

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI



Báo cáo cuối kì môn học:
Vi xử lý

LẬP TRÌNH TRÊN VI ĐIỀU KHIỂN AT89S52

Sinh viên thực hiện:

Tạ Quang Tùng
MSSV: 20154280

Giáo viên hướng dẫn:

TS. Ngô Lam Trung

Hà Nội, Ngày 13 tháng 6 năm 2018

Mục lục

1	Giới thiệu về vi điều khiển 89S52	1
1.1	Các dòng vi điều khiển	1
1.2	Các chân của AT89S52	2
2	Lập trình ngắt trên 8051	4
2.1	Giới thiệu về ngắt	4
2.2	Các ngắt trong 8051	4
2.3	Các trình phục vụ ngắt 8051	4
2.4	Cho phép và cấm ngắt trong 8051	5
2.5	Mức độ ưu tiên các ngắt trong 8051	5
2.6	Ngắt timer trong 8051	6
3	Bài toán mạch tưới cây báo khô hạn	8
3.1	Mô tả bài toán	8
3.2	Ý tưởng thực hiện	9
3.3	Kết quả thực hiện	10

1 Giới thiệu về vi điều khiển 89S52

1.1 Các dòng vi điều khiển

Năm 1980 khi intel tung ra chip 8051, bộ vi điều khiển đầu tiên của họ MCS-51 và là chuẩn công nghệ cho nhiều họ vi điều khiển được sản xuất sau này. Năm 1980 Intel công bố chip 8051 (80C51), bộ vi điều khiển đầu tiên của họ vi điều khiển MCS-51 bao gồm:

- 4 KB ROM
- 128 byte RAM
- 32 đường xuất nhập
- 1 port nối tiếp và 2 timer/counter 16 bit

Hiện nay, Intel không còn cung cấp các loại Vi điều khiển họ MCS-51 nữa, thay vào đó các nhà sản xuất khác như Atmel, Philips/signetics, AMD, Siemens, Matra & Dallas, Semiconductors được cấp phép làm nhà cung cấp thứ hai cho các chip của họ MSC-51.

Chip Vi điều khiển được sử dụng rộng rãi trên thế giới cũng như ở Việt Nam hiện nay là Vi điều khiển của hãng Atmel. Các mã số chip được thay đổi chút ít khi được Atmel sản xuất. Mã số 80 chuyển thành 89, chẳng hạn 80C52 của Intel khi sản xuất ở Atmel mã số thành 89C52 (Mã số đầy đủ: AT89C52). Với tính năng chương trình tương tự như nhau. Tương tự 8051, 8053, 8055 có mã số tương đương ở Atmel là 89C51, 89C53, 89C55.

Sau khoảng thời gian cải tiến và phát triển, hãng Atmel tung ra thị trường dòng Vi điều khiển mang số hiệu 89Sxx với nhiều cải tiến và đặc biệt là có thêm khả năng nạp chương trình theo chế độ nối tiếp rất đơn giản và tiện lợi cho người sử dụng.

Dung lượng RAM Dung lượng ROM Chế độ nạp:

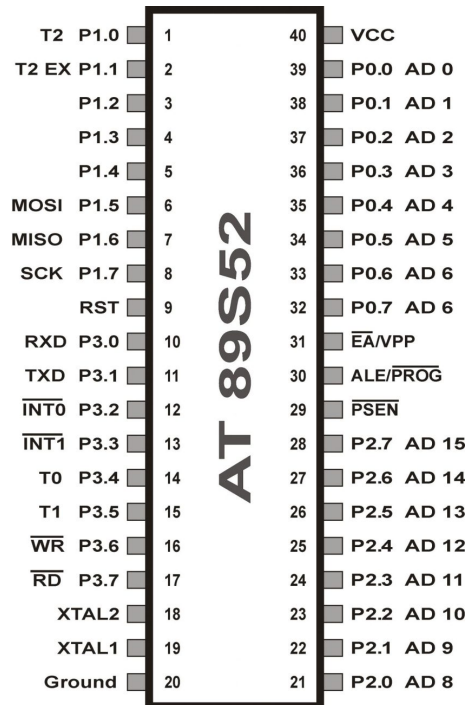
- 89S51 128 byte 4 Kbyte nối tiếp.
- 89S52 128 byte 8 Kbyte nối tiếp.
- 89S53 128 byte 12 Kbyte nối tiếp.
- 89S55 128 byte 20 Kbyte nối tiếp.

Tổng quát thông số kỹ thuật của AT89S52:

- 8 KB bộ nhớ có thể lập trình nhanh, có khả năng tới 1000 chu kỳ ghi/xoá.
- Tần số hoạt động từ: 0Hz đến 24 MHz
- 3 mức khóa bộ nhớ lập trình
- 3 bộ Timer/counter 16 Bit
- 128 byte RAM nội.
- 4 port xuất /nhập I/O 8 bit.
- Giao tiếp nối tiếp.

- 64 KB vùng nhớ mã ngoài
- 64 KB vùng nhớ dữ liệu ngoài.
- $4\mu s$ cho hoạt động nhân hoặc chia

1.2 Các chân của AT89S52



Hình 1: Sơ đồ chân

- Nhóm chân nguồn:
 - VCC: chân 40, điện áp cung cấp 5V DC.
 - GND: chân 20.
- Nhóm chân dao động: Gồm chân 18 và chân 19 (XTAL2 và XTAL1), cho phép ghép nối thạch anh vào mạch dao động bên trong vi điều khiển, được sử dụng để nhận nguồn xung clock từ bên ngoài để dao động, thường được ghép nối với thạch anh và các tụ để tạo nguồn xung clock ổn định.
 - XTAL1: ngõ vào đến mạch khuếch đại dao động đảo và ngõ vào đến mạch tạo dao động xung clock bên trong.
 - XTAL2: Ngõ ra từ mạch khuếch đại dao động đảo.
- Chân chọn bộ nhớ chương trình: chân 31 (EA/VPP) dùng để xác định chương trình thực hiện được lấy từ ROM nội hay ROM ngoài.
 - Chân 31 nối đất: sử dụng bộ nhớ bên ngoài.

- Chân 31 nối nguồn: sử dụng bộ nhớ chương trình (4KB) bên trong vi điều khiển.
- RST(chân reset): chân 9 dùng để thiết lập trạng thái ban đầu cho vi điều khiển.
- Chân cho phép bộ nhớ chương trình PSEN: PSEN (program store enable) tín hiệu được xuất ra ở chân 29 dùng để truy xuất bộ nhớ chương trình ngoài. Chân này thường được nối với chân OE (output enable) của ROM ngoài. Khi vi điều khiển làm việc với bộ nhớ chương trình ngoài, chân này phát ra tín hiệu kích hoạt ở mức thấp và được kích hoạt 2 lần trong một chu kỳ máy. Khi thực thi một chương trình ở ROM nội, chân này được duy trì ở mức logic không tích cực (logic 1).
- Chân ALE: chân 30, chốt địa chỉ. Khi Vi điều khiển truy xuất bộ nhớ từ bên ngoài, port 0 vừa có chức năng là bus địa chỉ, vừa có chức năng là bus dữ liệu do đó phải tách các đường dữ liệu và địa chỉ. Tín hiệu ở chân ALE dùng làm tín hiệu điều khiển để giải đa hợp các đường địa chỉ và các đường dữ liệu khi kết nối chúng với IC chốt.
- Nhóm chân điều khiển vào/ra:
 - Port 0, gồm 8 chân (từ chân 32 đến 39) có hai chức năng:
 - * Chức năng xuất/nhập: các chân này được dùng để nhận tín hiệu từ bên ngoài vào để xử lý, hoặc dùng để xuất tín hiệu ra bên ngoài, chẳng hạn xuất tín hiệu để điều khiển led đơn sáng tắt.
 - * Chức năng là bus dữ liệu và bus địa chỉ (AD7-AD0): 8 chân này (hoặc Port 0) còn làm nhiệm vụ lấy dữ liệu từ ROM hoặc RAM ngoài (nếu có kết nối với bộ nhớ ngoài), đồng thời Port 0 còn được dùng để định địa chỉ của bộ nhớ ngoài.
 - Port 1, gồm 8 chân (từ chân 1 đến chân 8) chỉ có chức năng làm các đường xuất/nhập, không có chức năng khác.
 - Port 2, gồm 8 chân (từ chân 21 đến chân 28) có hai chức năng:
 - * Chức năng xuất/nhập
 - * Chức năng là bus địa chỉ cao (A8-A15): khi kết nối với bộ nhớ ngoài có dung lượng lớn, cần 2 byte để định địa chỉ của bộ nhớ, byte thấp do P0 đảm nhận, byte cao do P2 này đảm nhận.
 - Port 3, gồm 8 chân (từ chân 10 đến 17) với chức năng chính xuất/nhập.
- Các chức năng khác:
 - P3.0 RxD: Ngõ vào nhận dữ liệu nối tiếp.
 - P3.1 TxD: Ngõ xuất dữ liệu nối tiếp.
 - P3.2 INT0: Ngõ vào ngắt cứng thứ 0.
 - P3.3 INT1: Ngõ vào ngắt cứng thứ 1.
 - P3.4 T0: Ngõ vào của timer/counter thứ 0.
 - P3.6 WR: Ngõ điều khiển ghi dữ liệu lên bộ nhớ ngoài
 - P3.7 RD: Ngõ điều khiển đọc dữ liệu từ bộ nhớ bên ngoài
 - P1.0 T2: Ngõ vào của timer/counter thứ 2
 - P1.1 T2 EX: Ngõ Nạp lại/thu nhận của timer/counter thứ 2

2 Lập trình ngắt trên 8051

2.1 Giới thiệu về ngắt

Ngắt (Interrupt) - là một số sự kiện khẩn cấp bên trong hoặc bên ngoài bộ vi điều khiển xảy ra, buộc vi điều khiển tạm dừng thực hiện chương trình hiện tại, phục vụ ngay lập tức nhiệm vụ mà ngắt yêu cầu – nhiệm vụ này gọi là trình phục vụ ngắt (ISR: Interrupt Service Routine).

Phân biệt ngắt với hồi vòng:

- Trong phương pháp sử dụng ngắt: mỗi khi có một thiết bị bất kỳ cần được phục 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 ngừng tất cả những gì nó đang thực hiện để chuyển sang phục vụ thiết bị gọi ngắt. Chương trình ngắt được gọi là trình phục vụ ngắt ISR (Interrupt Service Routine) hay còn gọi là trình quản lý ngắt (Interrupt Handler). Sau khi phục vụ ngắt xong, bộ vi xử lý lại quay trở lại điểm bị ngắt trước đó và tiếp tục thực hiện công việc.
- Trong phương pháp thăm dò: bộ vi điều khiển kiểm tra liên tục tình trạng của tất cả các thiết bị, nếu thiết bị nào có yêu cầu thì nó dừng lại phục vụ thiết bị đó. Sau đó nó tiếp tục kiểm tra tình trạng của thiết bị kế tiếp cho đến hết. Phương pháp thăm dò rất đơn giản, nhưng nó lại rất lãng phí thời gian để kiểm tra các thiết bị kể cả khi thiết bị đó không cần phục vụ. Trong trường hợp có quá nhiều thiết bị thì phương án thăm dò tỏ ra không hiệu quả, gây ra chậm trễ cho các thiết bị cần phục vụ.

2.2 Các ngắt trong 8051

Thực tế chỉ có 5 ngắt dành cho người dùng trong 8051 nhưng các nhà sản xuất nói rằng có 6 ngắt vì họ tính cả lệnh RESET. Sáu ngắt của 8051 được phân bố như sau:

- RESET: Khi chân RESET được kích hoạt từ 8051, bộ đếm chương trình nhảy về địa chỉ 0000H. Đây là địa chỉ bật lại nguồn.
- 2 ngắt dành cho các bộ định thời: 1 cho Timer0 và 1 cho Timer1. Địa chỉ tương ứng của các ngắt này là 000BH và 001BH.
- 2 ngắt dành cho các ngắt phần cứng bên ngoài: chân 12 (P3.2) và 13 (P3.3) của cổng P3 là các ngắt phần cứng bên ngoài INT0 và INT1 tương ứng. Địa chỉ tương ứng của các ngắt ngoài này là 0003H và 0013H.
- Truyền thông nối tiếp: có 1 ngắt chung cho cả nhận và truyền dữ liệu nối tiếp. Địa chỉ của ngắt này trong bảng vector ngắt là 0023H.

2.3 Các trình phục vụ ngắt 8051

Đối với mỗi ngắt thì phải có một trình phục vụ ngắt (ISR) hay trình quản lý ngắt để đưa ra nhiệm vụ cho bộ vi điều khiển khi được gọi ngắt. Khi một ngắt được gọi thì bộ vi điều khiển sẽ chạy trình phục vụ ngắt. Đối với mỗi ngắt thì có một vị trí cố định trong bộ nhớ để giữ địa chỉ ISR của nó. Nhóm vị trí bộ nhớ được dành riêng để lưu giữ địa chỉ của các ISR được gọi là bảng vector ngắt.

Ngắt	Cờ ngắt	Địa chỉ trình phục vụ ngắt	Số thứ tự ngắt
Reset	-	0000h	-
Ngắt ngoài 0	IE0	0003h	0
Timer 0	TF0	000Bh	1
Ngắt ngoài 1	IE1	0013h	2
Timer 1	TF1	001Bh	3
Ngắt truyền thông	RI/TI	0023h	4

Bảng 1: Bảng vector ngắt của 8051

Trong lập trình C trên Keil C cho 8051, chúng ta khai báo trình phục vụ ngắt bằng cách thêm interrupt X vào sau khai báo hàm (X là số thứ tự ngắt).

2.4 Cho phép và cấm ngắt trong 8051

Khi bật lại nguồn thì tất cả mọi ngắt đều bị cấm (bị che), có nghĩa là không có ngắt nào được bộ vi điều khiển đáp ứng trừ khi chúng được kích hoạt.

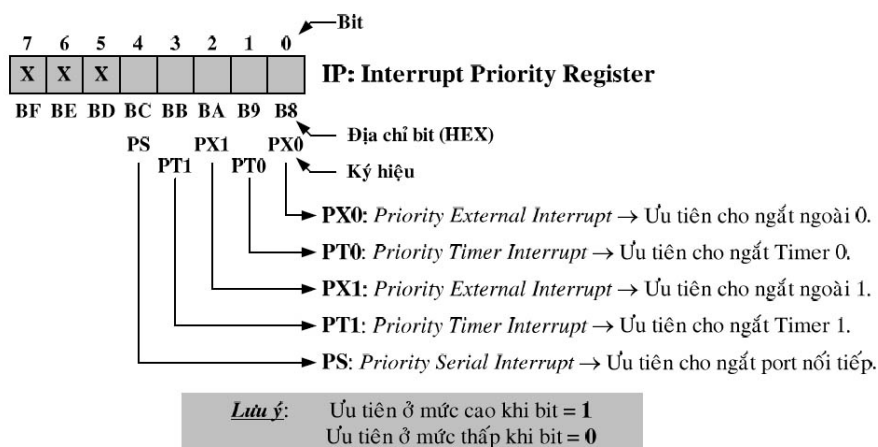
Các ngắt phải được kích hoạt bằng phần mềm để bộ vi điều khiển đáp ứng chúng. Có một thanh ghi được gọi là thanh ghi cho phép ngắt **IE** (Interrupt Enable) – ở địa chỉ A8h chịu trách nhiệm về việc cho phép và cấm các ngắt.

Bit	Tên	Địa chỉ	Chức năng
7	EA	AFh	Cho phép/cấm hoạt động của cả thanh ghi.
6	-	AEh	Chưa sử dụng.
5	-	ADh	Chưa sử dụng.
4	ES	ACH	Cho phép ngắt cổng truyền thông nối tiếp.
3	ET1	ABh	Cho phép ngắt Timer 1.
2	EX1	AAh	Cho phép ngắt ngoài 1.
1	ET0	A9h	Cho phép ngắt Timer 0.
0	EX0	A8h	Cho phép ngắt ngoài 0.

Bảng 2: Thanh ghi cho phép ngắt **IE**

2.5 Mức độ ưu tiên các ngắt trong 8051

Mỗi ngắt có 2 chế độ ưu tiên: 0 hoặc 1. Được thiết lập nhờ thanh ghi IP.



2.6 Ngắt timer trong 8051

Có 2 ngắt timer trong 8051 là timer 0 và timer 1 với chức năng giống nhau. Các ngắt timer được thiết lập chế độ hoạt động bằng thanh ghi TMOD.

Bit	Tên	Chức năng
7	Gate	-
6	C/T	1 nếu như sử dụng timer/counter 1 làm counter, 0 nếu như sử dụng làm timer.
5	M1	Bit cao thiết lập chế độ cho timer/counter 1.
4	M0	Bit thấp thiết lập chế độ cho timer/counter 1.
3	Gate	-
2	C/T	1 nếu như sử dụng timer/counter 0 làm counter, 0 nếu như sử dụng làm timer.
1	M1	Bit cao thiết lập chế độ cho timer/counter 0.
0	M0	Bit thấp thiết lập chế độ cho timer/counter 0.

Bảng 3: Các bit trong thanh ghi TMOD (Timer Mode Control)

M1	M0	Chế độ	Mô tả
0	0	0	Sử dụng thanh ghi TH làm bộ đếm 8 bit và thanh ghi TL làm bộ đếm 5 bit
0	1	1	Sử dụng thanh ghi TH làm bộ đếm 8 bit và thanh ghi TL làm bộ đếm 8 bit
1	0	2	Sử dụng chỉ thanh ghi TL làm bộ đếm 8 bit
1	1	3	Timer 0 trong chế độ 3 trở thành 2 bộ đếm 8 bit tách biệt. Timer 1 vẫn có thể được sử dụng nhưng không tạo ra tín hiệu ngắt

Bảng 4: Các chế độ của timer/counter

Bit	Kí hiệu	Chức năng
7	TF1	Cờ tràn của timer 1. Được xóa khi ngắt tương ứng được thực thi.
6	TR1	Điều khiển timer 1. Gán bằng 1 để bắt đầu đếm.
5	TF0	Cờ tràn của timer 0. Được xóa khi ngắt tương ứng được thực thi.
4	TR0	Điều khiển timer 0. Gán bằng 1 để bắt đầu đếm.
3	IE1	
2	IT1	Gán bằng 1 nếu timer 1 tích cực sườn âm, bằng 0 nếu tích cực mức thấp.
1	IE0	
0	IT0	Gán bằng 0 nếu timer 1 tích cực sườn âm, bằng 0 nếu tích cực mức thấp.

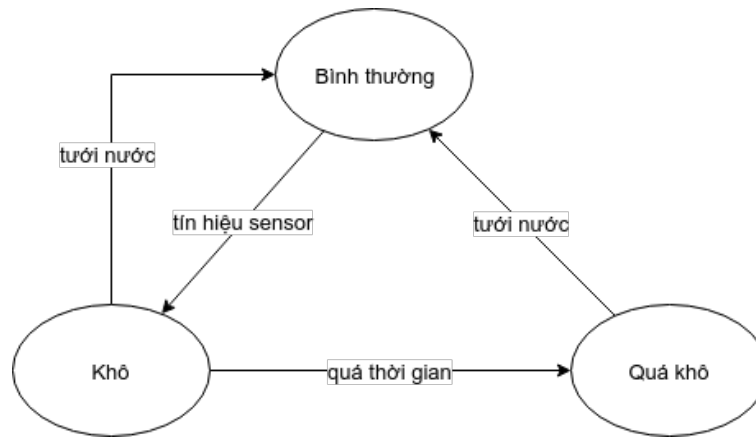
Bảng 5: Thanh ghi TCON (Timer Control)

Trong bài này chỉ sử dụng để làm timer (không sử dụng để làm counter) nên $C/T = 0$. Đồng thời 2 timer đều sử dụng ở chế độ 16 bit nên $M1 = 0$ và $M0 = 1$. Hai thanh ghi TH, TL là 2 thanh ghi lưu trữ giá trị đếm của timer. Khi giá trị này bị tràn thì cờ TF được set và tín hiệu ngắt được sinh ra.

Bộ đếm timer sẽ hoạt động với tần số bằng $1/12$ tần số của bộ dao động thạch anh.

3 Bài toán mạch tưới cây báo khô hạn

3.1 Mô tả bài toán



Hình 2: Các trạng thái của cây

Sơ đồ trạng thái khô hạn của cây bao gồm:

- **Bình thường:** Cây bắt đầu ở trạng thái này, đồng hồ báo khô hạn không hoạt động. Nếu nhận tín hiệu báo khô từ cảm biến, cây chuyển sang trạng thái **Khô**.
- **Khô:** Cây ở trạng thái này nếu như cảm biến đã báo hiệu độ ẩm dưới mức bình thường. Bắt đầu trạng thái này, đồng hồ báo thời gian khô hạn bắt đầu đếm. Cây sẽ chuyển sang trạng thái **Quá khô** nếu thời gian khô hạn vượt quá mức cho phép. Cây sẽ chuyển về trạng thái **Bình thường** nếu như được tưới nước.
- **Quá khô:** Ở trạng thái này đồng hồ báo thời gian sẽ hiển thị thời gian cây đã quá khô. Đồng thời tiếng chuông sẽ được reo liên tục để báo cần được tưới nước. Cây sẽ chuyển về trạng thái **Bình thường** nếu như được tưới nước.

Yêu cầu bài toán:

- Sử dụng một nút bấm để tưới nước và thiết lập thời gian giới hạn quá khô. Phân biệt ấn nhanh và giữ. Giữ nút sẽ chuyển từ chế độ này sang chế độ khác. Ấn nhanh ở chế độ hiển thị thời gian sẽ là tưới nước, ở chế độ thiết lập thời gian sẽ là thay đổi giá trị thời gian giới hạn (theo phút).
- Sử dụng một nút khác để mô phỏng việc báo quá khô, có chống nảy phím.
- Hiển thị thời gian (theo giây) lên màn hình LED 7 thanh, 4 số.
- Có báo chuông, nhấp nháy đèn khi ở trạng thái **Quá Khô**.

3.2 Ý tưởng thực hiện

Kiểm soát thời gian cho hệ thống

Để giữ thời gian cho hệ thống phục vụ cho nhiều tính toán khác nhau về thời gian thì ở đây sử dụng ngắt thời gian 0. Ngắt thời gian 0 được chạy ở chế độ ưu tiên cao nhất, liên tục thiết lập giá trị hai thanh ghi TH và TL để ngắt liên tục được gọi sau $50ms$. Giá trị biến nguyên không dấu 32 bit *counter0* được tăng từ 0 mỗi lần ngắt thời gian 0 được gọi.

Phân biệt ấn nhanh và giữ

Sử dụng ngắt ngoài 0 để làm nút tưới nước và thiết lập thời gian. Ngắt ngoài 0 sẽ được gọi liên tục với khoảng thời gian $< 1ms$ nếu như liên tục giữ.

Khi bắt đầu bấm để gọi ngắt ngoài 0 lần đầu tiên, sẽ lưu trữ giá trị của *counter0* đồng thời liên tục thiết lập TH1 TL1 cho ngắt thời gian 1 và thiết lập TR1 để bắt đầu chạy bộ đếm cho ngắt thời gian 1. Ngắt thời gian 1 sẽ được gọi sau $1ms$ nhưng nếu ngắt ngoài 0 liên tục được gọi (do được giữ nút) thì giá trị TH1 TL1 sẽ liên tục được đặt xuống giá trị dưới mức tràn, do đó ngắt thời gian 1 sẽ không được gọi nếu liên tục giữ nút.

Khi bỏ tay ra khỏi nút, sau $1ms$ thì ngắt thời gian 1 được gọi và dựa vào giá trị *counter0* lúc bắt đầu bấm nút và giá trị hiện tại ta có thể quyết định được khoảng thời gian giữ nút, từ đó phân biệt bấm nhanh và giữ (giữ nếu như quá $500ms$).

Hiển thị số trên đèn LED 7 thanh, 4 số

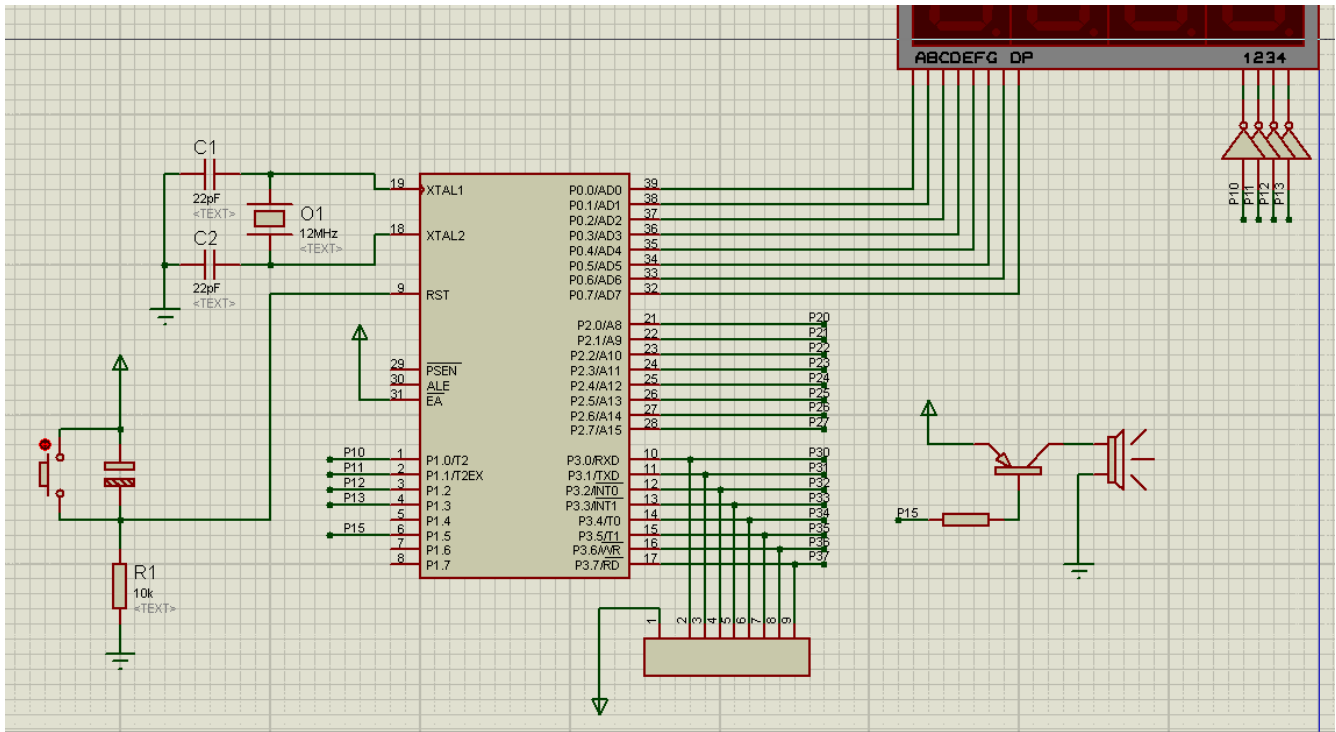
Trong bài toán này ta không xét đến dấu chấm trên đèn LED. 4 số trên LED 7 thanh được hiển thị đồng thời bằng cách hiển thị lần lượt từng số rồi xóa nó đi. Do hiện tượng lưu ảnh ở mắt nên người sẽ cảm nhận 4 đèn đều được hiển thị. Các đèn được hiển thị từ trái sang phải. Lần lượt bằng cách:

1. Đặt giá trị của P1.x tương ứng xuống mức 0.
2. Bật các thanh trong 7 thanh bằng đặt giá trị xuống mức 0 các thanh (từ thanh ghi P0) để tạo số tương ứng.
3. Delay một khoảng vài μs .
4. Tắt các đèn bằng đặt giá trị P1.x tương ứng trở về mức 1.

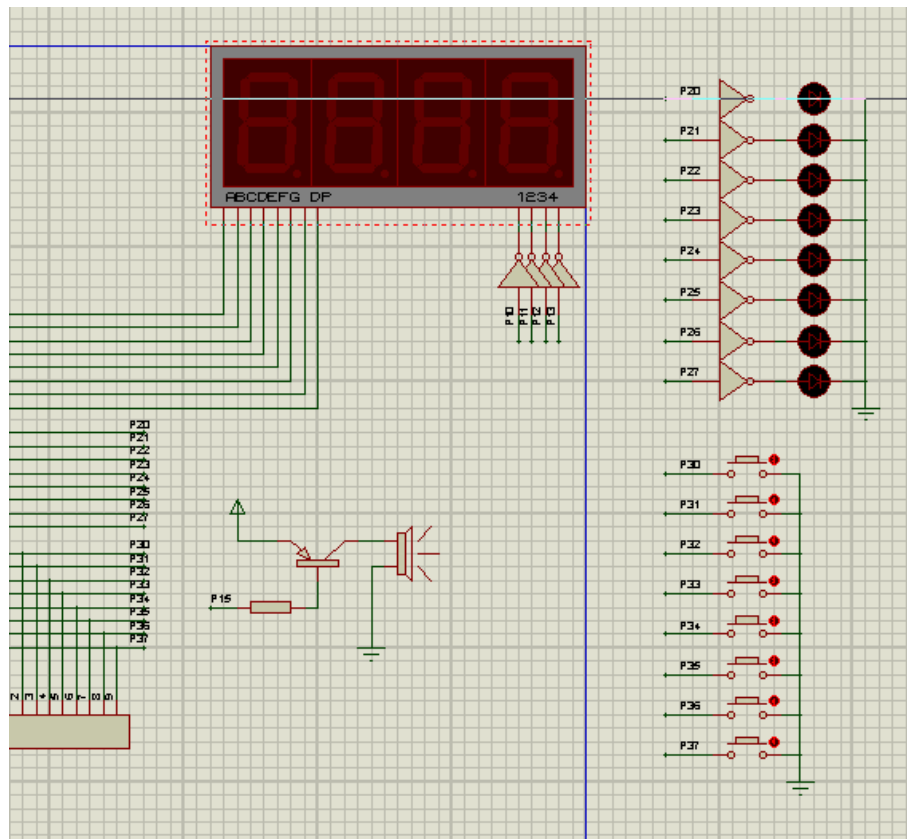
Phát chuông

Tín hiệu âm thanh được tạo ra nhờ liên tục đảo giá trị của thanh ghi P1.5. Trong bài toán này âm thanh được tạo ra với tần số $250Hz$ nhờ liên tục đảo bit thanh ghi P1.5 sau mỗi khoảng thời gian $2ms$ nhờ sử dụng ngắt thời gian 1. Tuy nhiên do có sử dụng chung với việc phân biệt ấn nhanh và giữ nút nên cần phải có biện pháp giảm xung đột.

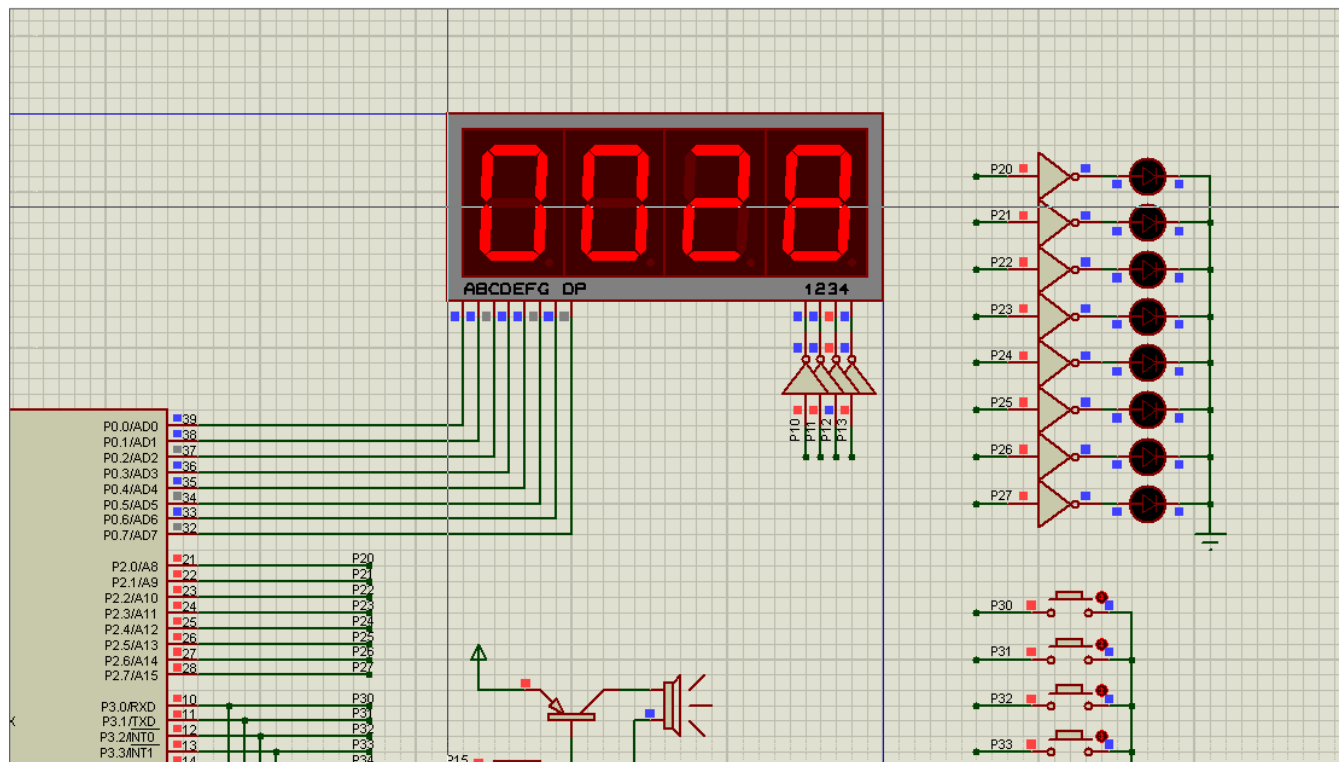
3.3 Kết quả thực hiện



Hình 3: Mạch mô phỏng trên Proteus (trái).



Hình 4: Mạch mô phỏng trên Proteus (phải).



Hình 5: Kết quả chạy chương trình trên Proteus.