TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HCM
KHOA ĐIỆN - ĐIỆN TỬ VIỄN THÔNG

★★★

NGUYỄN MINH TÙNG – MSSV: DV02104

THIẾT KẾ VÀ THI CÔNG HỆ THỐNG GỌI SỐ TỰ ĐỘNG

ĐỒ ÁN TỐT NGHIỆP

GIÁO VIÊN HƯỚNG DẪN

ThS. NGUYỄN THỊ NGỌC ANH

TRƯỜNG ĐẠI HỌC GTVT TP.HCM
THƯ VIỆN
ĐV 15/2007
978

NIÊN KHÓA 2002 – 2007

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TPHCM
KHOA ĐIỆN - ĐIỆN TỬ VIỄN THÔNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập - Tự Do - Hạnh Phúc
- - - o O o - - -

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

(Chú ý: SV phải đóng bản nhiệm vụ này vào trang thứ nhất của đồ án)

Họ và tên SV: NGUYỄN MINH TÙNG MSSV: DV02104

Ngành: ĐTVT Lớp: 24L

1. **Tên đồ án tốt nghiệp:**

Mạch gọi đỗ tự động (QMS)

2. **Nhiệm vụ (yêu cầu về nội dung và số liệu ban đầu):**

- Tìm hiểu vi điều khiển 89C51
- Tìm hiểu giao thức mạng 89C51 giữa các quầy
- Thiết kế và thi công hệ thống gọi số hàng đợi tự động cho 4 quầy

3. **Ngày giao nhiệm vụ đồ án:** 23/04/2007

4. **Ngày hoàn thành nhiệm vụ:** 23/07/2007

5. **Họ tên người hướng dẫn:** **Phần hướng dẫn**

1) Th.s Nguyễn Thị Ngọc Anh
2) .....
3) .....

1) .....
2) .....
3) .....

Nội dung và yêu cầu ĐATN đã được thông qua.
Ngày 02 tháng 08 năm 2007

TRƯỞNG KHOA
(Ký và ghi rõ họ tên)

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ MINH

PGS. TS. Trần Hoài An

NGƯỜI HƯỚNG DẪN CHÍNH
(Ký và ghi rõ họ tên)

Ng. T. Ngọc Anh

LỜI CẢM ƠN

Em xin chân thành cảm ơn quý thầy cô trong khoa Điện – ĐTVT, trường ĐH GTVT TpHCM đã hết lòng dạy dỗ em trong thời gian học tại trường.

Em cũng chân thành cảm cô hướng dẫn – Th.s Nguyễn Thị Ngọc Anh thuộc trường ĐHDL Kỹ Thuật Công Nghệ TpHCM đã tận tình hướng dẫn em thực hiện đề tài này.

Chân thành cảm ơn các bạn sinh viên trong lớp 241 và 242 đã cung cấp tài liệu, sách vở ... để giúp mình hoàn thành luận văn này.

Sinh viên thực hiện

Nguyễn Minh Tùng

MỤC LỤC

Trang

LỜI CẢM ƠN 1
MỤC LỤC 2
LỜI MỞ ĐẦU 3
TÓM TẮT NỘI DUNG ĐỀ TÀI 4
PHẦN MỘT : *CƠ SỞ LÝ THUYẾT* **6**
Chương 1: GIỚI THIỆU VỀ VI ĐIỀU KHIỂN 8051 7
1.1. Họ vi điều khiển X51 8
1.2. Cấu trúc của vi điều khiển 8051 11
1.3. Tổ chức bộ nhớ 15
1.4. Hoạt động của thanh ghi Timer 26
1.5. Hoạt động ngắt 30
1.6. Hoạt động của port nối tiếp 33
Chương 2 : SƠ LƯỢC VỀ GIAO TIẾP MÁY TÍNH 39
2.1. Giao tiếp nối tiếp 39
2.2. Giao tiếp song song 42
2.3. Giới thiệu sơ lược về IC truyền dữ liệu SN75176B 45
Chương 3 : GIAO THỨC NỐI MẠNG I^2C 46
3.1. Khái quát về giao thức I^2C 46
3.2. Sơ lược về Visual Basic 6.0 49
PHẦN HAI : *TÍNH TOÁN VÀ THIẾT KẾ* **51**
Chương 1 : SƠ ĐỒ KHỐI HỆ THỐNG 52
Chương 2 : TÍNH TOÁN, THIẾT KẾ VÀ CHỨC NĂNG TỪNG KHỐI 54
2.1. Bộ phận quản lý trung tâm 54
2.2. Khối hiển thị chính 55
2.3. Khối quầy 57
2.4. Khối giao tiếp máy tính 60
Chương 3 : SƠ ĐỒ NGUYÊN LÝ TOÀN HỆ THỐNG QMS 62
3.1. Sơ đồ khối 62
3.2. Hoạt động của mạch 63
Chương 4 : LƯU ĐỒ THUẬT TOÁN 64
4.1. Khối quầy 64
4.2. Khối xủ lý trung tâm 65
Chương 5 : KẾT LUẬN 66
5.1. Ưu khuyết điểm của đề tài 66
5.2. Hướng phát triển đề tài 66
Tài liệu tham khảo
PHẦN BA : *PHỤ LỤC* **67**
Chương trình

LỜI MỞ ĐẦU

Ngày nay, khi đất nước ta đã gia nhập tổ chức thương mại thế giới (WTO) và nền kinh tế phát triển theo xu hướng toàn cầu hoá thì các thủ tục (đặc biệt là thủ tục hành chính) và dịch vụ phải nhanh, gọn và hiệu quả để thu hút đầu tư trong và ngoài nước, góp phần phát triển kinh tế xã hội của đất nước .

Một khi kinh tế xã hội phát triển, nhu cầu cuộc sống tăng cao thì con người càng trở nên bận rộn với công việc hàng ngày của họ vì vậy việc mất thời gian khi làm các thủ tục hành chính; khám chữa bệnh; thanh toán phí các dịch vụ như: điện thoại, điện lực . . . đã trở thành một vấn đề phiền toái với chúng ta.

Từ thực trạng trên, em đã chọn và thực hiện đề tài *hệ thống gọi số tự động QMS (Queue Management System),* còn gọi là hệ thống quản lý hàng đợi. Sự ra đời của hệ thống này đã giảm thiểu thời gian chờ đợi của người dân tại các bệnh viện, cơ quan hành chính, bưu điện . . .thay vì phải chen lấn nhau chờ đợi, gây mất trật tự nơi công cộng, giờ đây người dân chỉ phải mất vài phút (nhiều nhất là 5 phút) để thực hiện các dịch vụ. Bên cạnh đó, hệ thống còn giúp hạn chế nạn tham ô, quan liêu, cửa quyền làm ảnh hưởng đến sự phát triển của đất nước.

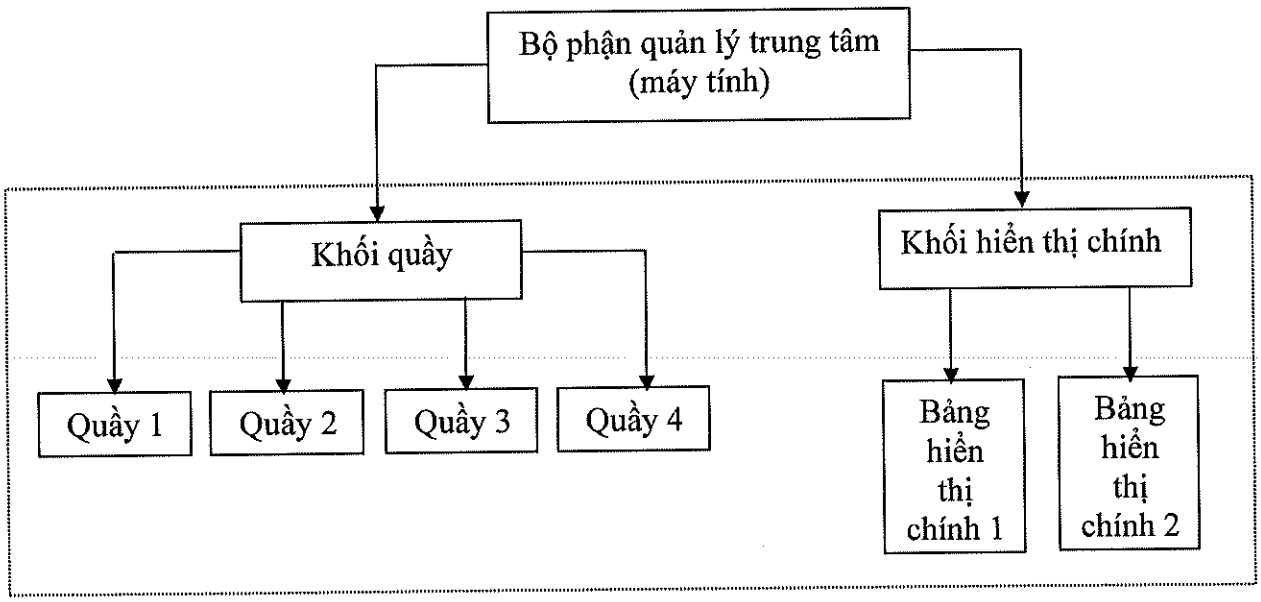
Tuy nhiên, vì thời gian và kiến thức còn hạn chế nên trong quá trình thực hiện đề tài không tránh khỏi sai sót. Rất mong nhận được sự góp ý của thầy cô và các bạn.

TÓM TẮT NỘI DUNG ĐỀ TÀI

Hệ thống gọi số tự động QMS có tác dụng giảm thiểu thời gian chờ đợi khi thực hiện giao dịch tại bệnh viện, cơ quan hành chính, công ty điện lực . . . Để làm được hệ thống này cần thiết kế hai bộ phận chính là bộ phận quản lý và bộ phận giao dịch khách hàng.

- Bộ phận quản lý: dùng máy tính làm trung tâm điều khiển với giao diện được thiết kế bằng Visual Basic để quản lý cơ sở dữ liệu của khách hàng .
- Bộ phận giao dịch khách hàng: dùng vi điều khiển để thu, phát và hiển thị số của khách hàng. Bộ phận này gồm hai khối là khối quầy (gồm 4 quầy) và khối hiển thị chính (2 bảng hiển thị chính).

Hai bộ phận này giao tiếp với nhau qua cổng COM của máy tính .



Hình 1: Sơ đồ khối tổng quát hệ thống QMS

Hệ thống hoạt động như sau:

Khi khách hàng thực hiện giao dịch ở nơi có sử dụng hệ thống QMS sẽ được nhân viên cấp cho một phiếu có số thứ tự và ngồi chờ hệ thống gọi số của mình để thực hiện giao dịch. Các số thứ tự này được máy tính quản lý và cập nhật, máy tính giao tiếp với khối quầy và khối hiển thị chính qua cổng COM.

Khối quầy có 4 quầy, mỗi quầy có một IC 89C51 là trung tâm xử lý của quầy, có chức năng gửi yêu cầu tới máy tính và nhận dữ liệu từ máy tính. Ngoài ra, mỗi quầy còn có một bảng hiển thị lớn dùng 4 LED 7 đoạn lớn cho khách hàng biết số thứ tự đang giao dịch ở quầy đó, một bảng hiển thị nhỏ dùng 4 LED 7 đoạn nhỏ cho nhân viên quan sát, bảng hiển thị nhỏ này được gắn chung với hai phím nhấn và một LED đơn.

Khối hiển thị chính có 2 bảng hiển thị chính dùng để hiển thị số thứ tự và số quầy của 2 khách hàng giao dịch gần nhất. Mỗi bảng hiển thị chính gồm 5 LED 7 đoạn lớn nhằm gây sự chú ý (kèm theo âm thanh phát ra từ loa máy tính), trong đó 4 LED hiển thị số khách hàng và 1 LED hiển thị số quầy khách hàng cần đến. Bảng hiển thị chính cũng có IC 89C51 để thu phát dữ liệu.

PHẦN MỘT

CƠ SỞ LÝ THUYẾT

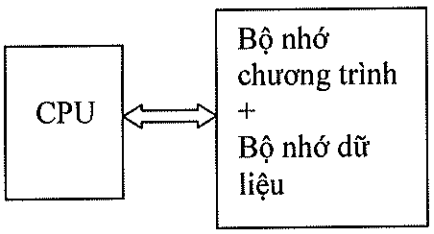
CHƯƠNG 1: GIỚI THIỆU VỀ VI ĐIỀU KHIỂN 8051

Quá trình phát triển nhanh của các loại vi xử lý cho phép ứng dụng chúng trong quá trình sản xuất như các bộ điều khiển. Khác với các loại vi xử lý sử dụng để xử lý thông tin (các máy tính PC xử lý văn bản, cơ sở dữ liệu, tính toán khoa học kỹ that), các vi điều khiển (microcontroller) được thiết kế khi lắp đặt trực tiếp vào ứng dụng cụ thể. Trong đó suốt quá trình vận hành của mình vi điều khiển chỉ thực hiện duy nhất một chức năng. Việc ứng dụng rộng rãi các bộ vi điều khiển dẫn đến yêu cầu sản xuất tất cà các vi mạch trong một vỏ. Việc tích hợp vi mạch trên một vỏ làm giảm giá thành và làm tăng tính tin cậy của vi điều khiển.

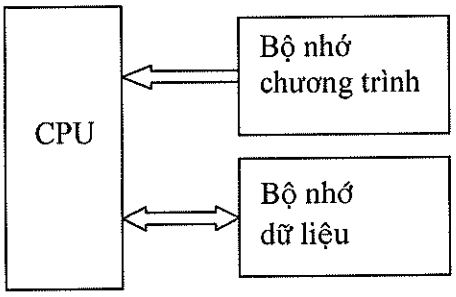
Yêu cầu tăng cao tốc độ xử lý dẫn đến các cấu trúc của vi điều khiển khác với các cấu trúc của các bộ vi xử lý như Z80,8086 và các vi xử lý dùng cho máy tính PC. Cấu trúc của vi điều khiển cho phép truy cậpnhanh bộ nhớ dữ liệu, truy cập và xử lý nhanh các dữ liệu, truy cập nhanh các thiết bị (phép nhân và chia dấu phẩy đứng, các phép toán đại số Boole với 1 bit ...)

Tổng quát ta có thể nói rằng vi điều khiển là một máy tính trong một chip (monolithic microcomputer). Trong đó đã bao gồm CPU, bộ nhớ, các thiết vị vào/ra phổ biến. Cấu trúc của vi điều khiển được thiết kế để phục vụ một mục đích cụ thể mà chỉ cần tối thiểu các mạch phụ trợ. Chức năng của vi điều khiển được xác định bởi chương trình lưu trong bộ nhớ ROM hoặc EPROM. Chương trình được nạp một lần và không thay đổi trong suốt quá trình hoạt động.

Bộ nhớ của các vi điều khiển có cấu trúc khác với bộ nhớ của vi xử lý Z80 cùng loại. Tại các họ vi xử lý ta dùng cấu trúc bộ nhớ Neuman, trong đó bản đồ bộ nhớ dùng chung cho cả chương trình (thường trong ROM – Read only memory) và dữ liệu (RWM – read write memory). Các bộ điều khiển sử dụng cấu trúc Harward. Tại đây bộ nhớ được chia làm hai loại : bộ nhớ chương trình (ROM) và bộ nhớ dữ liệu (RWM).



Hình 2: Cấu trúc Neuman

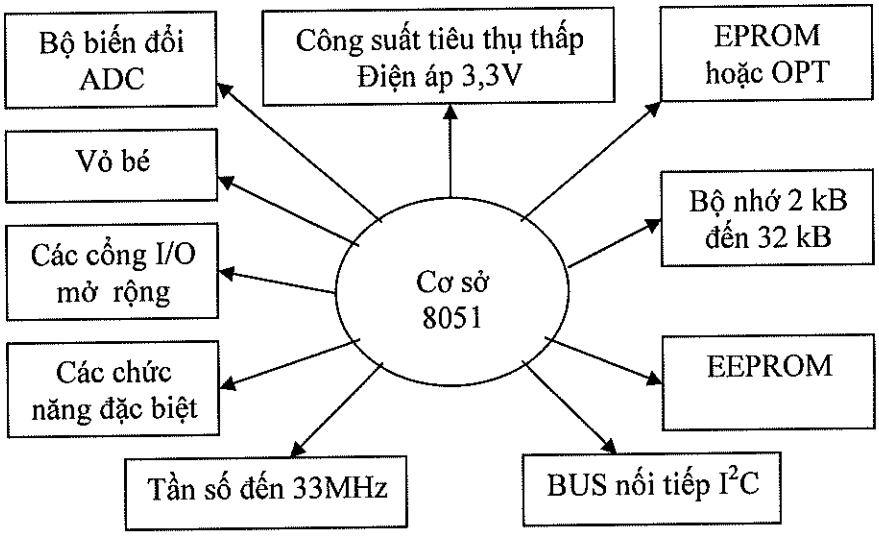


Hình 3: Cấu trúc Harward

Cấu trúc không đổi của các loại vi điều khiển lại là giới hạn đối với việc áp dụng trong thực tế. Tuy vậy trên thị trường có rất nhiều loại vi điều khiển cho các loại ứng dụng điều khiển khác nhau, mỗi loại tự phân biệt bằng lượng tối thiểu thiết bị được tích hợp bên trong.

1.1. HỌ VI ĐIỀU KHIỂN X51

Vi điều khiển 8051 ra đời vào năm 1980 và đã là loại vi điều khiển tương đối "lớn tuổi". Tuy vậy 8051 đã nhận được sự mến chuộng đặt biệt của các nhà thiết kế hệ thống đến nỗi hiện nay tồn tại rất nhiều nhà sản xuất tập trung phát triển các sản phẩm trên cơ sở 8051 và phát triển thêm các thiết bị ngoại vi khác. Ví dụ chỉ riêng Phillips đã sản xuất 24 loại vi điều khiển trên cơ sở 8051 như được thể hiện trên hình 1-1. Các loại vi điều khiển này khác như sau : bộ nhớ chương trình từ 2kB đến 32 kB, bộ nhớ này có loại có thể nạp 1 lần hoặc nhiều lần (EPROM), bộ nhớ EEPROM để nhớ các hằng số, bộ nhớ dữ liệu mở rộng đến 256 kB, bộ ADC 8 hoặc 10 bit với bộ chọn 8 kênh, các cổng vào/ra mở rộng, bộ so sánh hoặc lưu dữ liệu, bus 2 dây I2C. Các nhà sản xuất cung cấp các vi điều khiển có tần số từ 12 MHz đến 33 MHz với loại vỏ DIL, PLCC đến loại nhỏ PQFP. Một số nhà sản xuất tập trung vào các vi điều khiển với các phép tính số học (Siemens 80C537) hoặc việc định thời khác nhau (Dallas DS80C320). Tổng kết các hướng phát triển cụ thể của vi điều khiển trên cơ sở 8051 được tổng kết như trên hình 4 .



Hình 4: Các thành phần mở rộng của họ vi điều khiển trên cơ sở 8051

Bảng tiếp theo giới thiệu một số loại sản phẩm do các hãng Intel, Phillips, Siemens, Atmel sản xuất trên cơ sở của 8051. Ta gọi chung làhọ x51.

Các ký hiệu viết tắt :

WD: có trang bị watchdog

Ca/Co PWM : có trang thiết bị hệ thống lưu capture, so sánh và điều chế độ rộng xung.

Loại	Tần số MHz	ROM	RAM byte	I/O 8 bit	ADC (bit)	Bộ đếm	WD	Kênh nối tiếp	Số mức ngắt	Ca/Co PWM	Vỏ
INTEL											
80C51	33	0	128	4	-	2	-	UART	5/2	-	DIP 40 hoặc PLCC44
80C51	33	4kB	128	4	-	2	-	UART	5/2	-	
87C51	33	E4kB	128	4	-	2	-	UART	5/2	-	

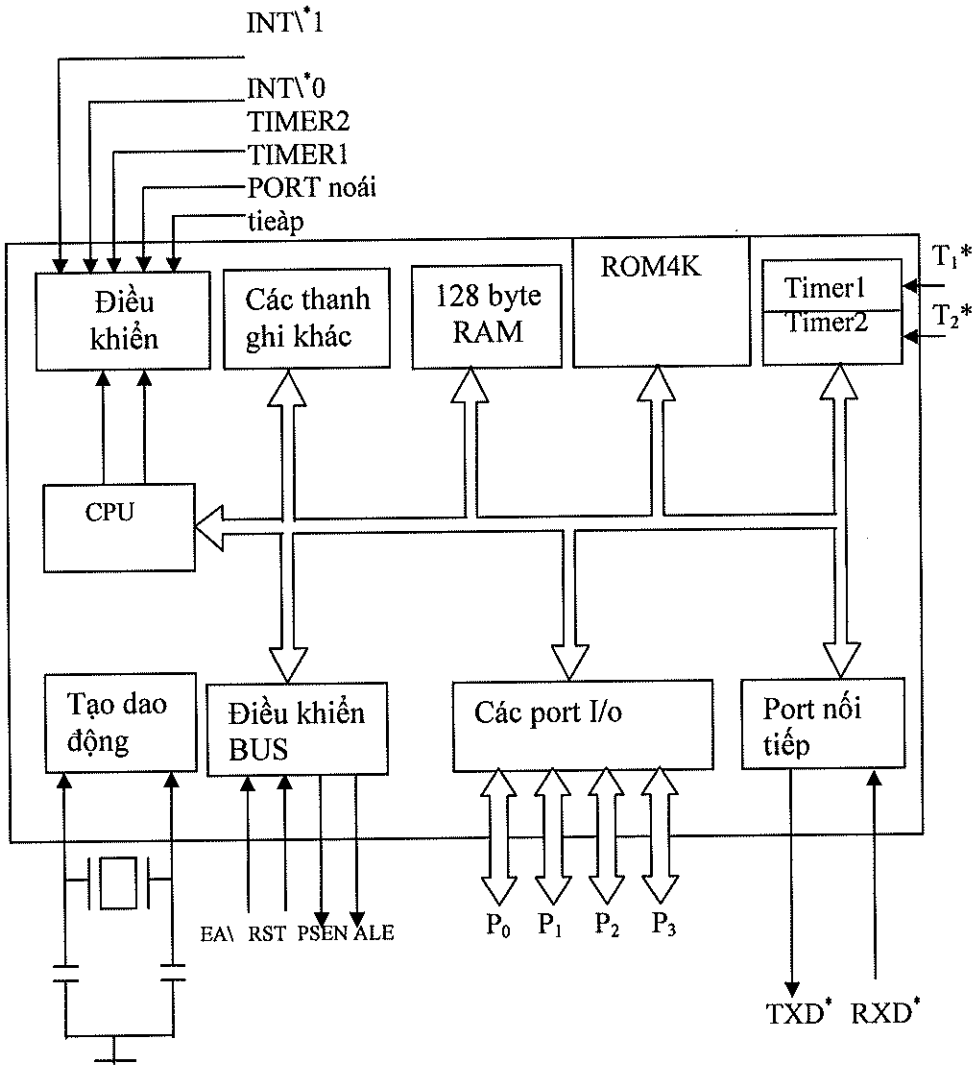
8xC51FA	12/16	8kB	256	4	Cout	3	0+1	UART	7/4	5/5/5	PLCC44
8xC51FB	12/16	16kB	256	4	Cout	3	0+1	UART	7/4	5/5/5	PLCC44
8xC51FC	12/16	32kB	256	4	Cout	3	0+1	UART	7/4	5/5/5	PLCC44
80C251SB	12/16	16kB	1kB	4	Cout	4	0+1	UART	7/4	5/5/5	PLCC44
PHILIPS											
87C550	16	E4kB	128	4	8/8	3	-	UART	6/2	1/0/0	PLCC44
87C654	16/24	E16kI	256	4	-	2	-	UART 12C	6/2	-	PLCC44
83C852		6kB EE2k	256	4	-	2	-	UART 12C	5/2	-	PLCC44
SIEMENS											
SAB80C515 SAB80C535	12/16	8 -	256	6+1	8/8	3	1+1	UART	12/4	4/4/0	PLCC68
SAB80C517 SAB80C537	12/16	8 -	256	7+1,5	12/8	4	2+2	2xUART	14/4	5/21/5	PLCC84 PQFP10(
ATMEL											
AT89C51	0-24	F4kB	256	4	-	2	-	UART	5/2	-	DIP 40 hoặc PLCC44
AT89C52	0-24	F8kB	256	4	Cout	3	-	UART	8/2	-	
AT89C2051	0-24	F2kB	128	1	2xkom	2	-	UART	5/2	-	

1.2. CẤU TRÚC CỦA VI ĐIỀU KHIỂN 8051

Vi điều khiển 8051 có cấu trúc bộ nhớ loại Harward tức là bộ nhớ chương trình và bộ nhớ dữ liệu phân biệt. Vi điều khiển có sơ đồ cấu trúc như trên hình 1-4. Để hoạt động thì vi điều khiển chỉ cần thêm 1 nguồn điện 5V và 1 Xtal bên ngoài. Bên trong vi điều khiển có CPU. CPU nối thông qua các bus bên trong với :

- Bộ nhớ chương trình ROM hoặc EPROM 4 kB.
- Bộ nhớ dữ liệu RAM 128 byte.

- 4 cổng vào ra I/O từ P0 đến P3.
- Nếu sử dụng bộ nhớ ngoài và các thiết bị ngoại vi ngoài thì vi điều khiển có các chân RD, WR (dữ liệu) và PSEN (chương trình) để giao tiếp với bộ nhớ bên ngoài. Bọ nhớ ngoài có thể có dung lượng đến 48 kB.
- Bộ điều khiển ngắt xử lý 5 nguồn yêu cầu ngắt, trong đó 2 nguồn ngoài, 2 bên trong từ 2 bộ đếm và 1 từ kênh nối tiếp. Mức độ ưu tiên của các ngắt được quy định cụ thể.
- 2 bộ đếm 16 bit đếm tần số bên trong vi điều khiển.
- Kênh giao tiếp nối tiếp 2 chiều.
- Bộ xử lý đại số Boole cho phép xử lý từng bit của RAM và các thiết bị ngoại vi.
- Với tần số 12MHz thì mỗi lệnh kéo dài 1μs hoặc 2μs, các lệnh dài hơn như nhân, chia kéo dài 4μs.



Hình 5: Cấu trúc bên trong của 8051

1.2.1. Các chân của 8051

8051 được đặt trong vỏ DIL với 40 chân hoặc gần đây với vỏ loại PLCC với 44 chân.

i. Hai chân dành cho nguồn Ucc=5V và Uss=0V

ii. Hai chân XTAL1 và XTAL2 cần được nối với thạch anh để cho mạch dao động bên trong. Kết nối thạch anh với 8051 có thể thực hiện như trên hình 1-7. Tần số của mạch dao động này chính là tần số xung đồng hồ cho 8051.

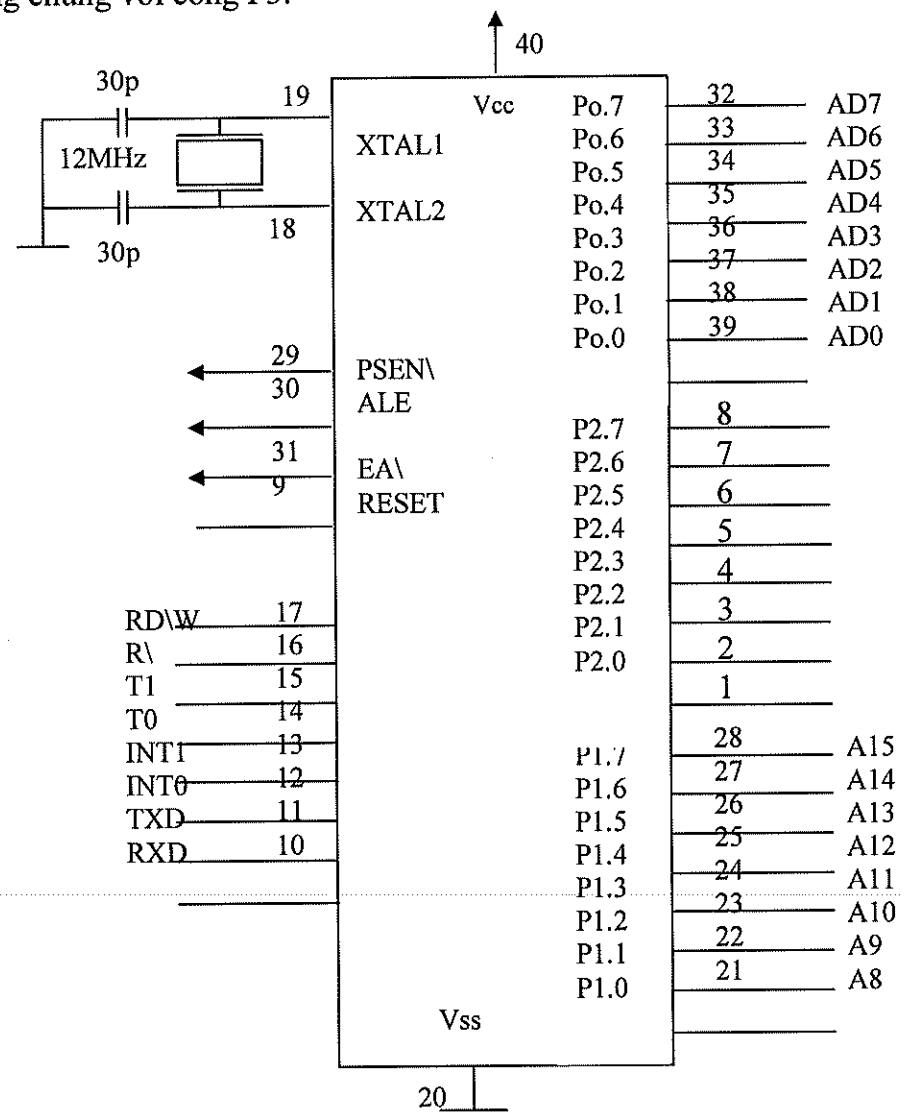
iii. Bốn cổng vào ra từ P0 đến P3 (8bit). Các cổng này còn có thể dùng làm chân ra cho các bus địa chỉ, bus dữ liệu, bus điều khiển khi ta sử dụng bộ nhớ bên ngoài.

iv. Chân Ġ để chọn sử dụng bộ nhớ bên ngoài hay bên trong.

v. Tín hiệu ALE,Ġ ,Ġ vàĠ để truy cập bộ nhớ bên ngoài.

vi. Chân RESET dùng reset 8051.

Các chân khác phục vụ ngắt, truyền thông nối tiếp, bộ đếm. Các chân này được dùng chung với cổng P3.



Hình 6: Các chân của 8051

1.2.2. Chức năng của các chân 8051

Port 0: từ chân 32 đến chân 39 (P0.0 _P0.7). Port 0 có 2 chức năng: trong các thiết kế cỡ nhỏ không dùng bộ nhớ mở rộng nó có chức năng như các đường I/O, đối với thiết kế luôn có bộ nhớ mở rộng nó được kết hợp giữa bus địa chỉ và bus dữ liệu.

Port 1: từ chân 1 đến chân 9 (P1.0 _ P1.7). Port 1 là port IO dùng cho giao tiếp với thiết bị ngoài nếu cần.

Port 2: từ chân 21 đến chân 28 (P2.0 _P2.7). Port 2 là một port có tác dụng kép dùng như các đường xuất nhập hoặc là byte cao của bus địa chỉ đối với các thiết bị dùng bộ nhớ mở rộng.

Port 3: từ chân 10 đến chân 17 (P3.0 _ P3.7). Port 3 là port có tác dụng kép. Các chân của port này có nhiều chức năng, có công dụng chuyển đổi có liên hệ đến các đặc tính đặc biệt của 8051 như ở bảng sau :

Bit	Tên	Chức năng chuyển đổi
P3.0	RXD	Ngõ vào 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 TIMER/ COUNTER thứ 0.
P3.5	T1	Ngõ vào của TIMER/ COUNTER thứ 1.
P3.6	WR\	Tín hiệu ghi dữ liệu lên bộ nhớ ngoài.
P3.7	RD\	Tín hiệu đọc bộ nhớ dữ liệu ngoài.

PSEN (Program store enable):

PSEN là tín hiệu ngõ ra có tác dụng cho phép đọc bộ nhớ chương trình mở rộng và thường được nối đến chân OE\ của Eprom cho phép đọc các byte mã lệnh.

PSEN ở mức thấp trong thời gian 8051 lấy lệnh. Các mã lệnh của chương trình được đọc từ Eprom qua bus dữ liệu, được chốt vào thanh ghi lệnh bên trong 8051 để

giải mã lệnh. Khi 8051 thi hành chương trình trong ROM nội PSEN ở mức cao.

ALE (Address Latch Enable):

Khi 8051 truy xuất bộ nhớ bên ngoài, Port 0 có chức năng là bus địa chỉ và dữ liệu do đó phải tách các đường dữ liệu và địa chỉ. Tín hiệu ra ALE ở chân thứ 30 dùng làm tín hiệu điều khiển để giải đa hợp các đường địa chỉ và dữ liệu khi kết nối chúng với IC chốt.

Tín hiệu ở chân ALE là một xung trong khoảng thời gian port 0 đóng vai trò là địa chỉ thấp nên chốt địa chỉ hoàn toàn tự động.

EA\ (External Access): Tín hiệu vào EA\ ở chân 31 thường được mắc lên mức 1 hoặc mức 0. Nếu ở mức 1, 8051 thi hành chương trình từ ROM nội. Nếu ở mức 0, 8051 thi hành chương trình từ bộ nhớ mở rộng. Chân EA\ được lấy làm chân cấp nguồn 21V khi lập trình cho Eprom trong 8051.

RST (Reset): Khi ngõ vào tín hiệu này đưa lên mức cao ít nhất 2 chu kỳ máy, các thanh ghi bên trong được nạp những giá trị thích hợp để khởi động hệ thống. Khi cấp điện mạch phải tự động reset.

Các ngõ vào bộ dao động X1, X2:Bộ tạo dao động được tích hợp bên trong 8051. Khi sử dụng 8051, người ta chỉ cần nối thêm tụ thạch anh và các tụ. Tần số tụ thạch anh thường là 12 Mh.

1.3. TỔ CHỨC BỘ NHỚ

Bộ nhớ của 8051 được chia làm hai phần : bộ nhớ chương trình và bộ nhớ dữ liệu. Các bộ nhớ này được truy cập bằng các lệnh khác nhau. Bộ nhớ còn có thể chia ra làm bộ nhớ bên trong và bộ nhớ bên ngoài. Bộ nhớ bên ngoài là bộ nhớ ta cần ghép thêm vào cho các chương trình yêu cầu nhiều bộ nhớ hơn.

1.3.1. Bộ nhớ chương trình

Để tránh trường hợp trùng lên nhau của bộ nhớ chương trình bên trong 4 kB và bộ nhớ chương trình bên ngoài 64kB thì vi điều khiển được trang bị chân nhận tín hiệu đầu vào EA. Trên hình 2-1 ta thấy :

- Nếu EA = 0 thì vi điều khiển đọc từ bộ nhớ chương trình bên ngoài.
- Nếu EA = 1 thì vi điều khiển đọc từ bộ nhớ chương trình bên trong từ địa chỉ 0000H đến 0FFFH, còn với địa chỉ từ 1000H đến FFFFH thì đọc từ bộ nhớ chường trình ngoài.

1.3.2. Bộ nhớ dữ liệu

➢ Bộ nhớ dữ liệu bên trong có độ dài đến 28 byte. Bộ nhớ dữ liệu bên ngoài có độ dài đến 216 byte.

➢ Vi điều khiển truy cập các loại bộ nhớ dữ liệu bên trong hoặc bộ nhớ dữ liệu bên ngoài bằng các lệnh khác nhau. Ví dụ với bộ nhớ dữ liệu bên trong có lệnh MOV (dịch chuyển), lệnh tương ứng với bộ nhớ dữ liệu bên ngoài là MOVC.

Một số vi điều khiển có được trang thiết bị bộ nhớ dữ liệu mở rộng bên trong tương đương với vùng các thanh ghi chức năng đặc biệt. Bộ nhớ này có thể truy cập bằng định vị địa chỉ gián tiếp.

1.3.3. Tổ chức bộ nhớ RAM bên trong

Việc phân biệt bộ nhớ dữ liệu bên ngoài và bên trong được phân biệt bằng cách sử dụng các lệnh khác nhau. Trên hình 2-2 là bản đồ phân bố bộ nhớ dữ liệu bên trong và các thanh ghi đặc biệt.

128 byte đầu tiên (từ 00H đến 7FH) của bộ nhớ RAM bên bao gồm theo thứ tự :

- ➢ 4 dãy (bank 0, bank 1, bank 2, bank 3), mỗi dãy gồm 8 thanh ghi từ R0, R1 đến R7. Các dãy này có địa chỉ từ 00H đến 1FH. Việc truy cập các thanh ghi R0-R7 là truy cập trực tiếp. Ta có đến 4 dãy tức là có 4 bộ R0-R7. việc chon bank là do lời chế độ PSW quyết định. Ta sẽ đề cập điều này ở lời chế độ.
- ➢ 16 byte tiếp theo (20H-2FH) là vùng của các bit định vị được. Bit thấp nhất của byte có địa chỉ thấp nhất (bit b0 ở địa chỉ 20H) có địa chỉ là 00H. sau đó địa chỉ tiếp tục tăng dần cho đến bit b7 của byte với địa chỉ 2FH là bit cao nhất, bit này có địa chỉ 127 (7FH). Tổng số bit có thể định vị trực tiếp của vùng này là 128.
- ➢ Các byte từ 30H đến 7FH là các thanh ghi có thể truy cập tự do bằng địa chỉ trực

tiếp hoặc gián tiếp.

128 địa chỉ tiếp theo (từ 80H đến FFH) dùng cho các thanh ghi đặc biệt. Nếu vi điều khiển được trang bị thêm bộ nhớ bên trong vùng địa chỉ 80H-FFH thì các địa chỉ này chỉ có thể truy cập gián tiếp.

7F **RAM ĐA DỤNG** 30								
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40

27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00

1F BANK 3 18
17 BANK 2 10
0F BANK 1 08

07 Bank thanh ghi 0 (mặc định cho R0-R7) 00								
CẤU TRÚC RAM NỘI								
F0	F7	F6	F5	F4	F3	F2	F1	F0
E0	E7	E6	E5	E4	E3	E2	E1	E0
D0	D7	D6	6D	6C	6B	6A	69	68
B8	-	-	-	BC	BB	BA	B9	B8
B0	B7	B6	B5	B4	B3	B2	B1	B0
A8	AF	AE	AD	AC	AB	AA	A9	A8
A0	A7	A6	A5	A4	A3	A2	A1	A0
99	Không có địa chỉ hóa từng bit							

98	9F	9E	9D	9C	9B	9A	99	98
90	97	96	95	94	93	92	91	90
8D	Không được địa chỉ hóa từng bit							
8C	Không được địa chỉ hóa từng bit							
8B	Không được địa chỉ hóa từng bit							
8A	Không được địa chỉ hóa từng bit							
89	Không được địa chỉ hóa từng bit							
88	8F	8E	8D	8C	8B	8A	89	88
87	Không được địa chỉ hóa từng bit							
83	Không được địa chỉ hóa từng bit							
82	Không được địa chỉ hóa từng bit							
81	Không được địa chỉ hóa từng bit							
80	87	86	85	84	83	82	81	80
THANH GHI CHỨC NĂNG ĐẶC BIỆT								

Ram đa dụng:

Mọi địa chỉ trong vùng ram đa dụng đều có thể được truy xuất tự do dùng kiểu địa chỉ trực tiếp hay gián tiếp. Ví dụ để đọc nội dung ô nhớ ở địa chỉ 5FH của ram nội vào thanh ghi tích lũy A : MOV A,5FH.

Hoặc truy xuất dùng cách địa chỉ gián tiếp qua R0 hay R1. Ví dụ 2 lệnh sau sẽ thi hành

cùng nhiệm vụ như lệnh ở trên:

MOV R0, #5FH

MOV A , @R0

Ram có thể truy xuất từng bit:

8051 chứa 210 bit được địa chỉ hóa từng bit, trong đó 128 bit chứa ở các byte có địa chỉ từ 20H đến 2FH, các bit còn lại chứa trong nhóm thanh ghi chức năng đặc biệt.

Ýtưởng truy xuất từng bit bằng phần mềm là một đặc tính mạnh của vi điều khiển nói chung. Các bit có thể được đặt, xóa, and, or,... với 1 lệnh đơn. Ngoài ra các port cũng có thể truy xuất được từng bít làm đơn giản phần mềm xuất nhập từng bit.

Ví dụ để đặt bit 67H ta dùng lệnh sau: SETB 67H.

Các bank thanh ghi:

Bộ lệnh 8051 hỗ trợ 8 thanh ghi có tên là R0 đến R7 và theo mặc định (sau khi reset hệ thống), các thanh ghi nàyở các địa chỉ 00H đến 07H. lệnh sau đây sẽ đọc nội dung ở địa chỉ 05H vào thanh ghi tích lũy: MOV A, R5.

Đây là lệnh 1 byte dùng địa chỉ thanh ghi. Tuy nhiên có thể thi hành bằng lệnh 2 byte dùng địa chỉ trực tiếp nằm trong byte thứ 2: MOV A, 05H.

Lệnh dùng các thanh ghi R0 đến R7 thì sẽ ngắn hơn và nhanh hơn nhiều so với lệnh tương ứng dùng địa chỉ trực tiếp.

Bank thanh ghi tích cực bằng cách thay đổi các bit trong từ trạng thái chương trình

(PSW). Giả sủ thanh ghi thứ 3 đang được truy xuất, lệnh sau đây sẽ di chuyển nội dung của thanh ghi A vào ô nhớ ram có địa chỉ 18H: MOV R0, A.

1.3.4. Các thanh ghi có chức năng đặc biệt

8051 có 21 thanh ghi chức năng đặc biệt (SFR: Special Funtion Register) ở vùng trên của RAM nội từ địa chỉ 80H đến FFH.

Chú ý: tất cả 128 địa chỉ từ 80H đến FFH không được định nghĩa, chỉ có 21

thanh ghi chức năng đặc biệt được định nghĩa sẵn các địa chỉ.

Thanh ghi trạng thái chương trình:

Thanh ghi trạng thái chương trình PSW (Program Status Word) ở địa chỉ DOH chứa các bít trạng thái như bảng sau:

Bit	Ký hiệu	Địa chỉ	Ý nghĩa
PSW.7	CY	D7H	Cờ nhớ
PSW.6	AC	D6H	Cờ nhớ phụ
PSW.5	F0	D5H	Cờ 0
PSW.4	RS1	D4H	Bit 1 chọn bank thanh ghi
PSW.3	RS0	D3H	Bit 0 chọn bank thanh ghi 00=bank 0: địa chỉ 00H – 07H 01=bank 1: địa chỉ 08H – 0FH 10=bank 2: địa chỉ 10H – 1FH 11=bank 3: địa chỉ 18H –1FH
PSW.2	0V	D2H	Cờ tràn
PSW.1	_	D1H	Dự trữ
PSW.0	P	D0H	Cờ parity chẵn lẽ.

- *Cờ nhớ:*C = 1 nếu phép toán cộng có tràn hoặc phép toán trừ có mượn và ngược lại C = 0. Ví dụ nếu thanh ghi A có giá trị FF thì lệnh sau:

  ADD A, #1

Phép cộng này có tràn nên bit C = 1 và kết quả trong thanh ghi A = 00H

Cờ nhớ có thể xem là thanh ghi 1 bit cho các lệnh luận lý thi hành trên bit.

ANL C, 25H

- *Cớ nhớ phụ:* Khi cộng các số BCD, cờ nhớ phụ AC = 1 nếu kết quả 4 bit thấp trong khoảng 0AH đến 0FH. Ngược lại AC = 0.
- *Cờ 0:* Cờ 0 là một bit cờ đa dụng dành cho các ứng dụng của người dùng.

Các bit chọn Bank thanh ghi truy xuất:

Các bit chọn bank thanh ghi (RS0 và RS1) xác định bank thanh ghi được truy xuất. Chúng được xóa sau khi reset hệ thống và được thay đổi bằng phần mềm nếu cần. Ví dụ lệnh sau cho phép bank thanh ghi 3 và di chuyển nội dung của bank thanh ghi R7 (địa chỉ bye 1FH) vào thanh ghi A:

```
SETB   RS1
SETB   RS0
MOV    A,R7
```

Thanh ghi B:

Thanh ghi B ở địa chỉ F0H được dùng cùng với thanh ghi tích lũy A cho các phép toán nhân và chia. Lệnh MUL AB sẽ nhân các giá trị không dấu 8 bit trong A và B rồi trả kết quả về 16 bit trong A (byte thấp) và B (byte cao). Lệnh DIV AB sẽ chia A cho B rồi trả kết quả nguyên trong A và phần dư trong B. thanh ghi cũng có thể xem như thanh ghi đệm đa dụng.

Con trỏ ngăn xếp:

Con trỏ ngăn xếp SP là một thanh ghi 8 bit ở địa chỉ 18H. Nó chứa địa chỉ của byte dữ liệu hiện hành trên đỉnh của ngăn xếp. Các lệnh trên ngăn xếp bao gồm các lệnh cất dữ liệu vào ngăn xếp và lấy dữ liệu ra khỏi ngăn xếp. Lệnh cất dữ liệu vào ngăn xếp sẽ làm tăng SP trước khi ghi dữ liệu và lệnh lấy dữ liệu ra khỏi ngăn xếp sẽ làm giảm SP. Ngăn xếp của 8051 được giữ trong ram nội và giới hạn các địa chỉ có thể truy xuất bằng địa chỉ gián tiếp, chúng là 128 byte đầu của 8051

Để khởi động SP với ngăn xếp bắt đầu tại địa chỉ 60 H, các lệnh sau đây được dùng:

```
MOV   SP,#5FH
```

Khi reset 8051, SP sẽ mang giá trị mặc định là 07H và dữ liệu đầu tiên sẽ được

cất vào ô nhớ ngăn xếp có địa chỉ là 08 H. Ngăn xếp được truy xuất trực tiếp bằng các lệnh PUSH và POP để lưu trữ tạm thời và lấy lại dữ liệu hoặc truy xuất ngầm bằng lệnh gọi chương trình con ACALL,LCALL và các lệnh trở về (RET. RETI) để lưu trữ giá trị của bộ đếm chương trình khi bắt đầu thực hiện chương trình con và lấy lại khi kết thúc chương trình con.

Con trỏ dữ liệu:con trỏ dữ liệu DPTR được dùng để truy xuất bộ nhớ ngoài là một thanh ghi 16 bit ở địa chỉ 82H (DPL: byte thấp) và 83H (DPH: byte cao). 3 lệnh sau sẽ ghi 55H vào ram ngoài ở địa chỉ 1000H:

```
MOV  A,#55H
MOV   DPTR, #1000
MOVX @DPTR,A
```

Các thanh ghi port xuất nhập:

Các port của 8051 bao gồm port 0 ở địa chỉ 80H, port 1 ở địa chỉ 90H, port 2 ở địa chỉ A0H, và port3 ở địa chỉ B0H. tất cả các port này đều có thể truy xuất từng bit nên rất thuận tiện trong khả năng giao tiếp.

Các thanh ghi timer:

8051 có chứa 2 bộ định thời/ đếm 16 bit được dùng cho việc định thời hoặc đếm sự kiện. Timer 0 ở địa chỉ 8AH (TL0: byte thấp) và 8CH (TH0: byte cao). Timer 1 ở địa chỉ 8BH (TL1: byte thấp) và 8DH (TH1: byte cao). Việc khởi động timer được Set bởi Timer Mode (TMOD) ở địa chỉ 89H và thanh ghi điều khiển timer (TCON) ở địa chỉ 88H, chỉ có TCON được địa chỉ hóa từng bit.

Các thanh ghi port nối tiếp:

8051 chứa một port nối tiếp dành cho việc trao đổi thông tin với các thiết bị nối tiếp như máy tính, modem hoặc giao tiếp nối tiếp với các IC khác. Một thanh ghi gọi là bộ đệm dữ liệu nối tiếp (SBUF) ở địa chỉ 99H sẽ giữ cả 2 dữ liệu truyền và dữ liệu nhận. Khi truyền dữ liệu thì ghi lên SBUF, khi nhận dữ liệu thì đọc SBUF. Các mode vận hành khác nhau được lập trình qua thanh ghi điều khiển port nối tiếp SCON ở địa

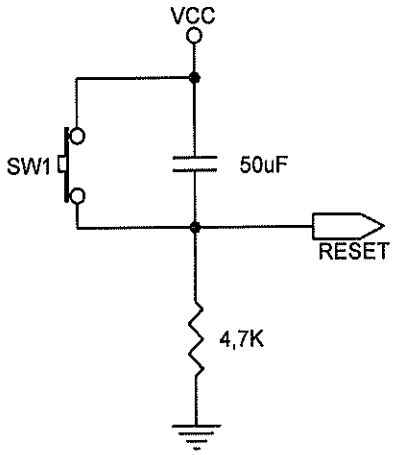
chỉ 98H.

Các thanh ghi ngắt:

8051 có cấu trúc 5 nguồn ngắt, 2 mức ưu tiên. Các ngắt bị cấm sau khi reset hệ thống và sẽ được cho phép bằng việc ghi thanh ghi cho phép ngắt (IE) ở địa chỉ A8H, cả 2 thanh ghi được địa chỉ hóa từng bit.

Thanh ghi điều khiển nguồn điện:Thanh ghi điều khiển công suất (PCON) ở địa chỉ 87H chứa các bit điều khiển.

Tín hiệu Reset



8051 có ngõ vào reset RST tác động ở mức cao trong khoảng thời gian 2 chu kỳ, sau đó xuống mức thấp để 8051 bắt đầu làm việc. RST có thể kích bằng tay bằng một phím nhấn thường mở, sơ đồ mạch reset như hình trên .

Sau khi reset hệ thống được tóm tắt như sau:

Thanh ghi	Nội dung
Đếm chương trình PC	0000H
Thanhghi tích lũy A	00H
Thanh ghi B	00H
Thanh ghi trạng thái	00H
SP	07H
DPTR	0000H
Port 0 đến Port 3	FFH
IP	XXX0000 B
IE	0XX00000 B
Các thanh ghi định thời	00H

Thanh ghi quan trọng nhất là thanh ghi bộ đếm chương trình PC được Reset tại địa chỉ 0000H. Khi ngõ vào RST xuống mức thấp, chương trình luôn bắt đầu tại địa chỉ 0000H của bộ nhớ chương trình. Nội dung của Ram trong chip không bị hay đổi bởi tác động của ngõ vào Reset.

HOẠT ĐỘNG CỦA THANH GHI TIMER

8051 có hai 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 8051.

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 ngõra. 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).

Truy xuất các timer của 8051 dùng sáu thanh ghi chức năng đặc biệt cho trong bảng sau:

SFR	Mục Đích	Địa chỉ	Địa chỉ hóa từng bit
TCON	Điều khiển Timer	88H	Có
TMOD	Chế độ Timer	89H	Không
TL0	Byte thấp của Timer 0	90H	Không
TL1	Byte thấp của Timer 1	91H	Không
TH0	Byte cao của Timer 0	92H	Không
TH1	Byte cao của Timer 1	93H	Không

Các thanh ghi chức năng của timer trong 8051.

Thanh ghi chế độ timer (TMOD)

Thanh ghi TMOD chứa hai nhóm 4 bit dùng để đặt chế độ làm việc cho Timer 0, và Timer 1.

Bit	Tên	Timer	Mô tả
7	GATE	1	Bit mở cổng, khi lên 1 timer chỉ chạy khi INT1 ở mức cao
6	C/T	1	Bit chọn chế độ Count/Timer 1 = bộ đếm sự kiện 0 = bộ định khoảng thời gian

5	M1	1	Bit 1 của chế độ mode
4	M0	1	Bit 0 của chế độ mode
3	GATE	0	Bit mở cổng, khi lên 1 timer chỉ chạy khi INT0 ở mức cao
2	C/T	0	Bit chọn chế độ Count/Timer
1	M1	0	Bit 1 của chế độ mode
0	M0	0	Bit 0 của chế độ mode
Tóm tắt thanh ghi chức năng TMOD.			

Thanh ghi điều khiển timer(TCON)

Thanh ghi TCON chứa các bit trạng thái và các bit điều khiển cho Timer 1, Timer 0.

Bit	Ký hiệu	Địa chỉ	Mô tả
TCON.7	TF1	8FH	Cờ báo tràn timer 1. Đặt bởi phần cứng khi tràn, được xóa bởi phần mềm, hoặc phần cứng khi bộ xử lý chỉ đến chương trình phục vụ ngắt.
TCON.6	TR1	8EH	Bit điều khiển timer 1 chạy đặt xóa bằng phần mềm để cho timer chạy ngưng.
TCON.5	TF0	8DH	Cờ báo tràn Timer 0.
TCON.4	TR0	8CH	Bit điều khiển Timer 0 chạy

TCON.3	IE1	8BH	Cờ cạnh ngắt 1 bên ngoài. Đặt bởi phần cứng khi phát hiện một cạnh xuống ở INT1 xóa bằng phần mềm hoặc phần cứng khi CPU chỉ đến chương trình phục vụ ngắt.
TCON.2	IT1	8AH	Cờ kiểu ngắt 1 bên ngoài. Đặt xóa bằng phần mềm để ngắt ngoài tích cực cạnh xuống /mức thấp.
TCON.1	IE0	89H	Cờ cạnh ngắt 0 bên ngòai
TCON.0	IT0	88h	Cờ kiểu ngắt 0 bên ngoài
Tóm tắt thanh ghi chức năng TCON			

Khởi động và truy xuất thanh ghi timer:

Thông thường các thanh ghi được khởi động một lần đầu ở chương trình để đặt ở chế độ làm việc đúng. Sau đó, trong thân chương trình, các thanh ghi timer được cho chạy, dừng, các bit được kiểm tra và xóa, các thanh ghi timer được đọc và cập nhật.... theo đòi hỏi các ứng dụng.

TMOD là thanh ghi thứ nhất được khởi động vì nó đặt chế độ hoạt động. Ví dụ, các lệnh sau khởi động Timer 1 như timer 16 bit (chế độ 1) có xung nhịp từ bộ dao động tên chip cho việc định khoảng thời gian:

MOV TMOD, #1B

Lệnh này sẽ đặt M1 = 1 và M0 = 0 cho chế độ 1, C/ T= 0 và GATE = 0 cho xung nhịp nội và xóa các bit chế độ Timer 0. Dĩ nhiên, timer không thật sự bắt đầu định thời cho đến khi bit điều khiển chạy TR1 được đặt lên 1.

Nếu cần số đếm ban đầu, các thanh ghi TL1/TH1 cũng phải được khởi động.

Một khoảng 100s có thể được khởi động bằng cách khởi động giá trị cho TH1/TL1 là FF9CH:

```
MOV  TL1, #9CH
MOV  TH1, #0FFH
```

Rồi timer được cho chạy bằng cách đặt bit điều khiển chạy như sau:

```
SETB  TR1
```

Cờ báo tràn được tự động đặt lên 1 sau 100(s. Phần mềm có thể đợi trong 100 (s bằng cách dùng lệnh rẽ nhánh có điều kiện nhảy đến chính nó trong khi cờ báo tràn chưa được đặt lên 1:

```
WAIT:  JNB  TF1, WAIT
```

Khi timer tràn, cần dừng timer và xóa cờ báo tràn trong phần mềm:

```
CLR  TR1
CLR  TF1
```

HOẠT ĐỘNG NGẮT (INTERRUPT)

Một ngắt là sự xảy ra một điều kiện, một sự kiện mà nó gây ra treo tạm thời 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 8051

Có 5 nguồn ngắt ở 8031: 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.

Khi có hai hoặc nhiều ngắt đồng thời, hoặc một ngắt xảy ra khi một ngắt khác đang được phục vụ, có cả hai sự tuần tự hỏi vòng và sơ đồ ưu tiên hai mức dùng để xác định việc thực hiện các ngắt. Việc hỏi vòng tuần tự thì cố định nhưng ưu tiên ngắt thì

có thể lập trình được.

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	EX0	A8H	Cho phép ngắt ngoài 0
Tóm tắt thanh ghi IE			

Các cờ ngắt

Khi điều kiện ngắt xảy ra thì ứng với từng loại ngắt mà loại cờ đó được đặt lên một để xác nhận ngắt.

Ngắt	Cờ	Thanh ghi SFR và vị trí bit
Bên ngoài 0	IE0	TCON.1
Bên ngoài 1	IE1	TCON.3
Timer 1	TF1	TCON.7
Timer 0	TF0	TCON.5
Port nối tiếp	TI	SCON.1
Port nối tiếp	RI	SCON.0
Các loại cờ ngắt		

Các vectơ ngắt

Khi chấp nhận ngắt, giá trị được nạp vào PC được gọi là vector ngắt. Nó là địa chỉ bắt đầu của ISR cho nguồn tạo ngắt, các vector ngắt được cho ở bảng sau :

Ngắt	Cờ	Địa chỉ vector
Reset hệ thống	RST	0000H
Bên ngoài 0	IE0	0003H
Timer 0	TF0	000BH

Bên ngoài 1	IE1	0013H
Timer 1	TF1	001BH
Port nối tiếp	TI và RI	0023H
Timer 2		002BH

Vector reset hệ thống (RST ở địa chỉ 0000H) được để trong bảng này vì theo nghĩa này, nó giống ngắt : nó ngắt chương trình chính và nạp cho PC giá trị mới.

HOẠT ĐỘNG CỦA PORT NỐI TIẾP

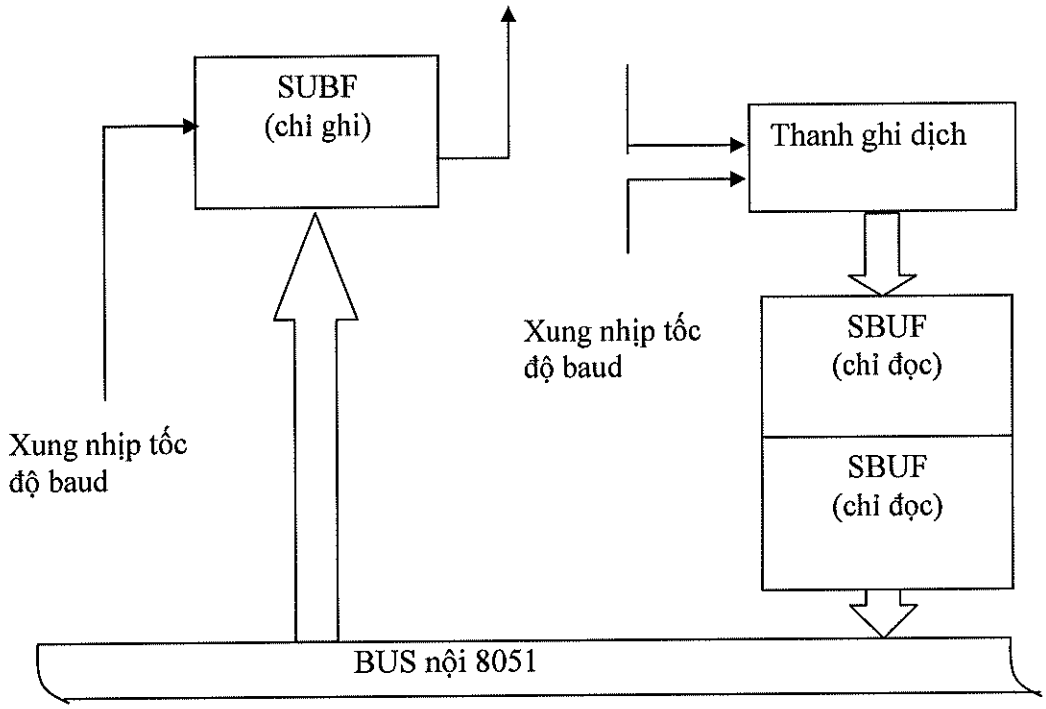
8051 có một port nối tiếp trong chip có thể hoạt động ở nhiều chế độ khác trên một dãy tần số rộng. Chức năng chủ yếu của một port nối tiếp là thực hiện chuyển đổi song song sang nối tiếp với dữ liệu xuất và chuyển đổi nối tiếp sang song song với dữ liệu nhập.

Truy xuất phần cứng đến port nối tiếp qua các chân TXD và RXD. Các chân này có các chức năng khác với hai bit của P3.1 ở chân 11 (TXD) và P3.0 ở chân 10 (RXD).

Port nối tiếp cho hoạt động song công (full duplex : thu và phát đồng thời) và đệm lúc thu (receiver buffering) cho phép một ký tự sẽ được thu và được giữ trong khi ký tự thứ hai được nhận. Nếu CPU đọc ký tự thứ nhất trước khi ký tự thứ hai được thu đầy đủ thì dữ liệu sẽ không bị mất.

Hai thanh ghi chức năng đặc biệt cho phép phần mềm truy xuất đến port nối tiếp là: SBUF và SCON. Bộ đếm port nối tiếp (SBUF) ở địa chỉ 99H thật sự là hai bộ đếm. Viết vào SBUF để truy xuất dữ liệu thu được. Đây là hai thanh ghi riêng biệt thanh ghi chỉ ghi để phát và thanh ghi để thu.

TXD (P3.1) RXD (P3.0)



Hình 7: Sơ đồ port nối tiếp

Thanh ghi điều khiển port nối tiếp (SCON) ở địa chỉ 98H là thanh ghi có địa chỉ bit chứa các bit trạng thái và các bit điều khiển. Các bit điều khiển đặt chế độ hoạt động cho port nối tiếp, và các bit trạng thái báo cáo kết thúc việc phát hoặc thu ký tự. Các bit trạng thái có thể được kiểm tra bằng phần mềm hoặc có thể được lập trình để tạo ngắt.

Tần số làm việc của port nối tiếp còn gọi là tốc độ baund có thể cố định (lấy từ bộ dao động của chip). Nếu sử dụng tốc độ baud thay đổi, timer 1 sẽ cung cấp xung nhịp tốc độ baud và phải được lập trình.

Thanh ghi điều khiển port nối tiếp

Chế độ hoạt động của port nối tiếp được đặt bằng cách ghi vào thanh ghi chế độ port nối tiếp (SCON) ở địa chỉ 98H. Sau đây các bảng tóm tắt thanh ghi SCON và các

chế độ của port nối tiếp :

Bit	**Ký hiệu**	**Địa chỉ**	**Mô tả**
SCON.7	SM0	9FH	Bit 0 của chế độ port nối tiếp
SCON.6	SM1	9EH	Bit 1 của chế độ port nối tiếp
SCON.5	SM2	9DH	Bit 2 của chế độ 2 nối tiếp. Cho phép truyền thông xử lý trong các chế độ 2 và 3, RI sẽ không bị tác động nếu bit thứ 9 thu được là 0
SCON.4	REN	9CH	Cho phép bộ thu phải đặt lên 1 để thu (nhận) các ký tự
SCON.3	TB8	9BH	Bit 8 phát, bit thứ 9 được phát các chế độ 2 và 3; được đặt và xóa bằng phần mềm
SCON.2	RB8	9AH	Bit 8 thu, bit thứ 9 thu được
SCON.1	TI	99H	Cờ ngắt phát. Đặt lên 1 khi kết thúc phát ký tự; được xóa phần mềm
SCON.0	RI	98H	Cờ ngắt thu. Đặt lên 1 khi kết thúc thu ký tự; được xóa bằng phần mềm

Bảng tóm tắt thanh ghi chế độ port nối tiếp SCON.

SM0	SM1	Chế độ	Mô tả	Tốc độ baud
0	0	0	Thanh ghi dịch	Cố định (Fosc/12)
0	1	1	UART 8 bit	Thay đổi (đặt bằng timer)
1	0	2	UART 9 bit	Cố định (Fosc/12 hoặc Fosc/64)
1	1	3	UART 9 bit	Thay đổi (đặt bằng timer)

Bảng các chế độ port nối tiếp.

Trước khi sử dụng port nối tiếp, phải khởi động SCON cho đúng chế độ. Ví dụ ,lệnh sau: MOV SCON,#01010010B

Khởi động port nối tiếp cho chế độ 1 (SM0/SM1=0/1), cho phép bộ thu (REN=1) và đặt cờ ngắt phát (TP=1) để chỉ bộ phát sẵn sàng hoạt động.

Khởi động và truy xuất các thanh ghi cổng nối tiếp

Cho phép thu:

Bit cho phép bộ thu (REN = Receiver Enable) trong SCON phải được đặt lên 1 bằng phần mềm để cho phép thu các ký tự. Thông thường thực hiện việc này ở đầu chương trình khi khởi động cổng nối tiếp, timer...Có thể thực hiện việc này theo hai cách. Lệnh :

SETB REN ; Sẽ đặt REN lên 1, hoặc lệnh :

MOV SCON,#xxx1xxxxB ;Sẽ đặt REN 1 và đặc hoặc xóa đi các bit khác trên SCON khi cần (các x phải là 0 hoặc 1 để đặc chế độ làm việc).

Bit dữ liệu thứ 9:

Bit dữ liệu thứ 9 cần phát trong các chế độ 2 và 3, phải được nạp vào trong TB8 bằng phần mềm. Bit dữ liệu thứ 9 thu được đặt ở RBS. Phần mềm có thể cần hoặc không cần bit dữ liệu thứ 9, phụ thuộc vào các đặc tính kỹ thuật của thiết bị nối tiếp sử dụng (bit dữ liệu thứ 9 cũng đóng vai một trò quan trọng trong truyền thông đa xử lý).

Thêm 1 bit parity:

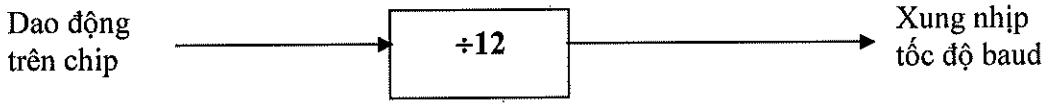
Thường sử dụng bit dữ liệu thứ 9 để thêm parity vào ký tự. Như đã xét ở các chương trước, pit P trong từ trạng thái chương trình (PSW) được đặt lên 1 hoặc bị xóa bởi chu kỳ máy để thiết lập kiểm tra chẵn với 8 bit trong thanh tích lũy.

Các cờ ngắt:

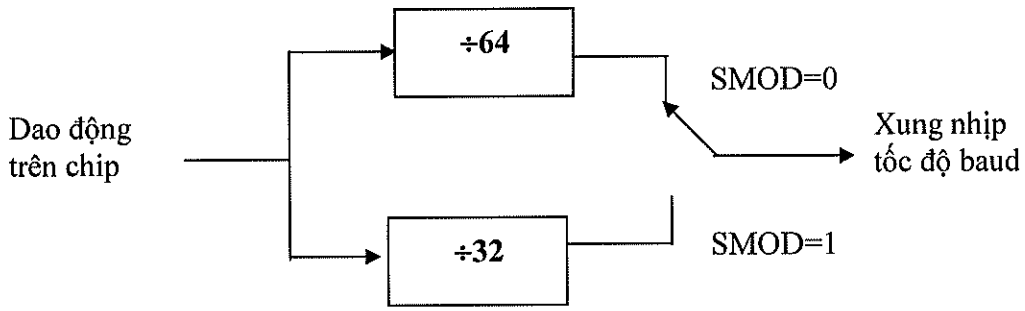
Hai cờ ngắt thu và phát (RI và TI) trong SCON đóng một vai trò quan trọng truyền thông nối tiếp dùng 8051/8031. Cả hai bit được đặt lên 1 bằng phần cứng, nhưng phải được xóa bằng phần mềm.

Tốc độ baud port nối tiếp

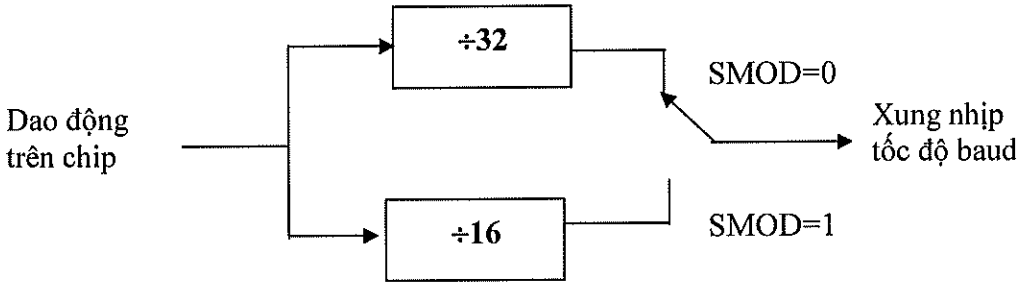
Như đã nói, tốc độ baud cố định ở các chế độ 0 và 2. Trong chế độ 0 nó luôn luôn là tần số dao động trên chip được chia cho 12 . Thông thường thạch anh ấn định tần số dao động trên chip của 8051 nhưng cũng có thể sử dụng nguồn xung nhịp khác. Giả sử với tần số dao động danh định là 12 MHz, tìm tốc độ baud chế độ 0 là 1 MHz.



a. Chế độ 0



b. Chế độ 2



c. Chế độ 1 và 3

Hình 8: Các nguồn tạo xung nhịp cho port nối tiếp

Mặc nhiên, sau khi reset hệ thống, tốc độ baud chế độ là 2 tần số bộ dao động chia cho 64. Tốc độ baud cũng ảnh hưởng bởi 1 bit trong thanh ghi điều khiển nguồn cung cấp (PCON). Bit 7 của PCON là bit SMOD. Đặt bit sMOD lên một làm gấp đôi tốc độ baud trong chế độ 1,2 và 3. Trong chế độ 2, tốc độ baud có thể bị gấp đôi từ giá trị mặc nhiên của 1/64 tần số dao động (SMOD=0) đến 1/32 tần số dao động (SMOD=1)

Vì PCON không được định địa chỉ theo bit, nên để đặt bit SMOD lên 1 cần phải theo các lệnh sau:

MOV A,PCON lấy giá trị hiện thời của PCON

SETB ACC.7 đặt bit 7 (SMOD) lên 1

MOV PCON,A ghi giá trị ngược về PCON

Các tốc độ baud trong các chế độ 1 và 3 được xác định bằng tốc độ tràn của timer 1. Vì timer hoạt động ở tần số tương đối cao, tràn timer được chia thêm cho 32 (hay 16 nếu SMOD=1) trước khi cung cấp xung nhịp tốc độ baud cho port nối tiếp.

CHƯƠNG 2: SƠ LƯỢC VỀ GIAO TIẾP MÁY TÍNH

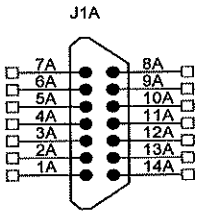
GIAO TIẾP NỐI TIẾP

Vài nét cơ bản về cổng nối tiếp

Cổng nối tiếp RS232 là 1 loại giao diện phổ biến rộng rãi nhất, ta gọi là cổng COM1, COM2 để tự do cho các ứng dụng khác nhau.

Giống như cổng máy in, cổng nối tiếp RS 232 cũng được sử dụng rất thuận tiện trong việc ghép nối máy tính với các thiết bị ngoại vi. Việc truyền dữ liệu qua cổng RS323 được tiến hành theo cách nối tiếp, nghĩa là các bit dữ liệu được gửi đi nối tiếp với nhau trên một đường dẫn.

Trước hết loại truyền này có khả năng dùng cho những khoảng cách lớn hơn, bởi vì khả năng gây nhiễu là nhỏ đáng kể hơn là khi dùng một cổng song song. Việc dùng cổng song song có một nhược điểm đáng kể là cáp truyền nhiều sợi và vì vậy rất đắt tiền, hơn nữa mức tín hiệu nằm trong khoảng 0…5V đã tỏ ra không thích ứng với khoảng cách lớn.



Bảng dưới đây là sự bố trí chân của phích cắm RS232 của máy tính PC.

CHÂN (loại 9 chân)	CHÂN (loại 25 chân)	KÝ HIỆU	Ý NGHĨA
1	8	DCD	Data carrier detect
2	3	RxD	Nhận dữ liệu
3	2	TxR	Phát dữ liệu
4	20	DTR	Dữ liệu đầu cuối sẵn sàng
5	7	GND	Nối đất

6	6	DSR	Thiết bị thông tin sẵn sàng
7	4	RTS	Yêu cầu gửi
8	5	CTS	Thiết bị thông tin sẵn sàng truyền
9	22	RI	Ring in dicator

Việc truyền dữ liệu xảy ra trên 2 đường dẫn qua chân cắm ra TxD, máy tính gửi dữ liệu của nó đến các thiết bị khác. Trong khi đó dữ liệu mà máy tính nhận được dẫn đến chân RsD. Các tín hiệu khác đóng vai trò như tín hiệu hỗ trợ khi trao đổi thông tin và vì thế không phải mọi ứng dụng điều dùng đến.

Mức tín hiệu trên chân ra RxD tuỳ thuộc vào đường dẫn TxD và thông tin thường nằm trong khoảng – 12V..+12V các bit dữ liệu được đảo ngược lại. Mức điện áp ở mức cao nằm trong khoảng – 3V và –12V và mức thấp nằm trong khoảng +3V và +12V. Trạng thái tĩnh trên đường dẫn có mức điện áp –12V.

Bằng tốc độ baud ta thiết lập tốc độ truyền dữ liệu các giá trị thông thường là 300, 1200, 2400, 4800, 9600 và 19200 baud. Ký hiệu baud là số lượng bit truyền trong 1 giây.

Còn một vấn đề nữa là khuôn mẫu (Format) truyền dữ liệu cần phải được thiết lập như nhau cả bên gửi cũng như bên nhận các thông số truyền có thể được thiết lập trên máy tính PC bằng các câu lệnh trên DOS. Ngày nay Window cũng có các chương trình riêng để sử dụng, khi có các thông số truyền dữ liệu như : tốc độ baud, số bít dữ liệu, số bit dừng, số bít chẵn lẻ (parity) có thể được thiết lập một cách đơn giản.

Sự trao đổi của các đường dẫn tín hiệu

Cũng như ở cổng máy in, các đường dẫn tín hiệu riêng biệt cho phép trao đổi qua lại các địa chỉ trong máy tính PC. Trong trường hợp này người ta thường sử dụng những vi mạch có độ tích hợp cao để có thể hợp nhất nhiều chức năng trên 1 chip.

Các máy tính PC thường có độ phát/ nhận không đồng bộ vạn năng gọi tắt là UART : Universal Asynchronous Receiver/Transmister. Để điều khiển sự trao đổi thông tin giữa máy tính và các thiết bị ngoại vi. Phổ biến nhất là vi mạch 8250 của hãng NSC hoặc các thiết bị

tiếp theo, chẳng hạn như 16C550. Bộ AURT này có 10 thanh ghi để điều khiển tất cả chức năng qua việc nhập vào, xuất ra dữ liệu theo cách nối tiếp liên quan đến nội dung của phần này chỉ đề cập đến 2 điều đáng quan tâm đó là : thanh ghi điều khiển modem và thanh ghi trạng thái modem.

Thanh ghi điều khiển modem :

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	LOOP	OUT2	UOT1	RTS	DTR

D0 = 1 đưa /DTR = 0

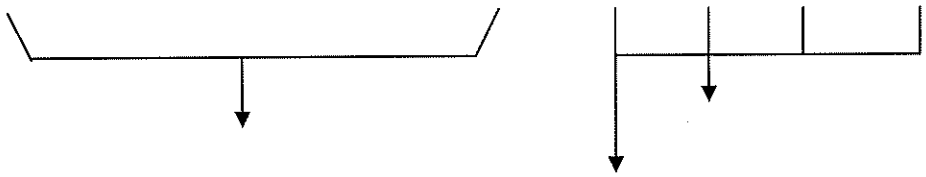
D0 = 0 đưa /DTR = 1

D1 = 1 đưa /RTS = 0

D1 = 0 đưa / RTS = 1

OUT1 và OUT2 điều khiển đầu ra phụ

Thanh ghi trạng thái modem : (địa chỉ cơ bản +6)



Có các giá trị của các bít out2 out1 nếu RI có biến đổi 

DTR, RTS, trong MCR khi bit LOOP = 1 nếu có sự thay đổi các tín hiệu tương ứng

RLSD	RI	DSR	CTS	ΔRLSD	ΔRI	ΔDSR	ΔCTR

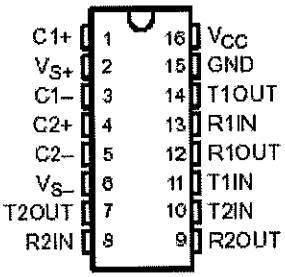
Các thanh ghi được trao đổi qua ô nhớ trong vùng vào/ra (input/output). Địa chỉ đầu tiên có thể tới được của cổng nối tiếp gọi là địa chỉ cơ bản (basic Address) các địa chỉ của các thanh ghi tiếp theo đươc đạt tới bằng việc cộng thêm số thanh ghi đã gặp của bộ UART vào địa chỉ cơ bản.

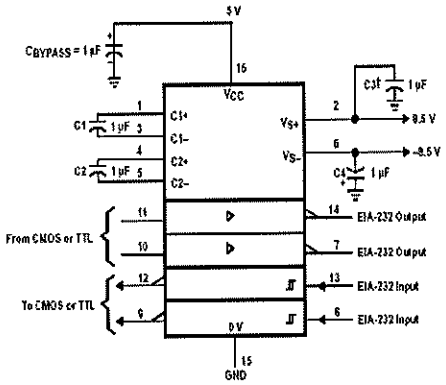
Địa chỉ cơ bản của cổng nối tiếp của máy tính PC được tóm tắt trong bảng sau :

CỔNG	ĐỊA CHỈ CƠ BẢN
COM 1	3F8H
COM 2	2F8H
COM 3	3E8H
COM 3	2E8H

2.1.3. Max 232

Sơ đồ chân



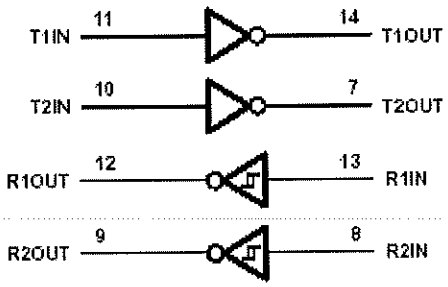


Nguồn 5V

Tốc độ 120bits/s

Hai ngõ thu phát

Sơ đồ luận lý và bảng luận lý



EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

GIAO TIẾP SONG SONG

Vài nét cơ bản về cổng máy in

Việc nối máy in với máy tính được thực hiện qua ổ cắm 25 chân ở phía sau máy tính. Nhưng đây không phải là chổ nối với máy in mà sử dụng máy tính vào việc khác, như truyền dữ liệu từ máy tính với 1 thiết bị khác, hay điều khiển thiết bị bằng máy tính thì việc ghép nối cũng được ghép nối qua cổng máy in.

Qua cổng này dữ liệu được truyền đi song song, nên đôi khi còn được gọi là cổng ghép nối song song và tốc độ truyền cũng đạt đên mức đáng kể. Tất cả các đường dẫn của cổng máy in đều tương thích với TTL. Nghĩa là chúng điều cung cấp mức điện áp nằm giữa 0V đến 5V. Do đó ta cần lưu ý là các đường dẫn vào cổng này không được đặt mức điện áp quá lớn.

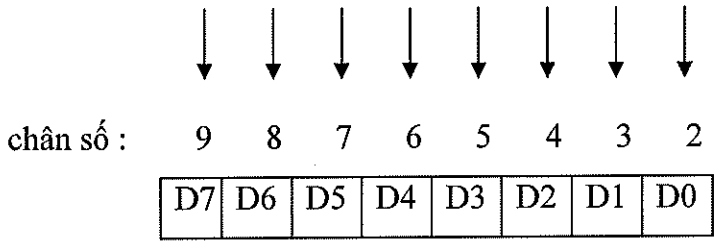
Sự sắp xếp các chân của cổng máy in với tất cả các đường dẫn được mô tả như sau :

Chân	Ký hiệu	Out/in	Chức năng
1	Strobe	Out	Byte được in
2…9	D0 .. D7	Out	Các đường dữ liệu D0 .. D7.
10	ACK	In	Phần thu báo cho phần phát biết đã thu xong một ký tự
11	Busy	In	Phần tín hiệu do phần thu báo cho phần phát biết là phần thu đang bận.
12	PE	In	Báo hết giấy
13	SLCT	In	Báo là chọn máy in
14	AF	Out	Máy tính báo ra máy in tự nạp giấy
15	ERROR	In	Báo các lỗi của máy in
16	INIT	Out	Reset máy in
17	SLCTIN	Out	Chọn máy in
18 .. 25	GND		Nối đất

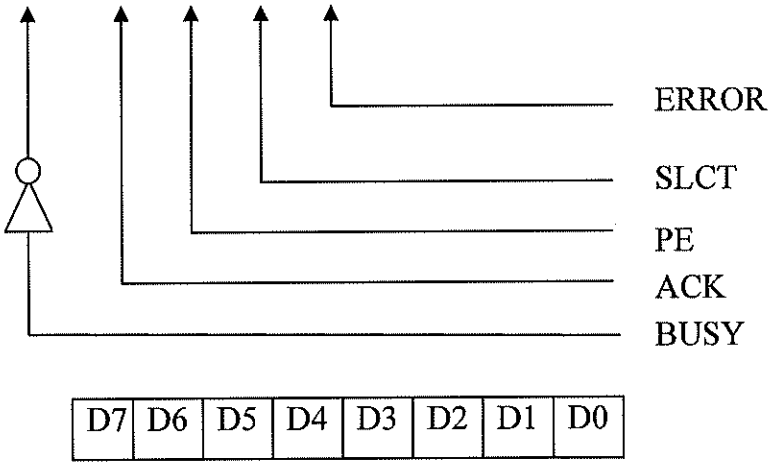
Trao đổi với các đường dẫn tín hiệu

LPT1 : Gồm có 3 thanh ghi : thanh ghi dữ liệu, thanh ghi trạng thái và thanh ghi điều khiển.

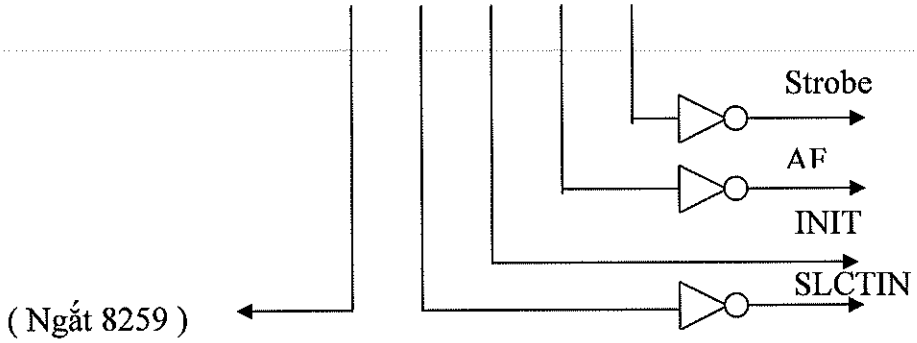
Thanh ghi dữ liệu : gồm có 8 bit dữ liệu, có địa chỉ là 378h.



Thanh ghi trạng thái : có địa chỉ là 379H.



Thanh ghi điều khiển có địa chỉ : 37AH



D7	D6	D5	D4	D3	D2	D1	D0

Hoạt động của việc trao đổi dữ liệu

Hoạt dộng của phần phát

Đọc Busy cho đến khi Busy\=1

Gửi dữ liệu ra bus dữ liệu

Cho Strobe = 0

Chuẩn bị dữ liệu tiếp theo

Quay về bước 1

Hoạt dộng của phần thu

Đọc Strobe cho đến khi Strobe = 1

Busy = 0 , ACK = 0

Đọc dữ liệu vào

Đưa ACK = 1

Xử lý dữ liệu

Cho Busy = 1 để phát ký tự tiếp theo

Quay về bước 1

GIỚI THIỆU SƠ LƯỢC VỀ IC TRUYỀN DỮ LIỆU SN75176B

SN75176B có thể truyền nhận 2 chiều, được thiết kế dùng cho những mạch cần phát cho nhiều điểm trên đường dài trên 1000 mét và trong 1 môi trường ồn ào.

SN75176B có 3 trạng thái ngõ ra 2 chiều, tức nghĩa khi nhận thì A,B là 2 ngõ ra, nhưng khi phát thì ngõ A,B này đóng vai trò là ngõ vào và có 1 ngõ ra ở phía ngược lại đường truyền.

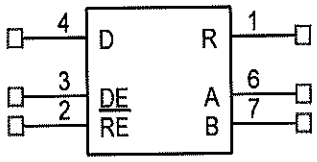
Và chính ưu điểm này, SN75176B được sử dụng trong hệ thống em đang thực hiện có chức năng truyền dữ liệu đi xa trong một môi trường nhiễu rất cao (do có nhiều dây dẫn), và giữ vai trò định hướng đi cho dữ liệu, dữ liệu đi theo chiều gửi hay nhận tuỳ thuộc vào yêu cầu của hệ thống.

SN75176B có 2 chân cho phép, 1 cho ngõ vào và 1 cho ngõ ra.

SN75176B hoạt động trong phạm vi điện áp lớn lẫn 2 chiều và ra, nhưng dòng điện ở ngõ ra theo chiều phát chỉ khoảng 60 mA, và IC này có thể tự tắt khi nhiệt độ quá 70^0C

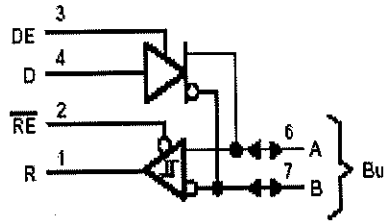
Nguồn của SN75176B là nguồn đơn 5V.

Sơ đồ chân



SN75176

Sơ đồ luận lý và bảng luận lý



DRIVER

INPUT D	ENABLE DE	OUTPUTS A	OUTPUTS B
H	H	H	L
L	H	L	H
X	L	Z	Z

RECEIVER

DIFFERENTIAL INPUTS A–B	ENABLE $\overline{RE}$	OUTPUT R
$V_{ID} \geq 0.2$ V	L	H
-0.2 V < V_{ID} < 0.2 V	L	?
$V_{ID} \leq -0.2$ V	L	L
X	H	Z
Open	L	?

H = high level, L = low level, ? = indeterminate.
X = irrelevant, Z = high impedance (off)

CHƯƠNG 3: GIAO THỨC NỐI MẠNG I^2C

KHÁI QUÁT VỀ GIAO THỨC I^2C

I^2C (Inter-Integrated Circuit) là giao thức gồm 2 đường bus, có tốc độ từ thấp đến trung bình, là bus truyền thông cho tín hiệu điện tử được phát triển bởi công ty Phillips vào đầu thập niên 80. I^2C được đưa ra nhằm giảm chi phí sản xuất các sản phẩm điện tử. I^2C có các ưu điểm như chi phí thấp, nhiều thuận lợi hơn trong việc kết nối các IC với nhau. Các ứng dụng ban đầu dùng giao thức I^2C là phần điều khiển âm lượng và độ tương phản trong máy thu thanh và tivi. Hơn 1 thập kỷ qua, giao thức I^2C đã được sử dụng rộng rãi hơn. Và ngày nay, I^2C lại dễ được tìm thấy trong nhiều ứng dụng điện tử với 1 tiềm năng phát triển không giới hạn.

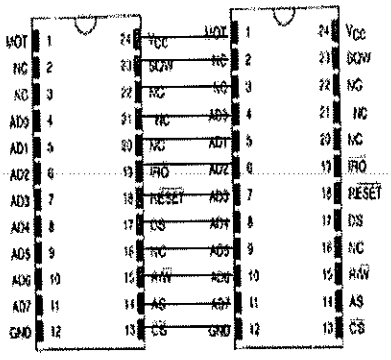
Vài ưu điểm của giao thức I^2C

Đối với tiêu chuẩn trong công nghiệp :

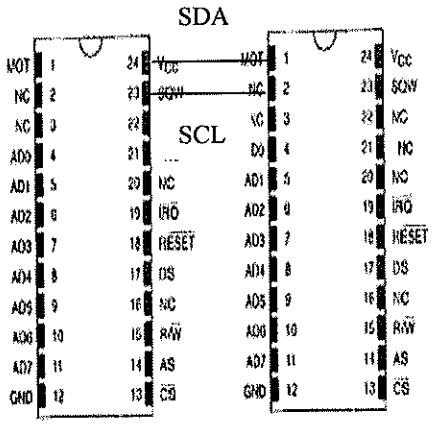
Giao thức I^2C được đưa ra bởi công ty điện tử Phillip của Hà Lan, cung cấp 1 định nghĩa mơi về giao thức truyền thông của tín hiệu điện tử hoạt động trên nền tảng I^2C. Định nghĩa này đã giúp rất nhiều cho các nhà sản xuất cũng như người phát triển các sản phẩm điện tử đi đúng mục đích của công nghệ này. Càng ngày càng nhiều công ty sử dụng giao thức này vào sản phẩm của họ. Vì vậy nó trở thành 1 tiêu chuẩn của ngành công nghiệp điện tử vào thời gian đó.

Chi phí thấp :

Điều trước tiên đối với I^2C, muốn giao tiếp song song nhiều IC lại với nhau thường đòi hỏi các IC phải có từ 24,28, hay nhiều chân hơn. Vì cần nhiều chân cho việc truyền nhận địa chỉ,cho phép truyền nhận, điều khiển và các dữ liệu IC với nhau. Trong giao tiếp song song, 8 bit dữ liệu được truyền nhận IC gửi đến IC nhận



trong 1 thao tác đơn.

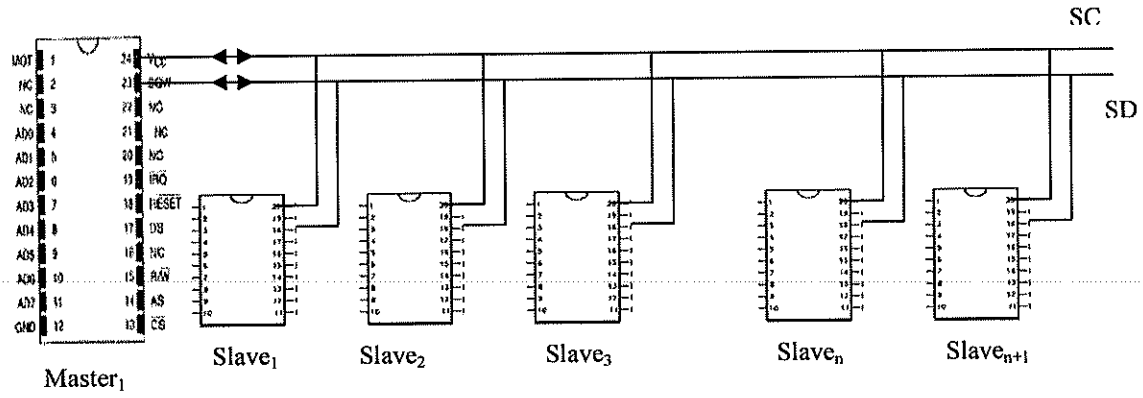


Đối với mạch nối tiếp các IC truyền thông với nhau bằng 2 đường bus, cho phép các IC giao tiếp với nhau bằng vài chân. Hai đường bus trong I^2C được gọi là Clok (SCL) và Data (SDA). Hai đường bus này có thể mang cả địa chỉ, mức cho phép, điều khiển, dữ liệu cùng 1 lúc. Đường bus SDA mang dữ liệu, còn đường bus SCL đồng bộ IC nhận trong lúc kết nối. Giao thức này trong mạch nối tiếp làm cùng 1 chức năng như mạch song song nhưng ít chân kết nối hơn so với mắc song song. Điều này giúp giảm kích cỡ và chi phí của IC trên nền tảng giao thức I^2C .

Cái lợi ích thứ 2 là trong việc thiết kế mạch in, với I^2C chúng ta giảm được nhiều chi phí cũng như kích thước của mạch in vì chúng có ít chân hơn. Cho nên đỡ tốn kém hơn, ít phức tạp hơn và mạch in sẽ nhỏ hơn.

Điểm đặc trưng của I^2C

Với việc giảm chi phí cũng đủ để thấy ưu điểm của I^2C. nhưng người phát minh ra nó không dừng lại ở chỗ đó, họ đã tạo ra nhiều công dụng hơn cho I^2C mà hình dưới đây đã chứng minh cho điều đó.



Hình 9: Một I^2C master và nhiều slave

✓ Master – Slave thứ tự :

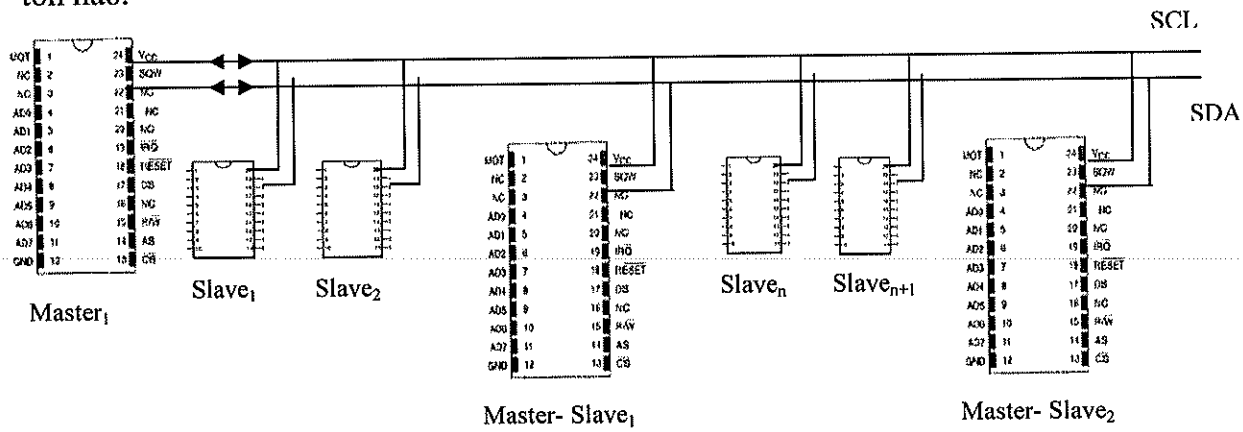
Các thiết bị trong giao thức I^2C bao gồm master và slave. Thiết bị khởi tạo 1 thông điệp đầu tiên gọi là master, còn thiết bị đáp lại thông điệp đó gọi là slave. Một thiết bị có thể gọi là master, slave, hay công tắc giữa master và slave tuỳ thuộc vào ứng dụng yêu cầu.

✓ Nhiều thiết bị :

I^2C có thể liên kết nhiều IC với nhau với chỉ 2 đường bus. Mỗi thiết bị slave có 1 địa chỉ duy nhất. Khi master gửi 1 thông điệp, thông điệp này chứa địa chỉ của thiết bị slave ở đầu thông điệp. Tất cả thiết bị slave đều thấy thông điệp này nhưng chỉ có slave có địa chỉ của nó trên thông điệp mới thực hiện việc trao đổi.

✓ Hỗ trợ nhiều master :

I^2C cũng hổ trợ nhiều master cùng 1 lúc trên đường bus, nhằm tối ưu hoá lưu thông trên đường truyền. Hỗ trợ nhiều master, I^2C phải giải quyết vấn đề xung đột tín hiệu, có nghĩa là 2 hoặc 3 master tranh chấp đướng truyền cùng một thời điểm. Điểm đặc trưng này gọi là bus arbitration loss detection, có nghĩa là cho phép 1 master dò tìm tín hiệu của nó có xung đột với ai hay không. Nếu có nó sẽ chấp dứt việc kết nối của nó và nhường quyền lại cho 1 master khác truyền tiếp thông điệp bằng 1 đường bus chưa tổn hao.



Hình 10: Nhiều master và nhiều slave

Tóm lại, I^2C là một tiêu chuẩn công nghiệp đã được ứng dụng trong một thời gian thách thức đối với ngành công nghiệp điện tử, 1 giao thức truyền thông đã được sử dụng trong nhiều sản phẩm điện tử. I^2C giảm chi phí và có những ưu điểm giáp nó trở thành 1 giao thức trong việc kết nối các IC từ vận tốc thấp đến trung bình.

SƠ LƯỢC VỀ VISUAL BASIC 6.0

Trước khi có Visual, muốn viết một chương trình có giao diện đẹp là rất khó, đơn cử một ví dụ là những chương trình có Menu một vài cấp thì cũng phải viết nhiều câu lệnh rồi, chưa nói tới việc chúng ta phải tạo ra những giao diện vừa đẹp mắt vừa linh hoạt. Trong những năm gần đây, sự ra đời của Windows nhất là từ khi có Window 95, 98, XP... các nhà thiết kế đã hướng con người lập trình theo hướng đối tượng, theo hướng này các bạn được thừa kế những cái gì đã có trước đó, do phần mềm đó cung cấp hoặc do mình tạo ra gọi là các DLL (Dynamic Link Library)

Việc lập trình Visual nói nôm na là dùng những đối tượng đã có sẵn và chúng ta thao tác để lấy những đối tượng cần dùng cho từng chương trình cụ thể theo ý muốn. Sau đó chúng ta viết lệnh cho những đối tượng cần thiết để có thể xử lý và đáp ứng những yêu cầu đòi hỏi phải thực hiện. Trong thực tế hiện nay trên thế giới và ngay cả nước ta đã thay đổi quan điểm lập trình cũ để dần dần lập trình theo hướng mới này. Hiện nay chúng ta thấy có Visual FoxPro, Visual C++, Visual Java, Visual Basic ..... là những công cụ khá mạnh để giúp con người lập trình trong nhiều lĩnh vực.

Ngày nay kỹ thuật lập trình Visual Basic được ứng dụng rộng rãi trong kỹ thuật và nhiều lĩnh vực khác. Visual Basic được xem như một công cụ phát triển phần mềm như các trình biên dịch C/C++ hay SDK. Nhưng lợi điểm khi dùng Visual Basic chính ở chỗ tiết kiệm thời gian và công sức hơn so với các ngôn ngữ lập trình khác khi xây dựng cùng một ứng dụng.

Visual Basic gắn liền với khái niệm lập trình trực quan (visual), nghĩa là khi thiết kế chương trình, bạn được nhìn thấy ngay kết quả qua từng thao tác và giao diện

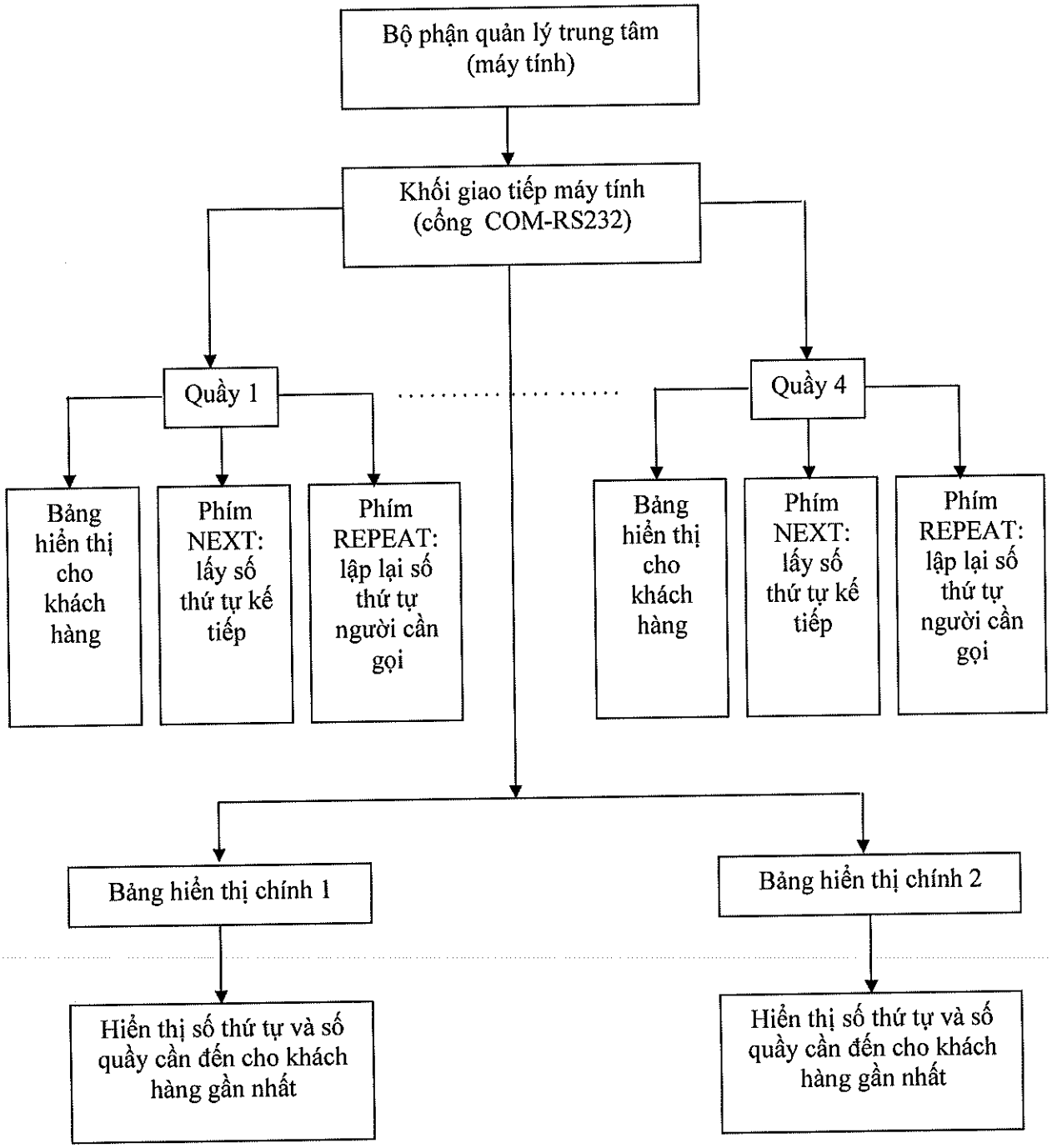
khi chương trình thực hiện. Đây là thuận lợi lớn so với các ngôn ngữ lập trình khác, Visual Basic cho phép chỉnh sửa đơn giản, nhanh chóng màu sắc, kích thước, hình dáng của các đối tượng có mặt trong ứng dụng.

Visual Basic là sản phẩm của hãng Microsoft , nó đã trải qua nhiều phiên bản và hiện nay là Visual Basic 6.0. Trong phiên bản này hiện có ba ấn bản đó là Learning, Pro và Interprise. Và em đã dùng ấn bản Pro để thực hiện thiết kế giao diện cho chương trình của hệ thống QMS.

PHẦN HAI

TÍNH TOÁN VÀ THIẾT KẾ

CHƯƠNG 1: SƠ ĐỒ KHỐI HỆ THỐNG



Bộ phận quản lý trung tâm (máy tính):

Chức năng: dùng để quản lý, cung cấp số thứ tự cho các quầy và hai bảng hiển thị chính .

Khối quầy :(có 4 quầy)

Chức năng: nhận dữ liệu khách hàng từ bộ phận quản lý trung tâm (máy tính) thông qua cổng COM và IC MAX232, việc giao dịch được điều khiển thông qua các khối:

Hiển thị: hiển thị số thứ tự của khách hàng đang giao dịch .

NEXT: Khi cần giao dịch với khách hàng kế tiếp thì nhấn phím NEXT.

REPEAT: Khi cần lặp lại số thứ tự của khách hàng cần giao dịch thì nhấn phím REPEAT.

Khối hiển thị chính: (có 2 bảng hiển thị chính)

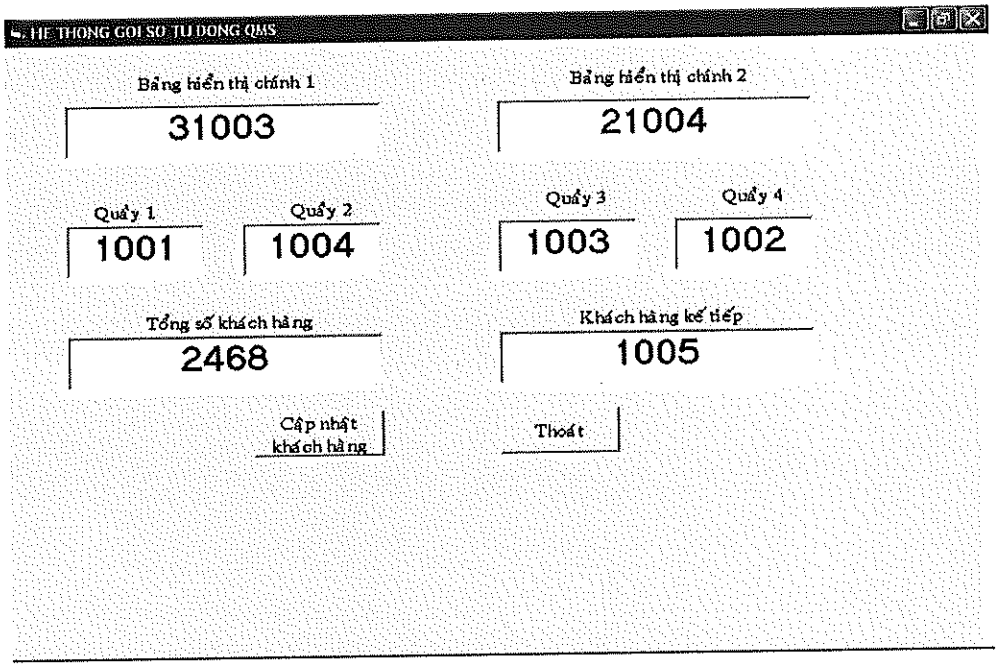
Chức năng: nhận và hiển thị số thứ tự và số quầy giao dịch của khách hàng gần nhất từ bộ phận quản lý trung tâm (máy tính) thông qua cổng COM và IC MAX232.

CHƯƠNG 2:TÍNH TOÁN VÀ THIẾT KẾ

BỘ PHẬN QUẢN LÝ TRUNG TÂM (MÁY TÍNH)

Chức năng

Dùng để quản lý, cung cấp số thứ tự cho các quầy và hai bảng thông tin chính. Dưới đây là màn hình giao diện trên máy tính trung tâm .



Hình 11: Màn hình giao diện khối xử lý trung tâm

Thành phần của khối xử lý trung tâm

Hai bảng hiển thị chính: hiển thị số thứ tự gần nhất của khách hàng được gọi và số quầy đang làm việc với số thứ tự đó. Hai bảng này thông báo cho khách hàng bằng cách phát giọng nói ra loa bằng trung tâm xử lý (máy tính) và số hiển thị được nhấp nháy.

Ô bốn quầy con: hiển thị số thứ tự của khách hàng được gọi.

Ô khách hàng kế tiếp: cho người quản lý biết số thứ tự khách hàng sẽ được gọi

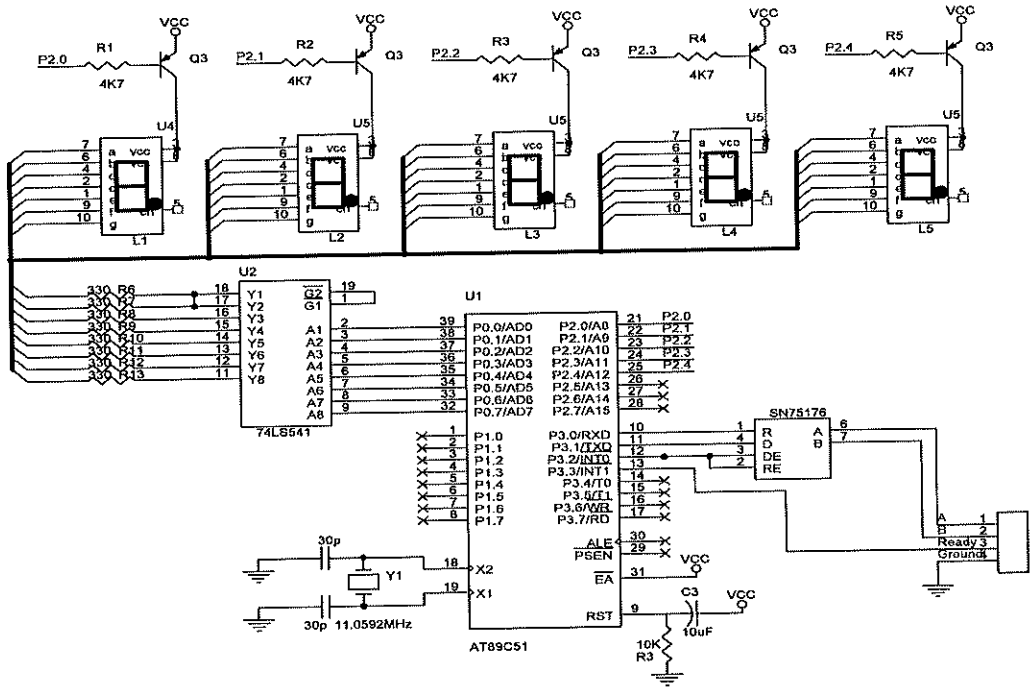
tiếp theo nếu một trong bốn quầy con yêu cầu.

Ô tổng số khách hàng : cho người quản lý biết số lượng khách hàng đã và đang chờ đợi .

Nút thêm khách hàng : cho người quản lý cập nhật số lượng khách hàng. Thực tế, khi có khách hàng vào sẽ có một người cấp cho khách hàng này một số thứ tự và số thứ tự này được cập nhật vào dữ liệu tổng số khách hàng .

Nút thoát chương trình: để thoát chương trình gọi số tự động .

KHỐI HIỂN THỊ CHÍNH



Hình 12 : Sơ đồ nguyên lý khối hiển thị chính

Các thành phần của khối hiển thị chính

Gồm một IC vi xử lý AT89C51 có nhiệm vụ gửi và nhận dữ liệu từ bộ phận quản lý trung tâm (máy tính). Bộ vi xử lý này còn có giao tiếp với các mạch sau:

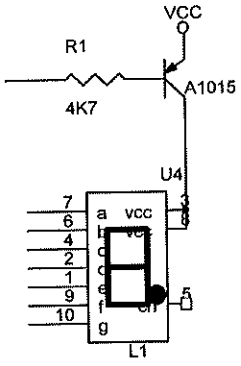
Mạch hiển thị: gồm 5 LED 7 đoạn lớn, 1 LED hiển thị số quầy và 4 LED hiển thị số thứ tự của khách hàng gần nhất.

IC SN75176B: dùng để truyền và nhận dữ liệu giữa khối hiển thị chính và bộ phận quản lý trung tâm.

IC 74LS541: dùng để nâng dòng cho LED 7 đoạn.

Tính toán

Mạch hiển thị



Xác định BJT: chọn transitor A1015 có các thông số sau : P_{max}=1W, I_{Cmax}=1mA, β=100 .

Để LED 7 đoạn được sáng rõ khi dùng phương pháp quét LED ta chọn dòng qua mỗi đoạn LED là 10mA và điện áp là 2V. Như vậy dòng I_C qua transitor khi cả LED 7 đoạn sáng là 70mA .

Xác định R1→R5:

Điều kiện để transitor bảo hoà sâu là : $Ic.K < \beta I_B$, với $K=1 \rightarrow 5$

Chọn K=2, suy ra : $I_B = \frac{I_C * K}{\beta} = \frac{70*2}{100} = 1,4mA$

Ta có: $V_{CC} = V_{BE} + R1*I_B$

Suy ra: $R1 = \frac{V_{CC} - V_{BE}}{I_{BQ}} = \frac{5-0.6}{1,4} = 3,14K\Omega$

Vậy chọn $R1 \rightarrow R5 = 4,7\ K\Omega$

Xác định R6→R13 hạn dòng:

$$R_{hd} = R6 = \frac{V_{CC} - V_{led}}{I_{led}} = \frac{5-2}{10mA} = 300\Omega$$

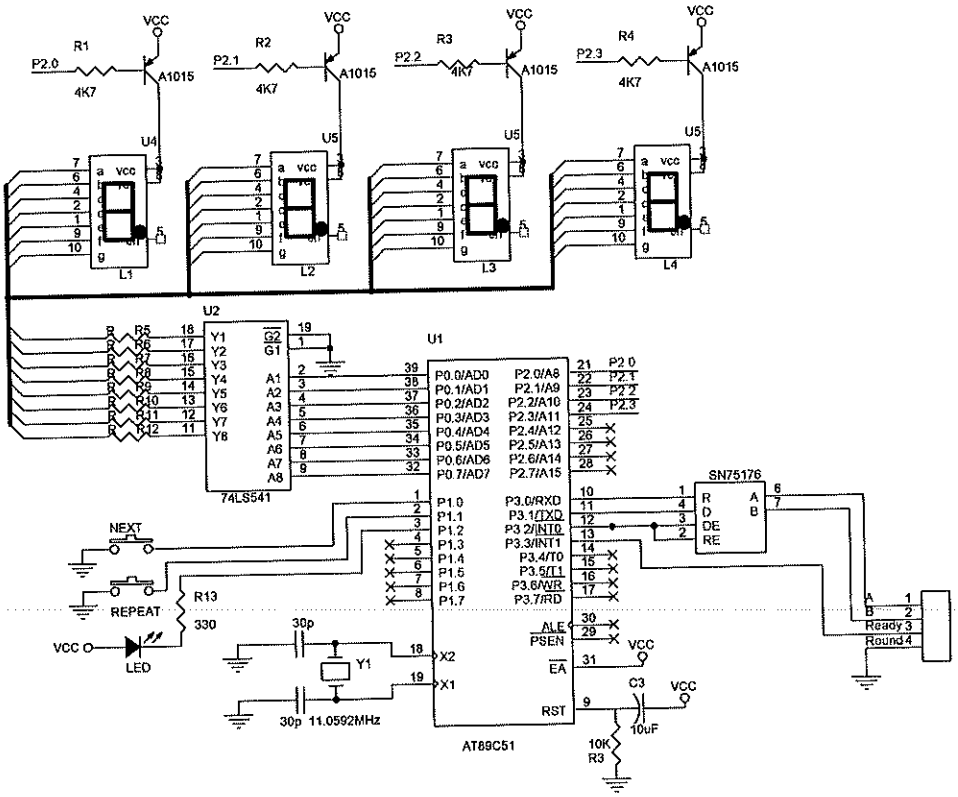
Chọn R6→R13 = 330Ω

Nguyên lý hoạt động

Khi tất cả các quầy gửi yêu cầu lên máy tính trung tâm thì máy tính sẽ gửi lại số thứ tự của khách hàng cần giao dịch theo thứ tự ưu tiên (quầy nào gửi yêu cầu trước thì sẽ nhận dữ liệu trước).

Đồng thời máy tính cũng gửi số thứ tự và số quầy của khách hàng gần nhất để hiển thị trên bảng hiển thị chính.

KHỐI QUẦY



Hình 13 : Sơ đồ nguyên lý khối quầy

Các thành phần của quầy

Gồm một IC vi xử lý AT89C51 có nhiệm vụ gửi và nhận dữ liệu từ bộ phận quản lý trung tâm (máy tính). Bộ vi xử lý này còn có giao tiếp với các mạch sau:

Mạch hiển thị: gồm 4 LED 7 đoạn để hiển thị số thứ tự của khách hàng.

Các phím nhấn:

Phím nhấn NEXT: dùng để yêu cầu khối xử lý trung tâm gửi về số thứ tự của khách hàng kế tiếp.

2.3. *Phím REPEAT:* khi được nhấn thì máy tính sẽ lặp lại số thứ tự của khách hàng cần gọi (ví dụ khi gọi số thứ tự một khách hàng lần 1 mà không thấy khách hàng tới giao dịch thì nhấn REPEAT để lập lại khách hàng đó) .

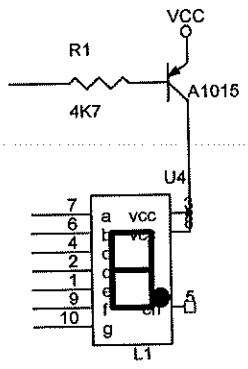
Mạch báo hiệu: gồm một LED có chức năng báo hiệu cho nhân viên biết còn khách hàng chờ đợi hay không (giả sử LED sáng là còn khách hàng, LED tắt là hết khách hàng)

IC SN75176B: dùng để truyền và nhận dữ liệu giữa quầy và bộ phận quản lý trung tâm.

IC 74LS541: dùng để nâng dòng cho LED 7 đoạn.

Tính toán

Mạch hiển thị



Tính toán tương tự như trên khối hiển thị chính, ta được:

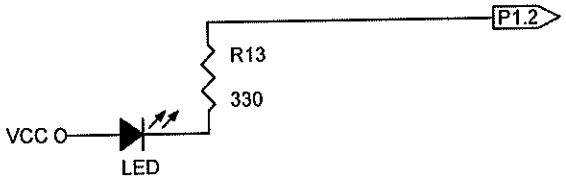
$$R1 = \frac{V_{CC} - V_{BE}}{I_{BQ}} = \frac{5-0.6}{1,4} = 3,14K\Omega$$

Chọn R1 → R4 = 4,7 KΩ

$$R_{hd} = R5 = \frac{V_{CC} - V_{led}}{I_{led}} = \frac{5-2}{10mA} = 300\Omega$$

Chọn R5 → R12 = 330Ω

Mạch báo hiệu



Xác định R13

Để LED đơn sáng thì dòng I_{led} = 10mA, với V_{led} = 2V

Suy ra : $R13 = \frac{V_{CC} - V_{led}}{I_{led}} \frac{5-2}{10mA} = 0.3K\Omega = 300\Omega$

Vậy chọn R13 = 330 Ω

Nguyên lý hoạt động

Mạch hiển thị: gồm 4 LED 7 đoạn lớn để hiển thị số thứ tự của khách hàng. Ngoài ra để nhân viên dễ dàng nhận biết được giá trị này thì ở mỗi quầy cũng có 4 LED 7 đoạn nhỏ để hiển thị .

Khối xử lý trung tâm của quầy:

Đầu tiên khối quầy gửi dữ liệu đến bộ phận quản lý trung tâm bằng 1 byte địa chỉ (Ví dụ: 4 quầy thì mỗi quầy có địa chỉ lần lượt là: 01, 02, 03, 04 để nhận dạng) .

Khi nhận được dữ liệu này thì trung tâm sẽ lần lượt cấp cho mỗi quầy một số thứ tự khách hàng để giao dịch (việc thực hiện này sẽ thực hiện theo thứ tự ưu tiên của

dữ liệu gửi đến trước hay sau). Đồng thời trung tâm hiển thị số thứ tự này trên bảng hiển thị chính và sẽ đọc trên loa cho khách hàng biết số quầy sẽ thực hiện giao dịch và sẽ cho hiển thị trên LED của quầy đó số thứ tự của khách hàng (Ví dụ: mời số 1234 đến quầy 1).

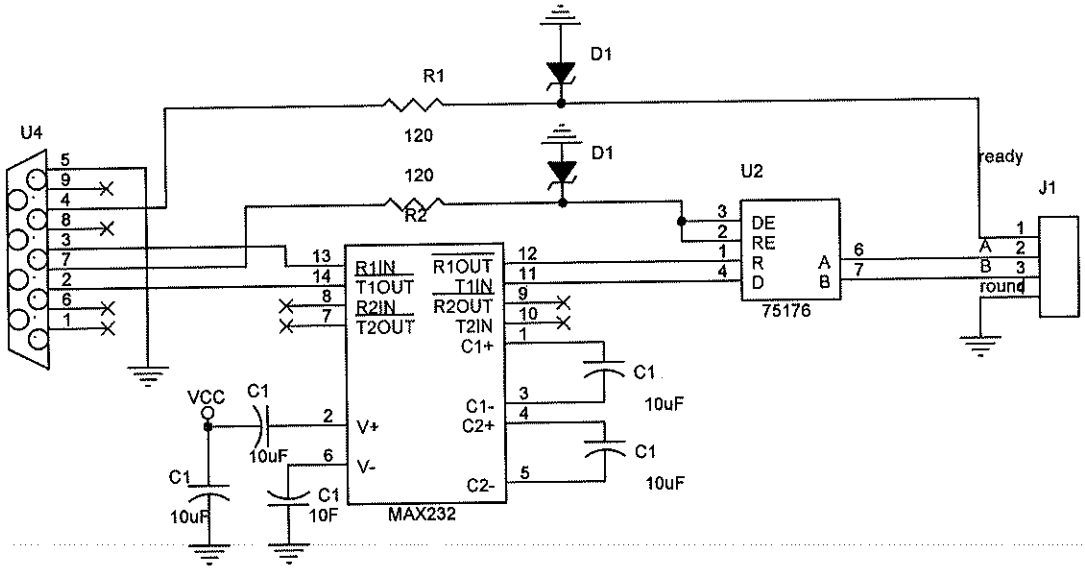
Lúc này khách hàng biết số thứ tự của mình sẽ đến quầy đang hiển thị số thứ tự của mình để thực hiện giao dịch. Tuy nhiên, để tránh trường hợp khách hàng không nghe thấy thì nhân viên tại quầy có thể nhấn REPEAT để đọc lại số thứ tự đó lần nữa.

Các phím nhấn:

Phím nhấn NEXT: dùng để yêu cầu khối xử lý trung tâm gửi về số thứ tự của khách hàng kế tiếp.

Phím REPEAT: dùng để yêu cầu khối xử lý trung tâm lập lại số thứ tự mà khi quầy con gọi lần thứ nhất không có khách hàng .

KHỐI GIAO TIẾP MÁY TÍNH



Hình 14:Sơ đồ nguyên lý khối giao tiếp máy tính

Chức năng

Thực hiện giao tiếp giữa máy tính và các quầy thông qua cổng COM.

Tính toán R1, R2

Chọn diode Zener có điện thế phân cực nghịch là $V_D = 5V$, $I_D = 5mA$

Nguồn ra từ máy tính là $V_{MT} = 12V$

Chọn $I = I_D + I_{MAX\ 75176} = 5 + 60 = 65mA$

Suy ra : $R1 = \frac{V_{MT} - V_D}{I} = \frac{12V - 5V}{65mA} = 107,7\Omega$

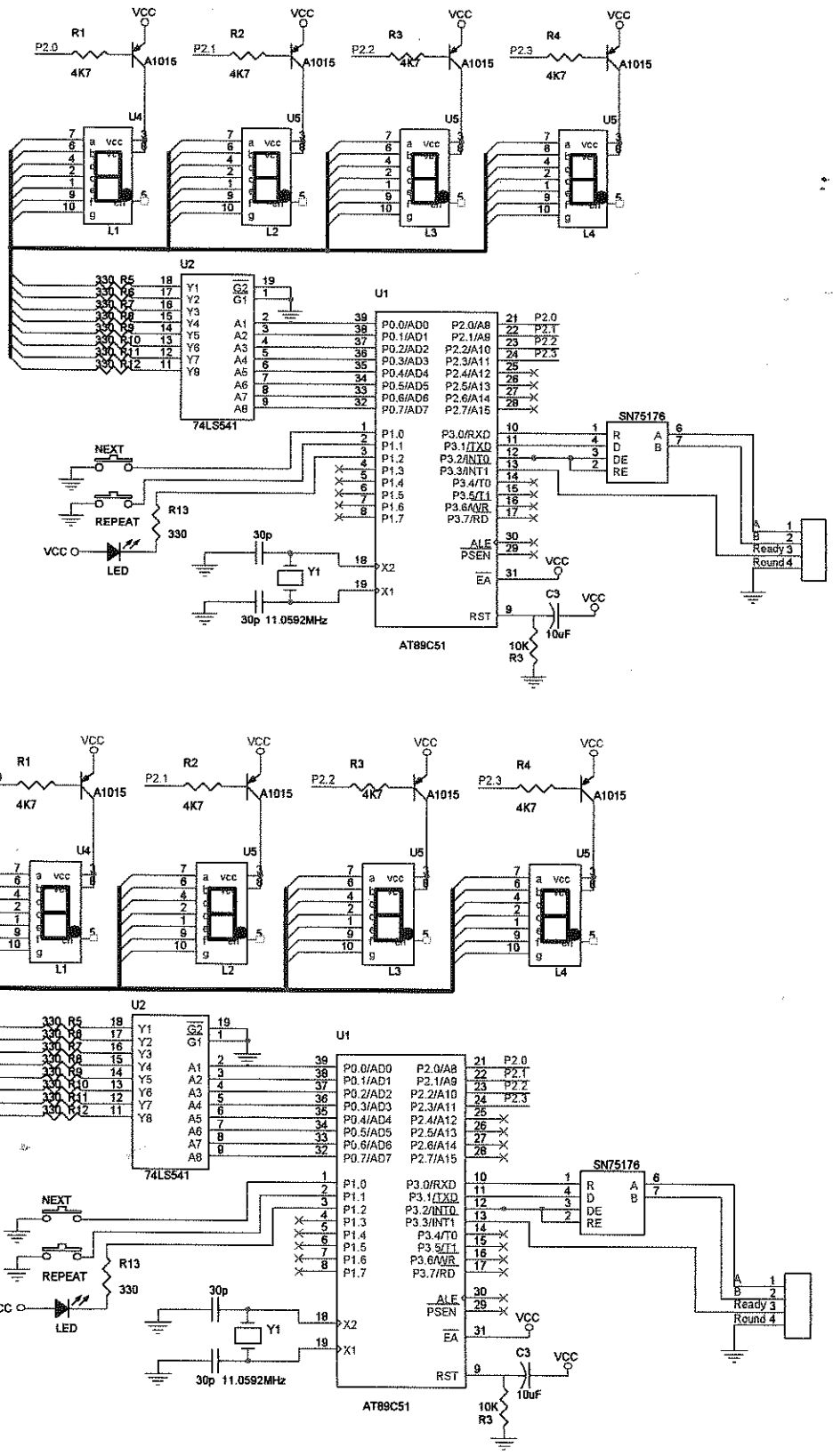
Vậy chọn R1= R2 = 120Ω

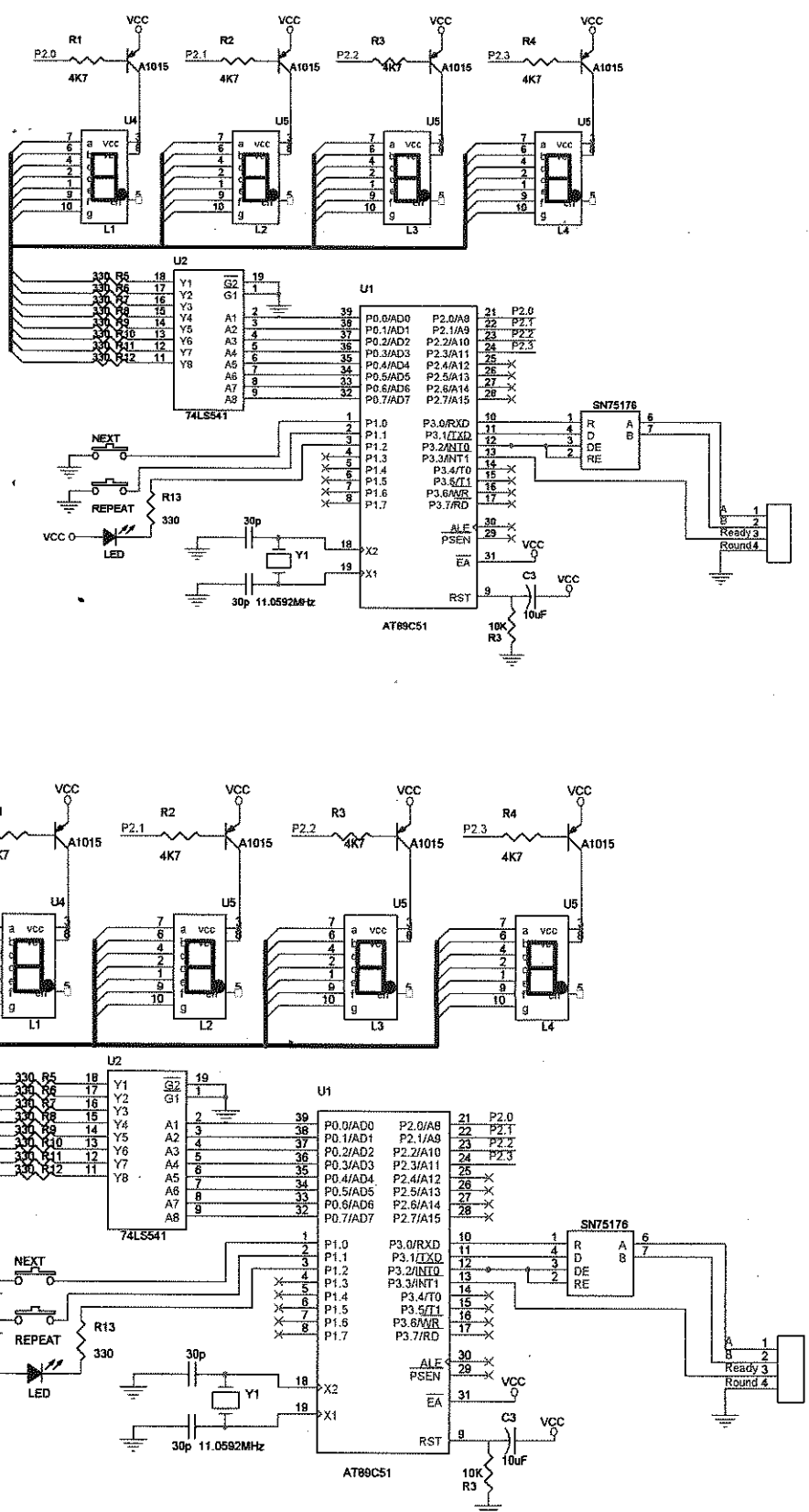
Nguyên lý hoạt động

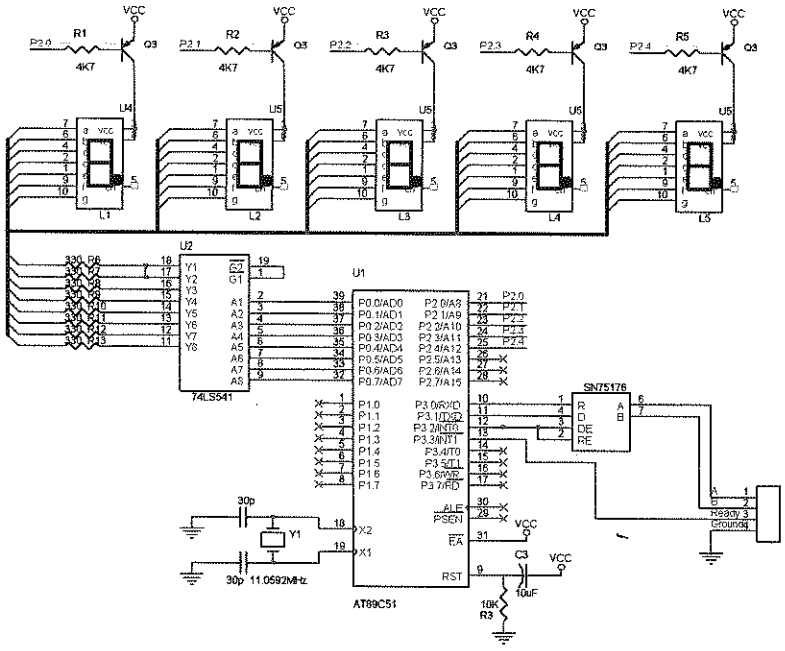
Máy tính gửi dữ liệu ra chân số 3 của cổng COM và được MAX232 tiếp nhận, IC này sẽ biến đổi mức logic của máy tính sang mức logic của vi điều khiển và truyền tới cho vi điều khiển AT89C51, vi điều khiển sẽ nhận dữ liệu và xử lý. Ngược lại khi vi điều khiển gửi dữ liệu lên máy tính sẽ thông qua chân số 2 của cổng COM để chuyển đổi mức logic của vi điều khiển sang mức logic của máy tính .

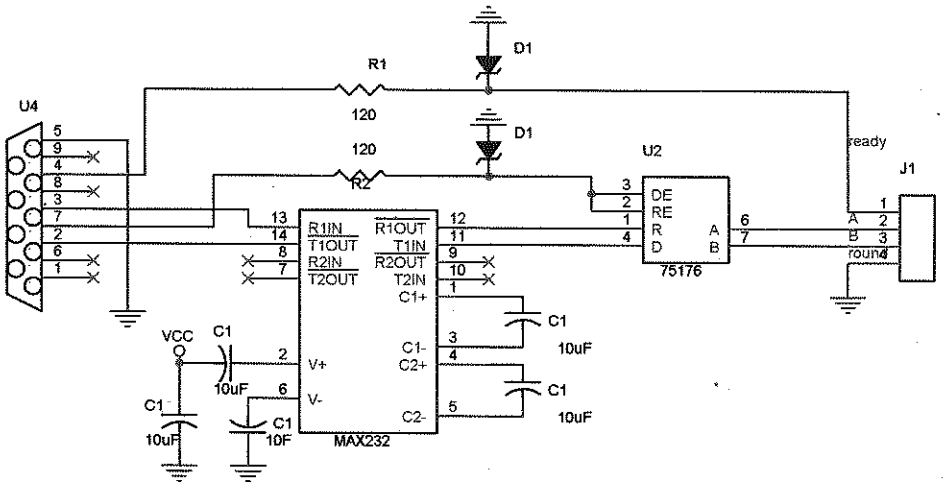
CHƯƠNG 3: SƠ ĐỒ NGUYÊN LÝ TOÀN HỆ THỐNG QMS

3.1. SƠ ĐỒ NGUYÊN LÝ









NGUYÊN LÝ HOẠT ĐỘNG CỦA TOÀN MẠCH

Khi khách hàng đến, người quản lý sẽ cấp cho khách hàng một mảnh giấy có số thứ tự trên đó và người quản lý tăng số lượng khách hàng trên máy tính lên 1 và số liệu này được đưa vào một file txt. Mặc định ban đầu của số này là 0 và ô khách hàng kế tiếp là . Nếu ô số lượng khách hàng lớn hơn 0 thì ô khách hàng kế tiếp sẽ tăng lên 1.

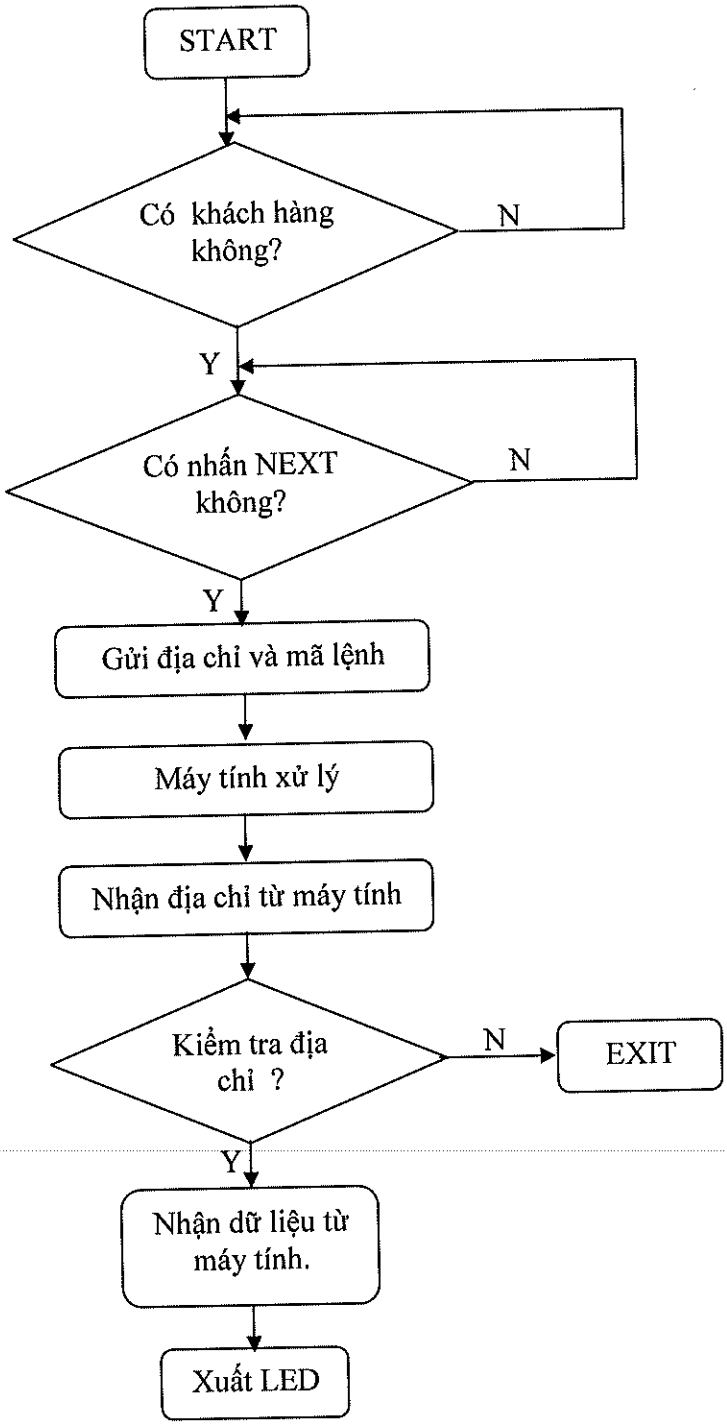
Khi có khách hàng chờ đợi thì máy tính sẽ báo cho các quầy biết bằng môt LED đơn sáng và nhân viên ở các quầy chuẩn bị làm việc. Khi một quầy con ấn NEXT gọi khách hàng thì máy tính nhận được lệnh và trả về cho quầy con số thứ tự của ô khách hàng kế tiếp trên giao diện máy tính và số thứ tự của ô này sẽ tăng lên 1. Số thứ tự này được phát trên loa, hiển thị nhấp nháy trên bảng chính và cả quầy con để thông báo cho khách hàng biết .

Tiếp theo khách hàng sẽ giao dịch với nhân viên tại quầy và ra về. Và cứ như thế cho đến khi không còn khách hàng nào chờ đợi, tất nhiên LED đơn ở quầy con sẽ tắt báo cho nhân viên biết đã hết khách hàng. Và khi có khách hàng đến thì chu trình được lặp lại.

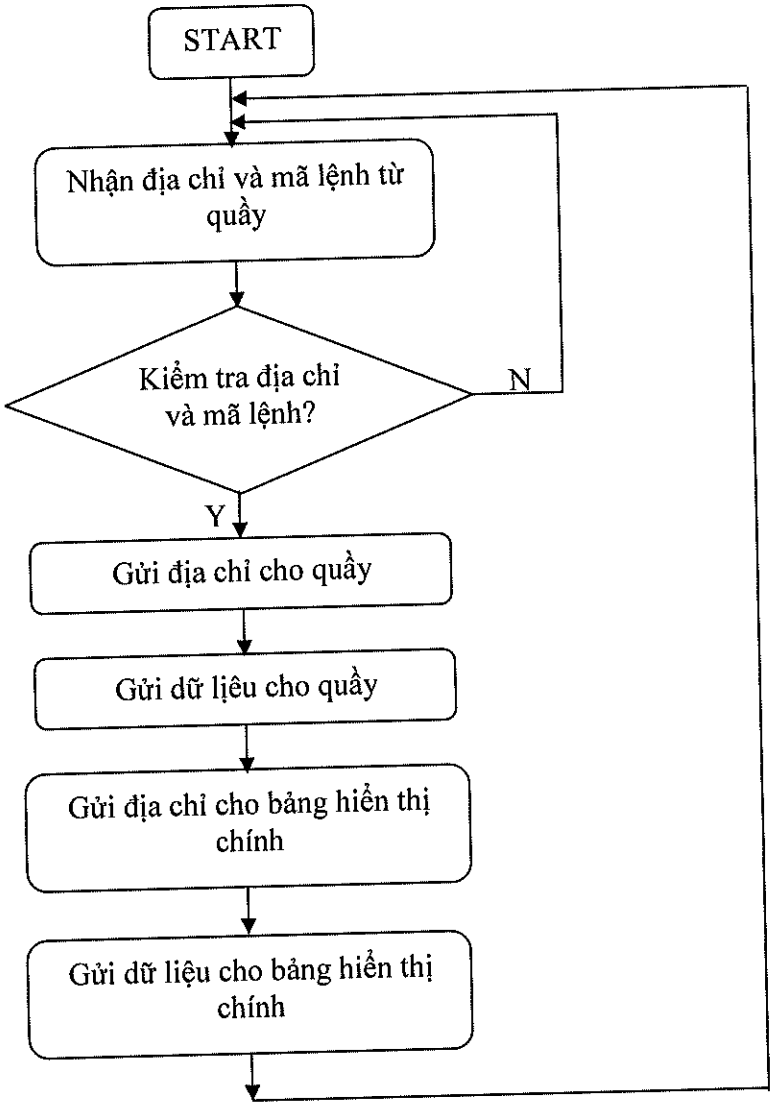
Trong quá trình quầy con gọi khách hàng, nếu nhân viên không thấy khách hàng đến giao dịch thì nhân viên sẽ ấn phím REPEAT để lặp lại cho khách hàng biết.

CHƯƠNG 4: LƯU ĐỒ THUẬT TOÁN

KHỐI QUẦY



BỘ PHẬN XỬ LÝ TRUNG TÂM



CHƯƠNG 5: KẾT LUẬN

ƯU KHUYẾT ĐIỂM CỦA ĐỀ TÀI

Hệ thống gọi số tự động QMS được làm dưới dạng module nên có khả năng mở rộng ra nhiều quầy (slave) khi cần. Giao diện thân thiện, đơn giản nên dễ sử dụng, quản lý. Giúp giảm thiểu thời gian giao dịch của khách hàng tại bệnh viện, bưu điện, điện lực . . . . Hệ thống QMS của em hoạt động ổn định, khả năng xảy ra lỗi tranh chấp đường truyền thấp. Tiết kiệm chi phí do dùng phương pháp quét LED và các linh kiện rẻ, dễ tìm.

Ngoài ra, với việc dùng máy tính quản lý cơ sở dữ liệu nên khi xảy ra mất điện thì dữ liệu vẫn không mất đi và hệ thống vẫn hoạt động bình thường khi có điện trở lại.

Bên cạnh những ưu điểm trên, đề tài của em còn một số hạn chế sau: chưa kết nối hệ thống với máy in để tự động hóa hoàn toàn; chưa cập nhật được chi tiết khách hàng; chưa tổng kết được khách hàng vào từng quý, năm; chưa có một hệ thống dự trữ trong trường hợp máy tính bị hỏng. Và nếu có đủ thời gian thì em sẽ hoàn thiện tốt luận văn này.

HƯỚNG PHÁT TRIỂN ĐỀ TÀI

Kết nối hệ thống với máy in để khi có khách hàng đến hệ thống sẽ tự động in cho khách hàng một số thứ tự để giao dịch. Khi thực hiện giao dịch xong hệ thống sẽ in cho khách hàng một biên trên đó có in tất cả thông tin cần thiết như họ tên, địa chỉ của khách hàng;ngày, giờ giao dịch …

Xây dựng một hệ thống dự trữ để đảm bảo cho hệ thống hoạt động hiệu quả khi máy tính trung tâm bị hỏng.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Lê Quang Đức & Hà Đắc Bình (2004) ,Kỹ thuật vi xử lý .

[2] Tống Văn On & Hoàng Đức Hải (2004), Hợp ngữ và lập trình ứng dụng, NXB Lao động - Xã hội .

[3] Ngô Diên Tập (2000), Lập trình ghép nối trong Windows, NXB Khoa học và Kỹ thuật, Hà Nội .

[4] Đậu Quang Tuấn (2000), Tự học lập trình Visual Basic 6.0, NXB trẻ .

[5] Dương Minh Trí (2002), Sơ đồ chân linh kiện bán dẫn, NXB Khoa học và Kỹ thuật.

[6] Datasheet c ủa 89C51, SN75176B, RS232, 74LS541 .

Các website

www.atmel.com

www.digichip.com

www.softcircuit.com

Và một số tài liệu , luận văn khác .

PHẦN BA

PHỤ LỤC

CHƯƠNG TRÌNH

KHỐI HIỂN THỊ CHÍNH

```
LED1        EQU  70H
LED2        EQU  71H
LED3        EQU  72H
LED4        EQU  73H
LED5        EQU  74H
LED6        EQU  75H
LED7        EQU  76H
LED8        EQU  77H
LED9        EQU  78H
LED10       EQU  79H
LEDCHOP     EQU  7AH
TT_LED      EQU  7BH
ID          EQU  7CH

KET_NOI     BIT  7EH
THU_DATA    BIT  7DH

    ORG  0000H
    LJMP BEGIN
    ORG  000BH
    LJMP QUET_LED
```

```
ORG  0023H
PUSH ACC
CLR  RI
MOV  A,SBUF
CLR  C
SUBB A,#30H

 JB     THU_DATA,LUU_DATA        ;DUNG CHUOI PC TRA LOI
THU LAI DATA DE HIEN THI

  CJNE A,#11H,KTRA_12H           ;QUAY 1 TRA LOI
PHIM NEXT
  SJMP PC_GOI_DATA
KTRA_12H:
  CJNE A,#12H,KTRA_21H           ;QUAY 1 TRA LOI
PHIM RE
  SJMP PC_GOI_DATA
KTRA_21H:
  CJNE A,#21H,KTRA_22H           ;QUAY 2 TRA LOI
PHIM NEXT
  SJMP PC_GOI_DATA
KTRA_22H:
  CJNE A,#22H,KTRA_31H           ;QUAY 2 TRA LOI
PHIM RE
  SJMP PC_GOI_DATA
KTRA_31H:
  CJNE A,#31H,KTRA_32H           ;QUAY 3 TRA LOI
```

```
PHIM NEXT
    SJMP PC_GOI_DATA
KTRA_32H:
    CJNE A,#32H,KTRA_41H          ;QUAY 3 TRA LOI
PHIM RE
    SJMP PC_GOI_DATA
KTRA_41H:
    CJNE A,#41H,KTRA_42H          ;QUAY 4 TRA LOI
PHIM NEXT
    SJMP PC_GOI_DATA
KTRA_42H:
    CJNE A,#42H,KTRA_THU_DATA     ;QUAY 4 TRA LOI
PHIM RE
    SJMP PC_GOI_DATA


KTRA_THU_DATA:
        CJNE A,#28H,EXIT
        SETB THU_DATA
        SJMP EXIT

LUU_DATA:
        MOV @R1,A
        INC  R1
        CJNE R1,#65H,EXIT
        MOV R1,#60H
```

```
        MOV  LEDCHOP,#10D
        CLR  7FH
        MOV  LED6,LED1
        MOV  LED7,LED2
        MOV  LED8,LED3
        MOV  LED9,LED4
        MOV  LED10,LED5
        MOV  70h,60H
        MOV  71h,61H
        MOV  72h,62H
        MOV  73h,63H
        MOV  74h,64H
        CLR  P1.0          ; CHUYEN PC VE NHAN DATA
        CLR  THU_DATA
        LJMP EXIT

PC_GOI_DATA:  SETB P1.0

EXIT: POP  ACC
           RETI

BEGIN:     MOV  TMOD,#21H
           MOV  TH1,#-26
           SETB TR1
           MOV  SCON,#50H
           clr  RI
           clr  TI
```

```
SETB EA
SETB ES
SETB ET0
MOV TL0,#18H          ;DELAY 1ms
MOV TH0,#0FCH
SETB TR0
CLR  TF0
MOV LED1,#10
MOV LED2,#10
MOV LED3,#10
MOV LED4,#10
MOV LED5,#10
MOV LED6,#10
MOV LED7,#10
MOV LED8,#10
MOV LED9,#10
MOV LED10,#10

MOV R2,#0D
MOV R1,#60H
MOV LEDCHOP,#0D
MOV TT_LED,#0
MOV ID,#10H           ;TUONG UNG QUAY 1
CLR  P3.2
CLR  P1.0
CLR  KET_NOI
CLR  THU_DATA
```

```
MAIN:
        JNB   KET_NOI,MAIN
        CLR   ES                    ;cam ngat noi tiep
        SETB P3.2
        CLR   P1.0
        LCALL      DELAY
        MOV  A,#30H
        ADD  A,ID
        MOV  SBUF,A
        JNB   TI,$
        CLR   TI
        MOV  A,#10H
        ADD  A,ID
        MOV  ID,A
        CJNE A,#50H,XOA_P32
        MOV  ID,#10H
XOA_P32:
        CLR   P3.2
        CLR   KET_NOI
        SETB ES
        LJMP MAIN


QUET_LED:
        PUSH ACC
```

```
        MOV  TL0,#18H
        MOV  TH0,#0FCH

        INC   R2
        MOV  A,R2
        CJNE A,#200D,QUET
        MOV  R2,#0D

        JB    P1.0,DAO_7FH          ;DANG CHO DATA TU PC =      KG
                                    QUET CAC QUAY KHAC
        SETB KET_NOI

DAO_7FH: CPL   7FH
        MOV  A,LEDCHOP
        CJNE A,#00H,GIAM
        SETB 7FH
        SJMP QUET
GIAM:   DEC   LEDCHOP
QUET:   JB     7FH,MO_LED
        MOV  P0,#0FFH
        SJMP EXIT_Q
MO_LED: MOV  P2,#0FFH
        SETB P1.6
        SETB P1.7
        MOV  A,TT_LED
        ADD   A,#70H
        MOV  R0,A
```

```
    MOV A,@R0
    LCALL     MA7D
    MOV B,A
    MOV A,TT_LED
    CJNE A,#0,KTRA_LED6
    MOV A,B
    CLR ACC.7
    SJMP OUT_CHAM
KTRA_LED6:
    CJNE A,#5,OUT_PORT
    MOV A,B
    CLR ACC.7
    SJMP OUT_CHAM
OUT_PORT:
    MOV A,B
OUT_CHAM:
    MOV P0,A
    MOV A,TT_LED
    CJNE A,#8,KTRA_LED9
    MOV P2,#0FFH
    CLR P1.6
    SETB P1.7
    SJMP TANG_TTLED
KTRA_LED9:
    CJNE A,#9,LED_1_8
    MOV P2,#0FFH
    SETB P1.6
```

```
        CLR  P1.7
        SJMP TANG_TTLED

LED_1_8:
        LCALL      CHONLED
        SETB P1.6
        SETB P1.7
        CPL  A
        MOV  C,ACC.0
        MOV  P2.4,C
        MOV  C,ACC.1
        MOV  P2.3,C
        MOV  C,ACC.2
        MOV  P2.2,C
        MOV  C,ACC.3
        MOV  P2.1,C
        MOV  C,ACC.4
        MOV  P2.0,C
        MOV  C,ACC.5
        MOV  P2.7,C
        MOV  C,ACC.6
        MOV  P2.6,C
        MOV  C,ACC.7
        MOV  P2.5,C
TANG_TTLED:
        INC  TT_LED
        MOV  A,TT_LED
```

```
    CJNE A,#10D,EXIT_Q
    MOV TT_LED,#0D
EXIT_Q:
    POP  ACC
    RETI
DELAY:
    MOV R7,#100
LOOP:
    DJNZ R7,LOOP
    RET
MA7D:
    INC  A
    MOVC      A,@A+PC
    RET
 DB
    0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H,0BFH,0FFH

 CHONLED:
    INC  A
    MOVC      A,@A+PC
    RET
 DB      16,8,4,2,1,128,64,32

 END
```

KHỐI QUẦY

```
LED1        EQU  70H
LED2        EQU  71H
LED3        EQU  72H
LED4        EQU  73H
LED5        EQU  74H
LEDCHOP     EQU  75H
TT_LED      EQU  76H
BPHIM       EQU  77H

THU_DATA          BIT   70H
TRA_LOI_TT  BIT   71H

   ORG  0000H
   LJMP BEGIN
   ORG  000BH
   LJMP QUET_LED

   ORG  0023H
   PUSH ACC
   CLR  RI
   MOV  A,SBUF
   CLR  C
   SUBB A,#30H

;  MOV  LED2,A
```

```
    JB          THU_DATA,CAT_DATA       ;CAT DU LIEU MAY
TINH GOI

    CJNE A,#10H,KTRA_KTX
    SETB TRA_LOI_TT
    LJMP EXIT

KTRA_KTX:
    CJNE A,#28H,EXIT                    ;KY TU DAU = 1 LA
CHUOI SO PC GOI XUONG
    SETB THU_DATA
    LJMP EXIT

CAT_DATA:                               ;CAT VA HIEN
THI SO PC GOI XUONG
    MOV @R1,A
    INC  R1
    CJNE R1,#65H,EXIT
    MOV R1,#60H
    CLR  THU_DATA
    MOV A,60H
    CJNE A,#1,EXIT
    MOV LEDCHOP,#10D
    CLR  7FH
    MOV 70H,60H
    MOV 71H,61H
```

```
    MOV 72H,62H
    MOV 73H,63H
    MOV 74H,64H

EXIT:
    POP  ACC
    RETI

BEGIN:
    MOV TMOD,#21H
    MOV TH1,#-26
    SETB TR1
    MOV SCON,#50H
    clr  RI
    clr  TI
    SETB EA
    SETB ES
    SETB ET0

    MOV TL0,#18H          ;DELAY 1ms
    MOV TH0,#0FCH
    SETB TR0
    CLR  TF0
    MOV LED1,#10
    MOV LED2,#10
    MOV LED3,#10
    MOV LED4,#10
```

```
        MOV  LED5,#10
        MOV  R2,#0D
        MOV  R1,#60H
        MOV  LEDCHOP,#0D
        MOV  TT_LED,#0
        MOV  BPHIM,#0
        CLR  P3.2
        CLR  THU_DATA
        CLR  TRA_LOI_TT

MAIN:
        JNB  TRA_LOI_TT,KTRA_P10
        LCALL     DELAY
        SETB P3.2                          ;TRA  LOI  TRUNG
TAM
        CLR  ES
        LCALL     DELAY
        MOV  A,#30H
        ADD  A,BPHIM
        MOV  SBUF,A
        MOV  BPHIM,#0
        JNB  TI,$
        CLR  TI
        CLR  P3.2
        CLR  TRA_LOI_TT
        SETB ES
        SJMP MAIN
```

```
KTRA_P10:
    JB          P1.0,KTRA_P11
    MOV BPHIM,#11H
    SJMP MAIN
KTRA_P11:
    JB          P1.1,MAIN
    MOV BPHIM,#12H
    SJMP MAIN
```

```
QUET_LED:
    PUSH ACC
    MOV TL0,#18H
    MOV TH0,#0FCH

    INC  R2
    MOV A,R2
    CJNE A,#200D,QUET
    MOV R2,#0D
    CPL  7FH
    MOV A,LEDCHOP
    CJNE A,#00H,GIAM
    SETB 7FH
    SJMP QUET
```

```
GIAM:   DEC   LEDCHOP
QUET:   JB      7FH,MO_LED
   MOV  P0,#0FFH
   SJMP EXIT_Q

MO_LED:        MOV  P2,#0FFH
   MOV  P0,#0FFH
   MOV  A,TT_LED
   ADD  A,#70H
   MOV  R0,A
   MOV  A,@R0
   LCALL        MA7D
   MOV  P0,A
   MOV  A,TT_LED
   LCALL        CHONLED
   CPL   A
   MOV  C,ACC.0
   MOV  P2.0,C
   MOV  C,ACC.1
   MOV  P2.1,C
   MOV  C,ACC.2
   MOV  P2.2,C
   MOV  C,ACC.3
   MOV  P2.3,C
   MOV  C,ACC.4
   MOV  P2.4,C
   INC   TT_LED
```

```
    MOV  A,TT_LED
    CJNE A,#5D,EXIT_Q
    MOV  TT_LED,#0D
EXIT_Q:
    POP   ACC
    RETI


DELAY:
    MOV  R7,#100                    ;DELAY 30ms
LOOP:
    DJNZ R7,LOOP
    RET
MA7D:
    INC   A
    MOVC        A,@A+PC
    RET
  DB
    0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H,0BFH,0FFH,0F
FH,0FFH,0FFH,0FFH,82H

  CHONLED:
    INC   A
    MOVC        A,@A+PC
    RET
  DB        16,8,4,2,1
  END
```

CHƯƠNG TRÌNH TRÊN MÁY TÍNH BẰNG VISUAL BASIC

```
Option Expicit
Dim khkt As Integer
Dim tskh As Integer
Dim luu As String
Dim i As Long
Dim b(5) As Integer


Private Sub Form_Unload(Cancel As Integer)
        If (mscKetnoi.PortOpen = True) Then
                mscKetnoi.PortOpen = False
        End If
End Sub

Private Sub Form_Load()
        Dim fs As NewFileSystemObject
        Dim fi As File
        Dim ts As TextStream
        Dim duongdan As String
        Dim s As String
        duongdan = App.Path + "\data.txt"
        Set fi = fs.GetFile(duongdan)
        Set ts = fi.OpenAsTextStream(ForWriting)
        ts.Write("0")
        ts.Close
        txtTong.Text = "0000"
        txtNext.Text = "0000"

        mscKetnoi.CommPort = 1
        mscKetnoi.Settings = "9600,N,8,1"
        mscKetnoi.InputLen = 1
        mscKetnoi.PortOpen = True
End Sub

Private Sub mscKetnoi_Oncomm()
        Dim fs As NewFileSystemObject
        Dim fi As File
```

```
Dim ts As TextStream
Dim duongdan As String
Dim strTam As String
Dim s As String
Dim i As Integer
Dim sdata  As String
Dim Hbyte As Long
Dim Lbyte As  Long
Dim byte As Long
Dim Ktra  As String
Dim dataTruyenNew As String

If mscKetnoi.CommEvent = comEvReceive Then
        sdata = mscKetnoi.Input              'lay data
        Hbyte = Asc(Mid(sdata,1,1))          ' lay byte dau tien
        Lbyte = Asc(Mid(sdata,2,1))          ' lay byte thu hai
        txtDataReceive.Text = txtDataReceive.Text + " " + CStr(Hbyte) + " " +
        CStr(Lbyte)
        Ktra = CStr(byte)
        If CStr(Lbyte) = "5" Then
                If khkt <= tskk Then
                        strTam = dc + " " + Mid(luu,3,2) + " " + Mid(luu,1,2)
                        dataTruyenNew = " "
                If khkt < tskk Then
                        dataTruyenNew = Cstr(Hbyte) + " " + Mid(luu,3,2) + " " +
                        Mid(luu,1,2)
                Else
                        dataTruyenNew =  "0"
        End If
                Khkt = khkt + 1
End Sub

Private Sub cmd_Click()
        Dim strTam As String
                strTam = dc + " " + Mid(luu,3,2) + " " + Mid(luu,1,2)
End Sub

Private Sub cmdThem_Click()
        Dim fs As NewFileSystemObject
        Dim fi As File
```

```
        Dim ts As TextStream
        Dim duongdan As String
        Dim strTam As String
        Dim s As String
        Dim i As Integer

        duongdan = App.Path + "\data.txt"
        Set fi = fs.GetFile(duongdan)
        Set ts = fi.OpenAsTextStream(ForReading)
        strTam = ts.ReadAll
        ts.Close
        i = Val(strTam)
        i = i + 1
        tskh = i
        Set ts = fi.OpenAsTextStream(ForWriting)
        ts.Write(CStr(i))
        ts.Close
        If (txtTong.Text <> "0000" And txtNext.Text = "0000") Then
            txtNext.Text = "0001"
            khkt = 1
        End If
End Sub

Private Sub cmdTiep_Click()
        Dim strTam As String
        Dim tam As Byte
        Dim c As String
        Dim  ten As Variant
        b(0) = App.Path + "\moi.wav"
        b(1) = App.Path + "\so.wav"
        b(2) = App.Path + "\den.wav"
        b(3) = App.Path + "\quay.wav"

        If dc = "11" Then
            b(5) = App.Path + "\mot.wav"
        End If

        If dc = "21" Then
            b(5) = App.Path + "\hai.wav"
        End If
```

```
        If dc = "31" Then
                b(5) = App.Path + "\ba.wav"
        End If

        If dc = "41" Then
                b(5) = App.Path + "\bon.wav"
        End If

        strTam = txtNext.Text
        For tam = 1 To Len(strTam)
                c = Mid(strTam,tam,1)
                b(tam+1) = App.Path + "\" + c + ".wav"
        Next tam
                i = 0
                mp.FileName = aaa(0)
                mp.Play
        If khkt < tskh Then
                luu = txtNext.Text
                strTam = dc + " " + Mid(luu,3,2) + Mid (luu,1,2)
                khtt = khtt + 1
        End If
End Sub

Private Sub cmdThoat_Click()
        End
End Sub
```