	76	75	74	73	72	71	70		A8	AF	-	-	AC	AB	AA	A9	A8	IE
2D	6F	6E	6D	6C	6B	6A	69	68		A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
2C	67	66	65	64	63	62	61	60		99	Không được địa chỉ hóa bit								SBUF
2B	5F	5E	5D	5C	5B	5A	59	58		98	9F	9E	9D	9C	9B	9A	99	98	SCON
2A	57	56	55	54	53	52	51	50		90	97	96	95	94	93	92	91	90	P1
29	4F	4E	4D	4C	4B	4A	49	48		8D	Không được địa chỉ hóa bit								TH1
28	47	46	45	44	43	42	41	40		8C	Không được địa chỉ hóa bit								TH0
27	3F	3E	3D	3C	3B	3A	39	38		8B	Không được địa chỉ hóa bit								TL1
26	37	36	35	34	33	32	31	30		8A	Không được địa chỉ hóa bit								TL0
25	2F	2E	2D	2C	2B	2A	29	28		89	Không được địa chỉ hóa bit								TMOD
24	27	26	25	24	23	22	21	20		88	8F	8E	8D	8C	8B	8A	89	88	TCON
23	1F	1E	1D	1C	1B	1A	19	18		87	Không được địa chỉ hóa bit								PCON
22	17	16	15	14	13	12	11	10		83	Không được địa chỉ hóa bit								DPH
21	0F	0E	0D	0C	0B	0A	09	08		82	Không được địa chỉ hóa bit								DPL
20	07	06	05	04	03	02	01	00		81	Không được địa chỉ hóa bit								SP
1F	Bank 3								80	87	86	85	84	83	82	81	80	P0	
18									Bank 2										
17	Bank 1																		
10									Bank thanh ghi mặc định cho R0 – R7 (Bank 0)										
0F	RAM																		
08																			
07																			
00																			

CÁC THANH GHI CHỨC NĂNG ĐẶC BIỆT																	
----------------------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Bảng 2.2: Cấu trúc bộ nhớ RAM bên trong AT89C51

2.3.2. Vùng RAM định địa chỉ bit

AT89C51 chứa 210 vị trí bit được định địa chỉ trong đó 128-bit chứa trong các byte ở địa chỉ từ 20H đến 2FH và phần còn lại chứa trong các thanh ghi chức năng đặc biệt.

2.3.3. Các dãy thanh ghi

32 vị trí thấp nhất của bộ nhớ nội chứa các dãy thanh ghi.

2.4. Bộ nhớ ngoài

Các bộ vi điều khiển cần có khả năng mở rộng các tài nguyên trên chip (bộ nhớ, I/O, v.v ...) để tránh hiện tượng cổ chai trong thiết kế. Cấu trúc của MCS-51 cho ta khả năng mở rộng không gian bộ nhớ chương trình đến 64K và không gian bộ nhớ dữ liệu đến 64K. ROM và RAM được thêm vào khi cần.

2.4.1. Truy xuất bộ nhớ chương trình ngoài

Bộ nhớ chương trình ngoài là bộ nhớ chỉ đọc, được cho phép bởi tín hiệu PSEN.

2.4.2. Truy xuất bộ nhớ dữ liệu ngoài

Bộ nhớ dữ liệu ngoài là bộ nhớ đọc/ghi được cho phép bởi các tín hiệu RD và WR ở các chân P3.7 và P3.6.

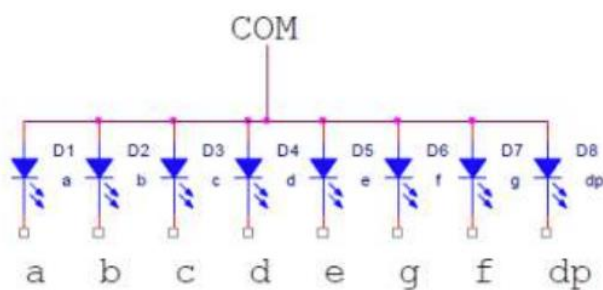
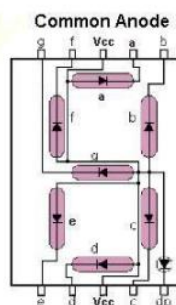
2.4.3. Giải mã địa chỉ

Nếu có nhiều EPROM hoặc nhiều RAM hoặc cả hai giao tiếp với 89C51 ta cần phải giải mã địa chỉ.

Một IC giải mã điển hình là 74HC138.

2.5. Giao tiếp ngoại vi

2.5.1. Led 7 thanh



Hình 2.3: Dạng led 7 thanh anode chung

Đối với dạng Led Anode chung, chân COM phải có mức logic 1 và muốn sáng Led thì tương ứng các chân a - f, dp sẽ ở mức logic 0. Bảng mã cho Led Anode chung (a là LSB, dp là MSB).

Số	dp	g	f	e	d	c	b	a	Mã hex
0	1	1	0	0	0	0	0	0	0C0H
1	1	1	1	1	1	0	0	1	0F9H
2	1	0	1	0	0	1	0	0	0A4H
3	1	0	1	1	0	0	0	0	0B0H
4	1	0	0	1	1	0	0	1	99H
5	1	0	0	1	0	0	1	0	92H
6	1	0	0	0	0	0	1	0	82H
7	1	1	1	1	1	0	0	0	0F8H
8	1	0	0	0	0	0	0	0	80H
9	1	0	0	1	0	0	0	0	90H

Bảng 2.3: Bảng mã cho led 7 thanh Anode chung

2.6. Thanh ghi điều khiển bộ định thời/bo đếm TCON (Timer/Counter Control Register)

Thanh ghi này bao gồm các bit trạng thái và các bit điều khiển bởi Timer 0, Timer 1.

Thanh ghi TCON có bit định vị.

Ký hiệu: TCON

Chức năng: điều khiển bộ định thời/bộ đếm.

Địa chỉ byte: 88H

Định địa chỉ bit: có.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
7	6	5	4	3	2	1	0

Định nghĩa bit:

Ký hiệu	Vị trí	Địa chỉ	Mô tả
TF1	TCON.7	8FH	Cờ tràn bộ định thời 1. Được Set bởi phần cứng khi bộ định thời/đếm bị tràn, được xóa bởi phần mềm hoặc phần cứng khi trình phục vụ ngắt được
TR1	TCON.6	8EH	Bit điều khiển bộ định thời 1 hoạt động. Được Set/Clear bởi phần mềm để điều khiển bộ định thời 11 hoạt động hoặc ngưng hoạt động.
TF0	TCON.5	8DH	Cờ tràn bộ định thời 0
TR0	TCON.4	8CH	Bit điều khiển bộ định thời 0 hoạt động.
IE1	TCON.3	8BH	Cờ ngắt ngoài 1 tác động cạnh. Được Set bởi phần cứng khi phát hiện có ngắt ngoài tác động cạnh, được xóa bởi phần cứng khi ngắt được xử lý.
IT1	TCON.2	8AH	Bit điều khiển chọn ngắt. Được Set/Clear để xác định ngắt ngoài thuộc loại tác động cạnh xuống hay tác động mức thấp
IE0	TCON.1	89H	Cờ ngắt ngoài 0 tác động cạnh.
IT0	TCON.0	88H	Bit điều khiển chọn loại ngắt.

2.7. Thanh ghi TMOD (Timer/Counter Mode Control Register)

Thanh ghi này gồm 2 nhóm 4 bit: 4-bit thấp đặt mode hoạt động cho Timer 0 và 4 bit cao đặt mode hoạt động cho Timer 1.

Ký hiệu: TMOD

Chức năng: điều khiển chọn chế độ định thời/bộ đếm.

Địa chỉ byte: 89H

Định địa chỉ bit: không.

Timer 1				Timer 0			
GATE	C/T	M1	M0	GATE	C/T	M1	M0
7	6	5	4	3	2	1	0

Định nghĩa bit:

Ký hiệu	Mô tả
GATE	Bit điều khiển cổng. Khi bit TRx trong TCON được Set bằng 1 và GATE = 1, bộ định thời/đếm chỉ hoạt động trong khi chân INTx ở mức cao(điều khiển cứng). Khi bit GATE = 0, bộ định thời/đếm chỉ hoạt động trong khi chân TRx ở mức cao(điều khiển mềm).
C/T	Bit chọn chức năng đếm hay định thời cho bộ định thời/đếm. Khi C/T = 0, bộ định thời/đếm hoạt động định thời(dùng xung Clock nhận từ ngõ vào Tx)
M1	Bit chọn chế độ
M0	Bit chọn chế độ

Hai bit M1 và M0 dùng để chọn mode Timer 0 và Timer 1:

M0	M1	Chế độ	Mô tả
0	0	0	Bộ định thời 13 bit
0	1	1	Bộ định thời/ đếm 16 bit
1	0	2	Bộ định thời/ đếm 8-bit tự động nạp lại
1	1	3	Bộ định thời 0: TL0 là bộ định thời/đếm 8-bit được điều khiển bởi các bit điều khiển bộ định thời 0. TH0 là bộ định thời 8-bit được điều khiển bởi các bit điều khiển bộ định thời 1. Bộ định thời 1: Bộ định thời/đếm 1 ngưng hoạt động.

2.8. Hệ các thanh ghi phục vụ định thời

Hệ thanh ghi này gồm 4 thanh ghi nó cho phép lập trình viên nạp giá trị cho bộ định thời. Thực tế chỉ 2 thanh ghi 16 bit nhưng nó được cắt đôi thành 4 thanh ghi 8 bit.

Trong đó 2 thanh phục vụ cho Timer 0 và 2 thanh phục vụ cho Timer 1.

Bộ Timer 0 có hai thanh ghi là TH0 và TL0, hai thanh ghi này không định địa chỉ bit:

Thanh ghi 8-bit TH0	Thanh ghi 8-bit TH0
<i>Địa chỉ: 8CH</i>	<i>Địa chỉ: 8AH</i>

Bộ Timer 1 có hai thanh ghi là TH1 và TL1, hai thanh ghi này không định địa chỉ bit:

Thanh ghi 8-bit TH1	Thanh ghi 8-bit TH1
Địa chỉ: 8CH	Địa chỉ: 8AH

Thanh ghi TIMER

Vi Điều Khiển 89C51 có 2 timer 16 bit, mỗi timer có bốn cách làm việc. Người ta sử dụng các timer để:

- Định khoảng thời gian.
- Đếm sự kiện.
- Tạo tốc độ baud cho port nối tiếp.

Trong các ứng dụng định khoảng thời gian, người ta lập trình timer ở những khoảng đều đặn và đặt cờ tràn timer. Cờ được dùng để đồng bộ hóa chương trình để thực hiện một tác động như kiểm tra trạng thái của các ngõ vào hoặc gửi sự kiện ra các ngõ ra. Các ứng dụng khác có thể sử dụng việc tạo xung nhịp đều đặn của timer để đo thời gian trôi qua giữa hai sự kiện (ví dụ đo độ rộng xung).

Thanh ghi ngắt (INTERRUPT)

Một ngắt là sự xảy ra một điều kiện, một sự kiện mà nó gây ra treo tạm thời chương trình chính trong khi điều kiện đó được phục vụ bởi một chương trình khác.

Các ngắt đóng một vai trò quan trọng trong thiết kế và cài đặt các ứng dụng vi điều khiển. Chúng cho phép hệ thống đáp ứng bất đồng bộ với một sự kiện và giải quyết sự kiện đó trong khi một chương trình khác đang thực thi.

- Tổ chức ngắt của 89C51:

- Có 5 nguồn ngắt ở 89C51: 2 ngắt ngoài, 2 ngắt từ timer và 1 ngắt port nối tiếp. Tất cả các ngắt theo mặc nhiên đều bị cấm sau khi reset hệ thống và được cho phép từng cái một bằng phần mềm. Mức độ ưu tiên của các ngắt được lưu trong thanh ghi IP (Interrupt Priority) hay nói cách khác thanh ghi IP cho phép chọn mức ưu tiên cho các ngắt (giá trị thanh ghi IP khi reset là 00h).

Bit	Ký hiệu	Địa chỉ bit	Mô tả
IP.7	-	-	Không được mô tả
IP.6	-	-	Không được mô tả
IP.5	ET2	BDH	Chọn mức ưu tiên cao (=1) hay thấp (=0) tại cổng timer 2
IP.4	ES	BCH	Chọn mức ưu tiên cao (=1) hay thấp (=0) tại cổng nối tiếp
IP.3	ET1	BBH	Chọn mức ưu tiên cao (=1) hay thấp (=0) tại timer 1
IP.2	EX1	BAH	Chọn mức ưu tiên cao(=1) hay thấp (=0) tại ngắt ngoài 1
IP.1	ET0	B9H	Chọn mức ưu tiên cao(=1) hay thấp (=0) tại timer 0
IP.0	EX0	B8H	Chọn mức ưu tiên cao(=1) hay thấp (=0) tại ngắt ngoài 0
Tóm tắt thanh ghi IP			

Bảng 2.4: Tóm tắt thanh ghi IP

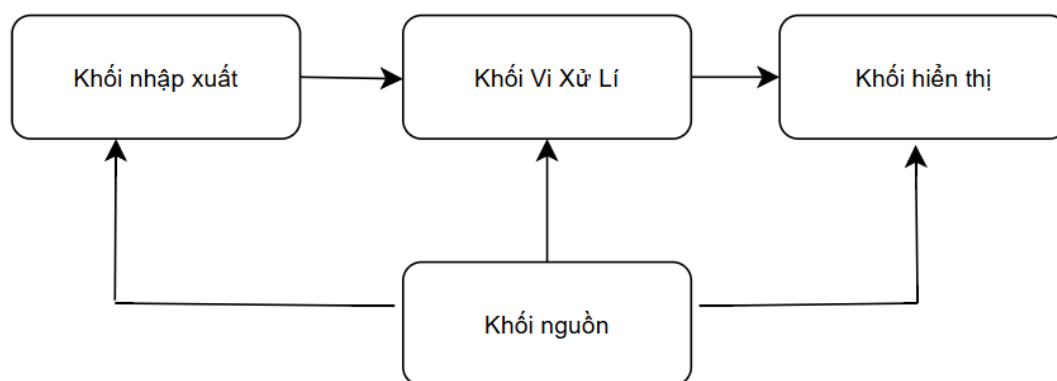
- Nếu 2 ngắt xảy ra đồng thời thì ngắt nào có mức ưu tiên cao hơn sẽ được phục vụ trước
 - Nếu 2 ngắt xảy ra đồng thời có cùng mức ưu tiên thì thứ tự ưu tiên được thực hiện từ cao đến thấp như sau: ngắt ngoài 0 - timer 0 - ngắt ngoài 1 - timer 1 - cổng nối tiếp timer 2
 - Nếu chương trình của một ngắt có mức ưu tiên thấp đang chạy mà có một ngắt xảy ra với mức ưu tiên cao hơn thì chương trình này tạm dừng để chạy một chương trình khác có mức ưu tiên cao hơn.
- **Cho phép và cấm ngắt:**
- Mỗi nguồn ngắt được cho phép hoặc cấm ngắt qua một thanh ghi chức năng đặt biệt có định địa chỉ bit IE (Interrupt Enable: cho phép ngắt) ở địa chỉ A8H.

Bit	Ký hiệu	Địa chỉ bit	Mô tả
IE.7	EA	AFH	Cho phép/ Cấm toàn bộ
IE.6	-	AEH	Không được mô tả
IE.5	ET2	ADH	Cho phép ngắt từ Timer 2(8052)
IE.4	ES	ACH	Cho phép ngắt port nối tiếp
IE.3	ET1	ABH	Cho phép ngắt từ Timer 1
IE.2	EX1	AAH	Cho phép ngắt ngoài 1
IE.1	ET0	A9H	Cho phép ngắt từ Timer 0
IE.0			

Bảng 2.5: Tóm tắt thanh ghi IE

CHƯƠNG 3: PHÂN TÍCH MẠCH ĐIỀU KHIỂN HỆ THỐNG ĐÈN GIAO THÔNG

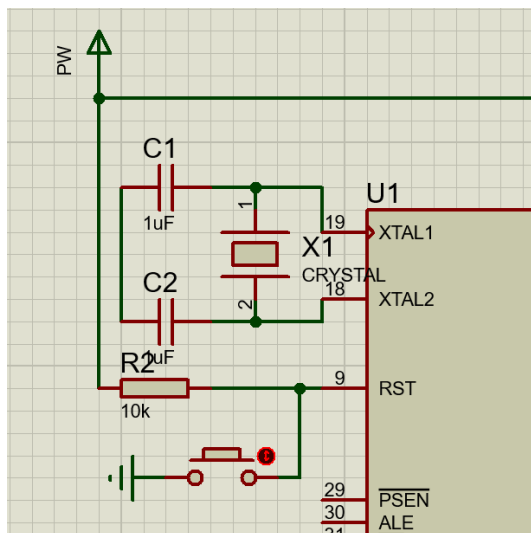
3.1. Sơ đồ khối



Hình 3.1: Sơ đồ khối

3.2. Thiết kế phần cứng

3.2.1. Mạch dao động và Reset



Hình 3.2: Sơ đồ nguyên lý của mạch dao động và Reset

- **Mạch dao động**

Tinh thể thạch anh (Quartz Crystal) là loại đá trong mờ trong thiên nhiên, chính là dioxyt silicium (SiO_2).

Tinh thể thạch anh dùng trong mạch dao động là một lát mỏng được cắt ra từ tinh thể. Tùy theo mặt cắt mà lát thạch anh có đặc tính khác nhau. Lát thạch anh có diện tích từ nhỏ hơn 1cm^2 đến vài cm^2 được mài rất mỏng, phẳng (vài vài mm) và 2 mặt thật song song với nhau. Hai mặt này được mạ kim loại và nổi chân ra ngoài để dễ sử dụng

Đặc tính của tinh thể thạch anh là tính áp điện (Piezoelectric Effect) theo đó khi ta áp một lực vào 2 mặt của lát thạch anh (nén hoặc kéo dãn) thì sẽ xuất hiện một điện thế xoay chiều giữa 2 mặt. Ngược lại dưới tác dụng của một điện thế xoay chiều, lát thạch anh sẽ rung ở một tần số không đổi và như vậy tạo ra một điện thế xoay chiều có tần số không đổi. Tần số rung động của lát thạch anh tùy thuộc vào kích thước của nó đặc biệt là độ dày mặt cắt. Khi nhiệt độ thay đổi, tần số rung động của thạch anh cũng thay đổi theo nhưng vẫn có độ ổn định tốt hơn rất nhiều so với các mạch dao động không dùng thạch anh (tần số dao động gần như chỉ tùy thuộc vào thạch anh mà không lệ thuộc mạch ngoài).

- **Mạch Reset**

Chân Reset (chân 9) có tác dụng reset chip AT89C51, mức tích cực của chân này là mức 1, để reset ta phải đưa mức 1(5V) đến chân này với thời gian tối thiểu 2 chu kỳ máy (tương đương 2 μ S - đối với thạch anh 12MHz).

Tần số làm việc AT89C51 $f = f(TA)/12$

Chu kỳ máy = thời gian xử lý 1 lệnh: $T = 1/f$

Khi reset thì trạng thái của RAM nội không thay đổi.

Thanh ghi	Nội dung
Bộ đếm chương trình PC	0000H
Thanh ghi tích lũy A	00H
Thanh ghi B	00H
Từ trạng thái chương trình PSW	00H
Con trỏ stack SP	07H
Con trỏ dữ liệu DPTR	0000H
Port 0 -3	FFH
Thanh ghi ưu tiên ngắt IP	xxx00000B(8031/8051) xx000000B(8032/8052)
Thanh ghi cho phép ngắt IE	0xx00000B(8031/8051) 0x000000B(8032/8052)
Các thanh ghi định thời	00H
Thanh ghi điều khiển nối tiếp SCON	00H
Bộ đệm dữ liệu nối tiếp SBUF	00H
Thanh ghi điều khiển nguồn PCON(HMOS)(CMOS)	0xxxxxxxB 0xxx0000B

Bảng 3.1: Trạng thái của các thanh ghi khi Reset

3.2.2. Mạch hiển thị đếm ngược LED 7 đoạn

Khởi hiển thị đếm ngược led 7 đoạn: hiển thị thời gian của các chế độ thời gian, được hiển thị trên led 7 đoạn, các chân của led được nối thông qua một điện trở thành để nối với vi điều khiển.

Việc giải mã nhị phân sang led 7 đoạn đều được thực hiện trên bộ giải mã 74247. Led 7 đoạn được nối theo kiểu anode chung, trong mạch này: chân Anode chung của các led được điều khiển gián tiếp từ chân P0.0 đến P1.3. Led hoạt động với dòng từ 10mA đến 20mA. Sử dụng các port (Q):QA, QB, QC, QD, QE, QF, QG để điều khiển các chân Cathode. Với các chân Anode chung được điều khiển thông qua các transistor H1061, điều khiển hoạt động ở chế độ bão hòa cung cấp dòng cho LED sáng. Các chân 3.0, 3.1, 3.5, 3.6, 3.7 dùng để điều khiển thời gian sáng tối của các đèn báo xanh đỏ vàng của hai làn đường.

Cụ thể bảng địa chỉ như sau và mức điều khiển các Led 7 đoạn như sau.

a. Địa chỉ điều khiển các chân Cathode của Led 7 đoạn:

- PQA: Thanh A của các led 7 đoạn.
- PQB: Thanh B của các led 7 đoạn.
- PQC: Thanh C của các led 7 đoạn.
- PQD: Thanh D của các led 7 đoạn.
- PQE: Thanh E của các led 7 đoạn.
- PQF: Thanh F của các led 7 đoạn.
- PQG: Thanh G của các led 7 đoạn.

b. Mức điều khiển:

Các thanh của led 7 đoạn A nốt chung tương ứng từ A tới G sẽ sáng khi các chân Cathode mức thấp và chân Anode chung ở mức cao.

Vì vậy các led 7 thanh sẽ sáng khi chân tương ứng từ PQA đến PQB ở mức thấp, và các chân Anode chung ở mức cao, hay các chân điều khiển cực B các Transistor NPN tương ứng phải ở mức cao.

Các thanh led tương ứng sẽ tắt khi thiếu một trong hai điều kiện, hoặc cả hai điều kiện trên.

c. Địa chỉ điều khiển các chân Anode chung của led 7 đoạn:

- P2.0: chân A nốt chung (LED hàng chục) của làn đường 1.
- P2.1: chân A nốt chung (LED hàng đơn vị) của làn đường 1.
- P2.2: chân A nốt chung (LED hàng chục) của làn đường 2.
- P2.3: chân A nốt chung (LED hàng đơn vị) của làn đường 2.

3.2.3. Mạch hiển thị LED đơn

- Địa chỉ:
 - P3.7: Nối với đèn đỏ đường 1.
 - P3.6: Nối với đèn vàng đường 1.
 - P3.5: Nối với đèn xanh đường 1.
 - P3.4: Nối với đèn đỏ đường 2.
 - P3.1: Nối với đèn vàng đường 2.
 - P3.0: Nối với đèn xanh đường 2.

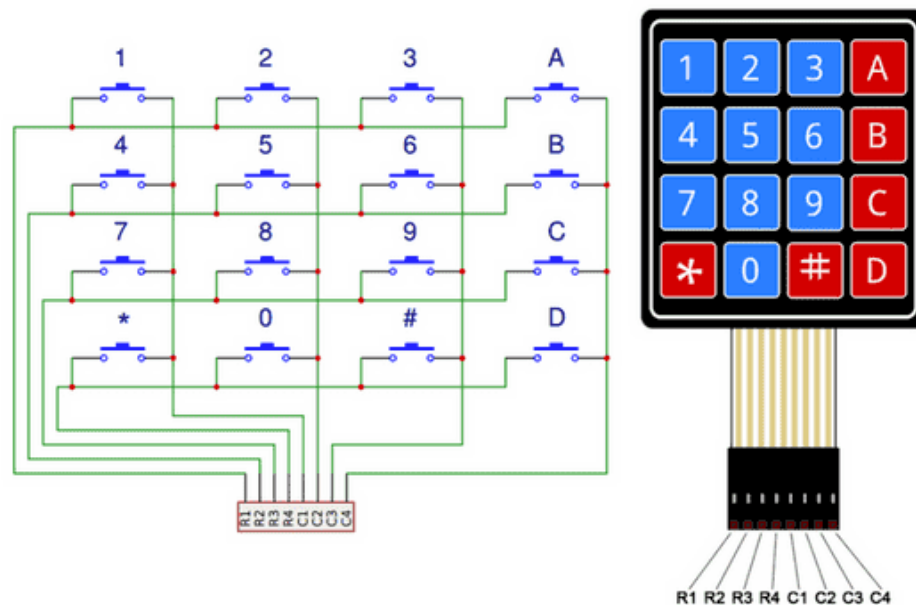
Vì LED đơn có dòng hoạt động khoảng 10mA trở lên nên ta có thể chọn điện trở R sao cho phù hợp.

Nguồn được cấp: $V_{cc} = +5V$.

Điện áp phân cực thuận của các LED:

- Led xanh: $V_x = 2V - 2.8V$.
 - Led vàng: $V_v = 2V - 2.5V$.
 - Led đỏ: $V_d = 1.4V - 1.8V$.
- Mức điều khiển các đèn báo vàng xanh đỏ.
 - Mức 1: Đèn sáng.
 - Mức 0: Đèn tắt

3.2.4. Keypad



Hình 3.3: Keypad và sơ đồ nguyên lý hoạt động

Keypad là một thiết bị nhập liệu phổ biến, được sử dụng rộng rãi trong các hệ thống nhúng để nhập dữ liệu từ người dùng. Trong Proteus, mô phỏng nguyên lý hoạt động của keypad giúp bạn hiểu cách mà vi điều khiển đọc các phím được nhấn. Dưới đây là nguyên lý hoạt động cơ bản của một keypad 4x4 trong Proteus:

- Nguyên lý hoạt động của Keypad

1. Cấu trúc của Keypad:

- Keypad 4x4 gồm 4 hàng (rows) và 4 cột (columns), tạo thành ma trận 4x4.
- Mỗi phím trên keypad nằm tại giao điểm của một hàng và một cột.

2. Kết nối Keypad với Vi điều khiển:

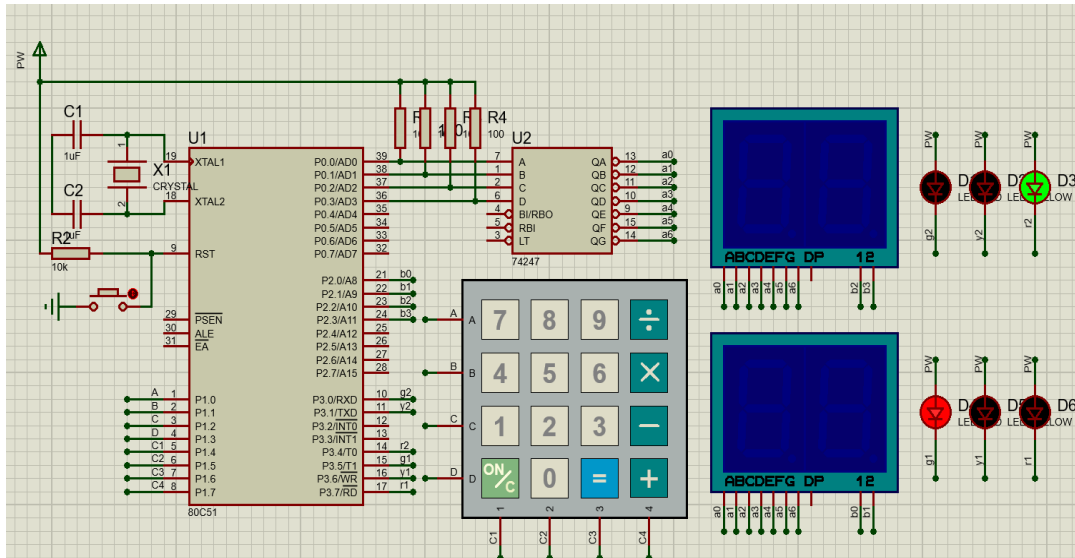
- Các hàng và cột của keypad được kết nối tới các chân GPIO (General Purpose Input/Output) của vi điều khiển.

- Thường, hàng (rows) được thiết lập làm đầu ra (output) và cột (columns) được thiết lập làm đầu vào (input) với điện trở kéo lên (pull-up resistor).

3. Quét Keypad để phát hiện phím nhấn:

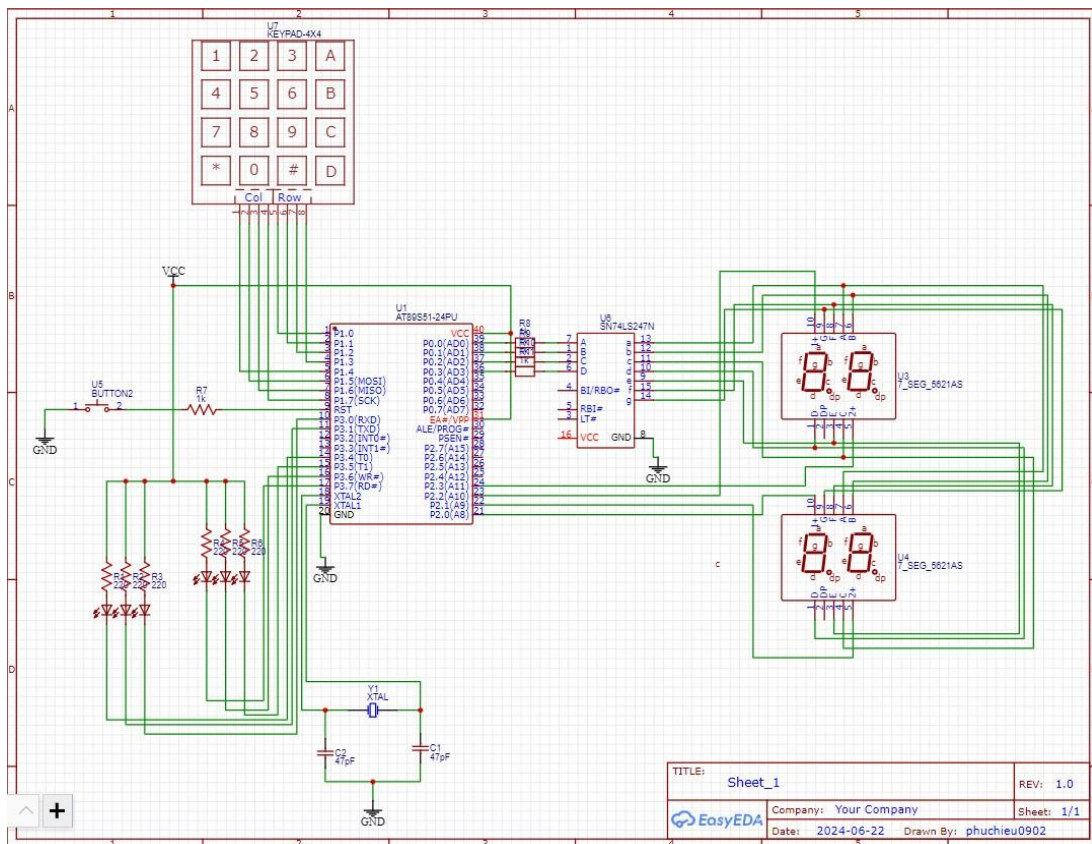
- Vi điều khiển sẽ lần lượt kích hoạt từng hàng (output) bằng cách đưa mức logic thấp (0) lên hàng đó, đồng thời để các hàng khác ở mức logic cao (1).
 - Sau đó, vi điều khiển sẽ đọc trạng thái của các cột (input).
 - Nếu một phím được nhấn, nó sẽ tạo thành một mạch kín giữa hàng đang được kích hoạt và cột tương ứng của phím đó. Điều này làm cho cột tương ứng chuyển từ mức logic cao (1) sang mức logic thấp (0).
 - Bằng cách biết hàng nào đang được kích hoạt và cột nào có mức logic thấp, vi điều khiển có thể xác định phím nào được nhấn.
- Địa chỉ:
- P1.0: Nối với hàng A.
 - P1.1: Nối với hàng B.
 - P1.2: Nối với hàng C.
 - P1.3: Nối với hàng D.
 - P1.4: Nối với cột 1.
 - P1.5: Nối với cột 2.
 - P1.6: Nối với cột 3.
 - P1.7: Nối với cột 4.

3.2.5. Sơ đồ nguyên lý của hệ thống



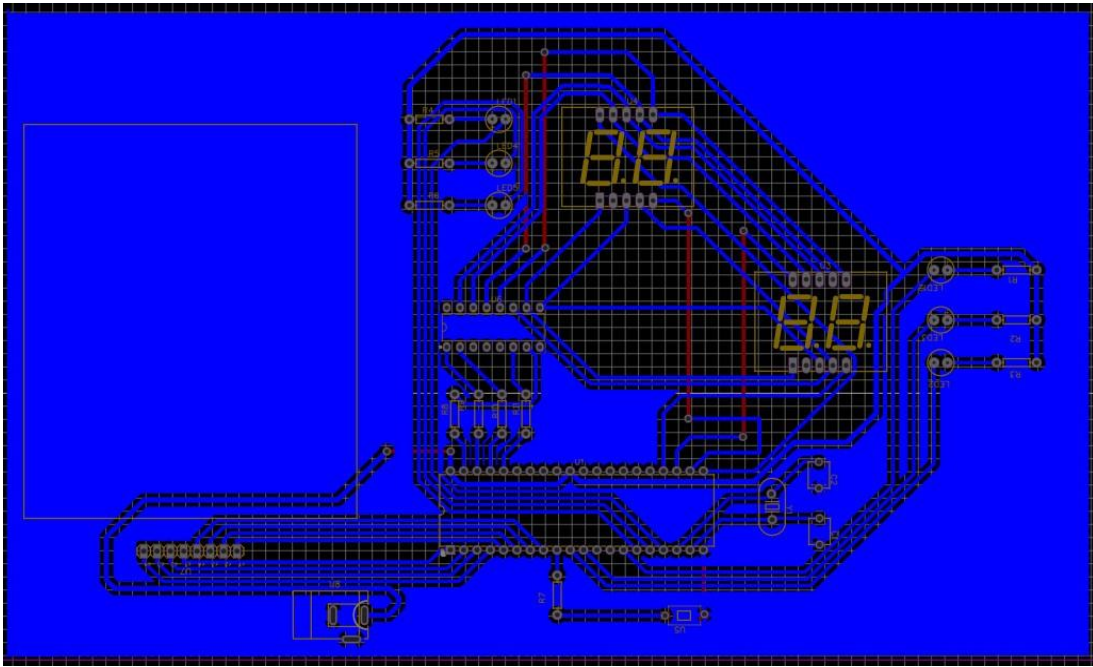
Hình 3.4: Sơ đồ nguyên lý của hệ thống

3.2.6. Sơ đồ thiết kế



Hình 3.5: Sơ đồ thiết kế của mạch

3.2.7. Sơ đồ PCB

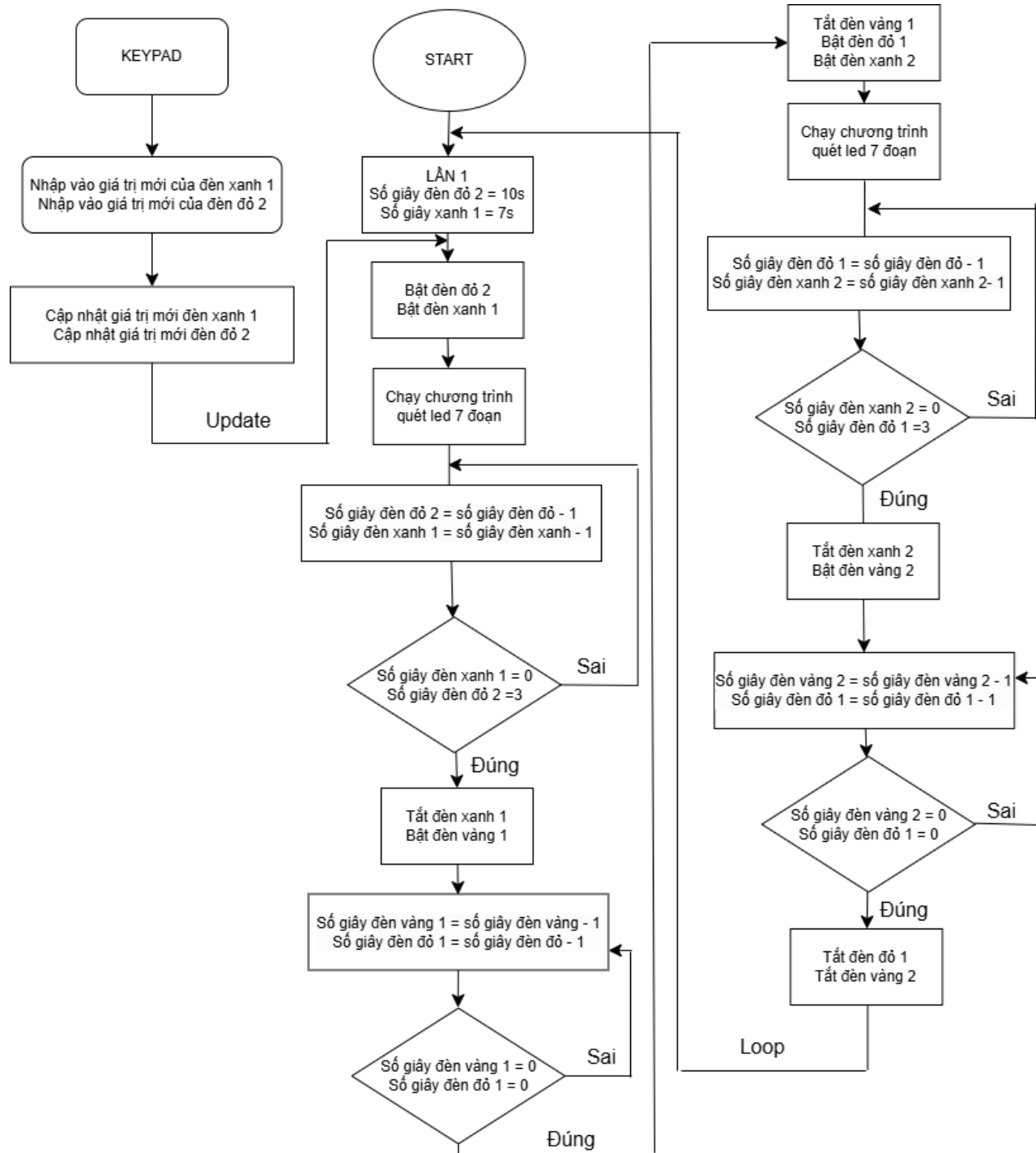


Hình 3.6: Sơ đồ PCB của mạch

3.3. Xây dựng, thiết kế chương trình

Sử dụng ngôn ngữ lập trình Assembly trên phần mềm Proteus 8.16 để thiết kế chương trình cho hệ thống.

3.3.1. Lưu đồ thuật toán



3.3.2. Chương trình Assembly cho 89C51

```
ORG 0000h
JMP Start
ORG 000BH ; Timer 0 interrupt vector
LJMP T0_ISR
ORG 0003H ; External interrupt 0 vector
LJMP INT0_ISR
ORG 0100h

Start:
TGXANH EQU 7
TGVANG EQU 3
TGDO EQU 10
MOV TMOD, #01H ; Timer 0 in mode 1
MOV TH0, #0EEH
MOV TL0, #06H
MOV IE, #82H ; Enable Timer 0 interrupt and global interrupt

SETB TR0 ; Start Timer 0
SETB IT0 ; Enable external interrupt 0

MOV R0, #0 ; Cycle counter
MOV R1, #7 ; Initial time for traffic light 1 (7-3-10)
MOV R2, #1 ; Initial state

MOV R3, #10 ; Initial time for traffic light 2
MOV R4, #1 ; Initial state

MOV R5, #0 ; Manage pause state

CLR P3.0
CLR P3.7

Loop:
; Multiplexing for 7-segment display for traffic light 1
SETB P2.1
MOV A, R1
MOV B, #0AH
DIV AB
MOV P0, B
```

```
CALL DELAY
CLR P2.1
SETB P2.0
MOV P0, A
CALL DELAY
CLR P2.0
```

```
; Multiplexing for 7-segment display for traffic light 2
```

```
SETB P2.3
MOV A, R3
MOV B, #0AH
DIV AB
MOV P0, B
CALL DELAY
CLR P2.3
SETB P2.2
MOV P0, A
CALL DELAY
CLR P2.2
```

```
CALL KEYPAD
LJMP Loop ; Jump back to main loop
```

```
INT0_ISR:
    RETI
```

```
T0_ISR:
    MOV TH0, #0EEH
    MOV TL0, #06H
    SETB TR0
    INC R0
    CJNE R0, #200, END_T0_ISR
```

```
; Handling traffic light 1 timing
```

```
DEC R1
```

```
MOV R0, #0
```

```
XANH:
```

```
CJNE R2, #1, VANG
```

```
CJNE R1, #0, CON_X
```

```
MOV R1, #TGVANG
```

```
INC R2
JMP VANG
```

```
CON_X:
    CLR P3.7
    SETB P3.5
VANG:
    CJNE R2, #2, DO
    CJNE R1, #0, CON_V
    MOV R1, #TGDO
    INC R2
    JMP DO
CON_V:
    CLR P3.6
    SETB P3.7
DO:
    CJNE R2, #3, GIAY_2
    CJNE R1, #0, CON_DO
    MOV R1, #TGXANH
    MOV R2, #1
    SETB P3.5
    JMP XANH
CON_DO:
    CLR P3.5
    SETB P3.6
```

; Handling traffic light 2 timing

```
GIAY_2:
    DEC R3
DO_2:
    CJNE R4, #1, XANH_2
    CJNE R3, #0, CON_DO_2
    MOV R3, #TGXANH
    INC R4
    JMP XANH_2
CON_DO_2:
    CLR P3.0
    SETB P3.1
XANH_2:
    CJNE R4, #2, VANG_2
```



```

    CJNE R3, #0, CON_X_2
    MOV R3, #TGVANG
    INC R4
    JMP VANG_2
CON_X_2:
    CLR P3.4
    SETB P3.0
VANG_2:
    CJNE R4, #3, END_T0_ISR
    CJNE R3, #0, CON_V_2
    MOV R3, #TGDO
    MOV R4, #1
    SETB P3.1
    JMP DO_2
CON_V_2:
    CLR P3.1
    SETB P3.4

END_T0_ISR:
    RETI

KEYPAD:
    MOV P1, #0F0H ; Set high nibble as input, low nibble as output
KIEMTRACOT:
    MOV P1, #0F7H ; Row D
    MOV A, P1
    ANL A, #0F0H
    CJNE A, #0F0H, SCAN_R0

    MOV P1, #0FBH ; Row C
    MOV A, P1
    ANL A, #0F0H
    CJNE A, #0F0H, SCAN_R1

    MOV P1, #0FDH ; Row B
    MOV A, P1
    ANL A, #0F0H
    CJNE A, #0F0H, SCAN_R2

    MOV P1, #0FEH ; Row A

```

```
MOV A, P1
ANL A, #0F0H
CJNE A, #0F0H, SCAN_R3
RET
```

SCAN_R0:

```
MOV DPTR, #MAHANGD
MOV B, #16
DIV AB
CJNE A, #0EH, END_CN
CJNE R5, #0, RESUME
```

PAUSE:

```
INC R5
CLR TR0
JMP END_SCAN0
```

RESUME:

```
DEC R5
SETB TR0
JMP END_SCAN0
```

END_CN:

```
LCALL SCAN
END_SCAN0:
RET
```

SCAN_R1:

```
MOV DPTR, #MAHANGC
MOV B, #16
DIV AB
LCALL SCAN
RET
```

SCAN_R2:

```
MOV DPTR, #MAHANGB
MOV B, #16
DIV AB
LCALL SCAN
RET
```

SCAN_R3:

```
MOV DPTR, #MAHANGA
```

```
MOV B, #16
DIV AB
LCALL SCAN
RET
```

SCAN:

```
RRC A
JNC MATCH
INC DPTR
SJMP SCAN
MATCH:
MOV A, #0
MOVC A, @A + DPTR
JB TR0, END_SCAN
CALL SETTING_TIME
END_SCAN:
RET
```

SETTING_TIME:

```
MOV R1, A
ADD A, #3
MOV R3, A
RET
```

DELAY:

```
MOV R7, #255 ; Adjusted delay value
DL1:
DJNZ R7, DL1
RET
```

ORG 300H

```
MAHANGA: DB 7, 8, 9, '0'
MAHANGB: DB 4, 5, 6, '0'
MAHANGC: DB 1, 2, 3, '0'
MAHANGD: DB 'C', 0, '=', '0'
```

END

CHƯƠNG 4: KẾT LUẬN

Sau khi đã hoàn thành xong đề án, nhóm em rút ra được nhiều bài học cũng như là những kết quả như sau:

- Giúp hiểu sâu hơn về họ vi điều khiển 8051 nói riêng, cũng như là vi điều khiển AT89C51 nói chung, bao gồm kiến trúc, chức năng, cách lập trình.
- Nắm vững hơn về lập trình bằng ngôn ngữ Assembly, tạo cơ sở để làm việc với các vi điều khiển khác hiệu quả hơn
- Học được những cách tích hợp các thành phần phần cứng với vi điều khiển, cũng như việc lựa chọn các linh kiện sao cho phù hợp và bố trí chúng trên mạch.

Bên cạnh đó, nhóm em cũng gặp những khó khăn và hạn chế trong quá trình thực hiện đề án như:

- Kỹ năng lập trình cũng như debug vẫn chưa được tối ưu hóa, khiến cho trong quá trình làm tốn nhiều thời gian
- Thông tin cũng như các nguồn tham khảo quá nhiều dẫn đến việc rất khó để có thể khoanh vùng những nội dung phù hợp cho đề án

Phương hướng phát triển: Trong tương lai, để có thể cải tiến chất lượng, cũng như đối với sự phát triển thêm của đề án này, nhóm em xin đề xuất một vài ý kiến như sau:

- Tích hợp một số cảm biến (cảm biến áp suất đo số lượng xe cộ đi lại, cảm biến hồng ngoại hoặc siêu âm)
- Ứng dụng IoT để tạo ra hệ thống đèn giao thông thông minh, có khả năng tối ưu và học hỏi liên tục

TÀI LIỆU THAM KHẢO

- [1] **"The 8051 Microcontroller and Embedded Systems: Using Assembly and C"** - Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay
- [2] **"Programming and Customizing the 8051 Microcontroller"** - Myke Predko
- [3] **"The 8051 Microcontroller"** - Kenneth Ayala
- [4] **“Giáo trình VI ĐIỀU KHIỂN”** - TS. Vũ Đức Lung, TS. Lê Quang Minh, ThS. Phan Đình Duy

Link Đồ án github: <https://github.com/hoangdylone/MoPhongDenGiaoThong>