BẢNG NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Giáo viên hướng dẫn: VÕ ĐÌNH TÙNG

Sinh viên thực hiện: TRẦN ANH TUẤN

Lớp: 08DDT1

MSSV: 0851010081

Tên đề tài: ***THIẾT KẾ VÀ THI CÔNG MẠCH ĐO VÀ ĐIỀU KHIỂN NHIỆT ĐỘ***

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm đánh giá : .....Xếp loại: .....

TP. Hồ Chí Minh, ngày tháng năm 2011

Giáo viên hướng dẫn

(Ký và ghi rõ họ tên)

BẢNG NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Giáo viên phản biện :

Họ tên sinh viên: TRẦN ANH TUẤN

Lớp: 08DDT1

MSSV: 0851010081

Tên đề tài: ***THIẾT KẾ VÀ THI CÔNG MẠCH ĐO VÀ ĐIỀU KHIỂN NHIỆT ĐỘ***

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm đánh giá : .....Xếp loại: .....

TP. Hồ Chí Minh, ngày tháng năm 2011

Giáo viên phản biện

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

******

- Lời đầu tiên em xin chân thành cảm ơn đến tất cả các thầy cô trong khoa cơ điện-điện tử trong thời gian qua đã tận tình giúp đỡ em để em thực hiện đồ án môn học 2. Đặc biệt là khoa Cơ Điện- Điện Tử đã tạo mọi điều kiện thuận lợi cho em để em hoàn thành đồ án môn học này.
- Em cũng vô cùng biết ơn đến thầy VÕ ĐÌNH TÙNG là người trực tiếp hướng dẫn và chỉ bảo cho em hết sức tật tình để em có thể hoàn thành đề tài thiết kế và thi công mạch ĐO VÀ ĐIỀU KHIỂN NHIỆT ĐỘ này.
- Với mong muốn học hỏi, em rất mong các thầy cô góp ý và hướng dẫn thêm cho em.

Em xin chân thành cảm ơn.

Ngày tháng năm 2011

SVTH: TRẦN ANH TUẤN

LỜI NÓI ĐẦU

******

- Ngày nay, con người cùng với những ứng dụng của khoa học kỹ thuật tiên tiến của thế giới, chúng ta đã và đang ngày một thay đổi, văn minh và hiện đại hơn.
- Sự phát triển của kỹ thuật điện tử đã tạo ra hàng loạt những thiết bị với các đặc điểm nổi bật như sự chính xác cao, tốc độ nhanh, gọn nhẹ…là những yếu tố rất cần thiết góp phần cho hoạt động của con người đạt hiệu quả ngày càng cao hơn.
- Điện tử đang trở thành một ngành khoa học đa nhiệm vụ. Điện tử đã đáp ứng được những đòi hỏi không ngừng của các ngành, lĩnh vực khác nhau cho đến nhu cầu thiết yếu của con người trong cuộc sống hàng ngày. Một trong những ứng dụng của rất quan trọng của ngành công nghệ điện tử là kỹ thuật đo và điều khiển nhiệt độ. Sử dụng cảm biến nhiệt được ứng dụng rất nhiều trong công nghiệp và các lĩnh vực khác trong cuộc sống với những thiết bị điều khiển nhiệt rất tinh vi.
- Xuất phát từ những ứng dụng đó, em đã thiết kế mạch đo và điều khiển nhiệt độ một trong những ứng dụng nhỏ của mạch đo và điều khiển nhiệt độ.
- Vì thời gian và trình độ còn hạn chế nên việc thực hiện đồ án còn nhiều thiếu sót … Kính mong nhận được sự chỉ dẫn và góp ý tận tình của tất cả quý thầy cô.
- Cuối cùng chúng em xin chân thành cảm ơn sự đóng góp ý kiến của tất cả quý thầy cô và sự nhiệt tình của các bạn đã giúp đỡ việc thực hiện đề tài trong suốt thời gian qua.

MỤC LỤC

Chương 1
KIẾN THỨC BỔ TRỢ

******

1.1 TỔNG QUAN VỀ VI ĐIỀU KHIỂN MCS-51

- Chương này giới thiệu tổng quan về họ vi điều khiển MCS-51(chủ yếu trên AT89C51): cấu trúc phần cứng, sơ đồ chân, các thanh ghi, đặc tính lập trình và các đặc tính về điện.

1.1.1 GIỚI THIỆU

- Họ vi điều khiển MCS-51 do Intel sản xuất đầu tiên vào năm 1980 là các IC thiết kế cho các ứng dụng hướng điều khiển. Các IC này chính là một hệ thống vi xử lý hoàn chỉnh bao gồm các các thành phần của hệ vi xử lý: CPU, bộ nhớ, các mạch giao tiếp, điều khiển ngắt.
- MCS-51 là họ vi điều khiển sử dụng cơ chế CISC (Complex Instruction Set Computer), có độ dài và thời gian thực thi của các lệnh khác nhau. Tập lệnh cung cấp cho MCS-51 có các lệnh dùng cho điều khiển xuất / nhập tác động đến từng bit.
- MCS-51 bao gồm nhiều vi điều khiển khác nhau, bộ vi điều khiển đầu tiên là 8051 có 4KB ROM, 128 byte RAM và 8031, không có ROM nội, phải sử dụng bộ nhớ ngoài. Sau này, các nhà sản xuất khác như Siemens, Fujitsu, … cũng được cấp phép làm nhà cung cấp thứ hai.
- MCS-51 bao gồm nhiều phiên bản khác nhau, mỗi phiên bản sau tăng thêm một số thanh ghi điều khiển hoạt động của MCS-51.

1.1.2 VI ĐIỀU KHIỂN AT89C51

- AT89C51 là vi điều khiển do Atmel sản xuất, chế tạo theo công nghệ CMOS có các đặc tính như sau:

+4 KB PEROM (Flash Programmable and Erasable Read Only Memory), 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

+128 Byte RAM nội.

+4 Port xuất /nhập I/O 8 bit.

+2 bộ Timer/counter 16 Bit.

+6 nguồn ngắt.

+Giao tiếp nối tiếp điều khiển bằng phần cứng.

+64 KB vùng nhớ mã ngoài

+64 KB vùng nhớ dữ liệu ngoài.

+Cho phép xử lý bit.

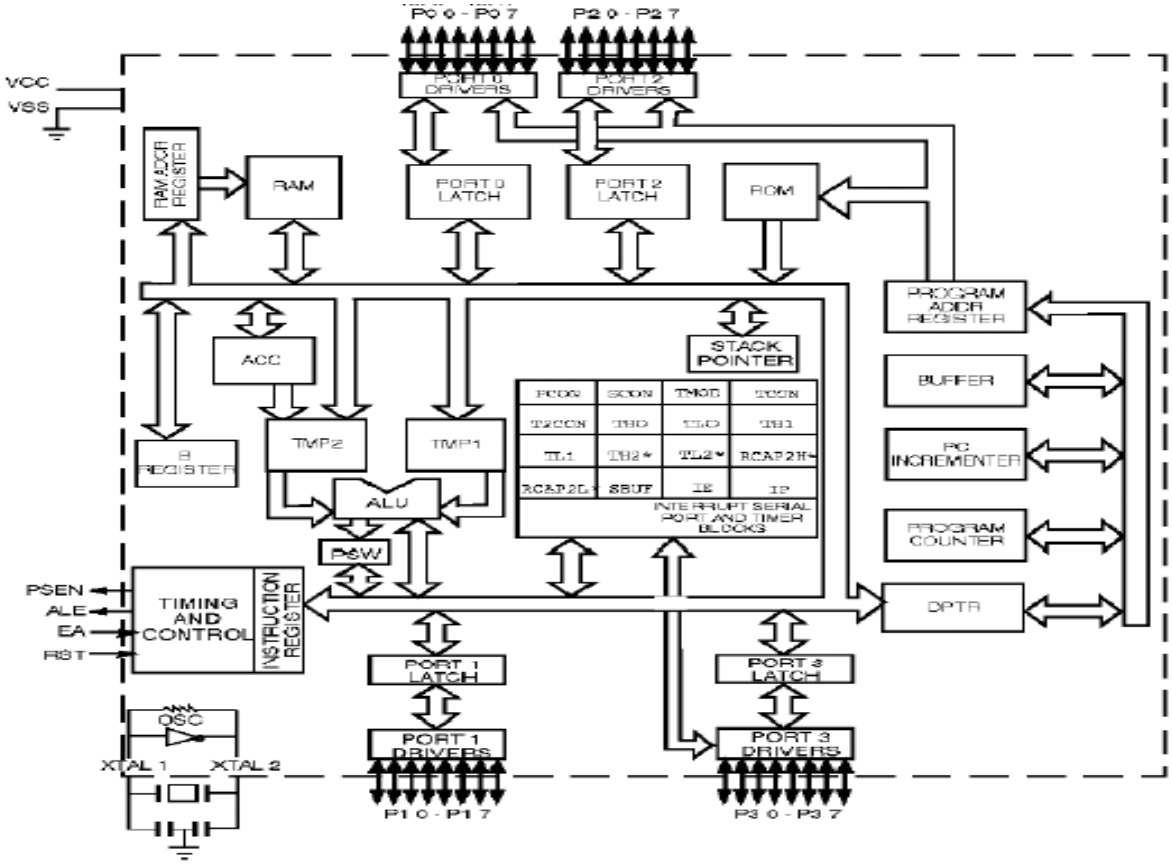
+210 vị trí nhớ có thể định vị bit.

+4 chu kỳ máy (4 μs đối với thạch anh 12MHz) cho hoạt động nhân hoặc chia.

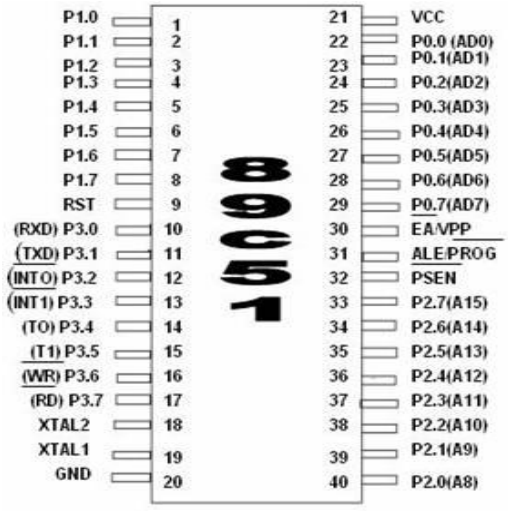
+Có các chế độ nghỉ (Low-power Idle) và chế độ nguồn giảm (Power-down).

- Ngoài ra, một số IC khác của họ MCS-51 có thêm bộ định thời thứ 3 và 256 byte RAM nội.

1.1.3 SƠ ĐỒ KHỐI AT89C51



1.1.4 SƠ ĐỒ CHÂN AT89C51



- **Port 0:** là port có 2 chức năng ở các chân 32 – 39 của AT89C51:

  - Chức năng IO (xuất / nhập): dùng cho các thiết kế nhỏ. Tuy nhiên, khi dùng chức năng này thì Port 0 phải dùng thêm các điện trở kéo lên (pull-up), giá trị của điện trở phụ thuộc vào thành phần kết nối với Port.

    +Khi dùng làm ngõ ra, Port 0 có thể kéo được 8 ngõ TTL.

    +Khi dùng làm ngõ vào, Port 0 phải được set mức logic 1 trước đó.

  - Chức năng địa chỉ / dữ liệu đa hợp: khi dùng các thiết kế lớn, đòi hỏi phải sử dụng bộ nhớ ngoài thì Port 0 vừa là bus dữ liệu (8 bit) vừa là bus địa chỉ (8 bit thấp).

  - Ngoài ra khi lập trình cho AT89C51, Port 0 còn dùng để nhận mã khi lập trình và xuất mà khi kiểm tra (quá trình kiểm tra đòi hỏi phải có điện trở kéo lên).

- **Port 1:** (chân 1 – 8) chỉ có một chức năng là IO, không dùng cho mục đích khác (chỉ trong 8032/8052/8952 thì dùng thêm P1.0 và P1.1 cho bộ định thời thứ 3).

  - Tại port 1 đã có điện trở kéo lên nên không cần thêm điện trở ngoài. Port 1 có khả năng kéo được 4 ngõ TTL và còn dùng làm 8 bit địa chỉ thấp trong quá trình lập trình hay kiểm tra.

  - Khi dùng làm ngõ vào, Port 1 phải được set mức logic 1 trước đó.

- **Port 2 :** (chân 21 – 28) là port có 2 chức năng:

  - Chức năng IO (xuất / nhập): có khả năng kéo được 4 ngõ TTL. Khi dùng làm ngõ vào, Port 2 phải được set mức logic 1 trước đó.

- Chức năng địa chỉ: dùng làm 8 bit địa chỉ cao khi cần bộ nhớ ngoài có địa chỉ
- 16 bit. Khi đó, Port 2 không được dùng cho mục đích IO.
- Khi lập trình, Port 2 dùng làm 8 bit địa chỉ cao hay một số tín hiệu điều khiển.

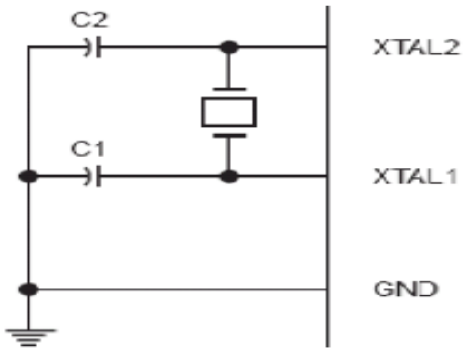
- **Port 3:** (chân 10 – 17) là port có 2 chức năng:
  - Chức năng IO: có khả năng kéo được 4 ngõ TTL. Khi dùng làm ngõ vào, Port 3 phải được set mức logic 1 trước đó.
  - Chức năng khác được mô tả trong bảng 1.1.

Bảng 1.1: Chức năng các chân của Port 3

Bit	Tên	Chức năng
P3.0	RxD	Ngõ vào port nối tiếp
P3.1	TxD	Ngõ ra port nối tiếp
P3.2	INT0	Ngắt ngoài 0
P3.3	INT1	Ngắt ngoài 1
P3.4	T0	Ngõ vào của bộ định thời 0
P3.5	T1	Ngõ vào của bộ định thời 1
P3.6	WR	Tín hiệu điều khiển ghi dữ liệu lên bộ nhớ ngoài.
P3.7	RD	Tín hiệu điều khiển đọc từ bộ nhớ dữ liệu ngoài.

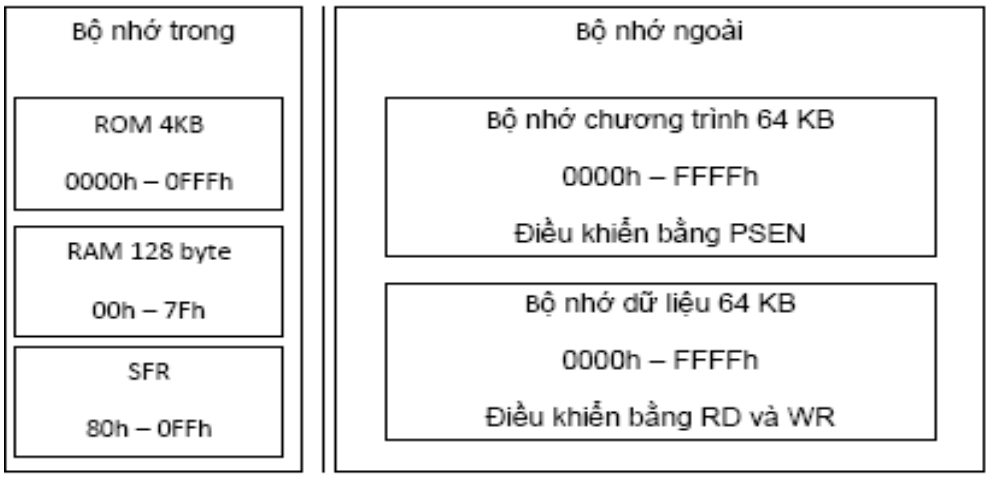
- **NGUỒN:**
  - Chân 40: VCC = 5V ± 20%
  - Chân 20: GND
- **PSEN (Program Store Enable):** (chân 29) cho phép đọc bộ nhớ chương trình mở rộng đối với các ứng dụng sử dụng ROM ngoài, thường được nối đến chân OC (Output Control) của ROM để đọc các byte mã lệnh. PSEN **sẽ** ở mức logic 0 trong thời gian AT89C51 lấy lệnh.Trong quá trình này, PSEN sẽ tích cực 2 lần trong 1 chu kỳ máy. Mã lệnh của chương trình được đọc từ ROM thông qua bus dữ liệu (Port0) và bus địa chỉ (Port0 + Port2).
  - Khi 8951 thi hành chương trình trong ROM nội, PSEN sẽ ở mức logic 1.

- **ALE/PROG (Address Latch Enable / Program):** (chân 30) cho phép tách các đường địa chỉ và dữ liệu tại Port 0 khi truy xuất bộ nhớ ngoài. ALE thường nối với chân Clock của IC chốt (74373, 74573).
  - Các xung tín hiệu ALE có tốc độ bằng 1/6 lần tần số dao động trên chip và có thể được dùng làm tín hiệu clock cho các phần khác của hệ thống. Xung này có thể cấm bằng cách set bit 0 của SFR tại địa chỉ 8Eh lên 1. Khi đó, ALE chỉ có tác dụng khi dùng lệnh MOVX hay MOVC. Ngoài ra, chân này còn được dùng làm ngõ vào xung lập trình cho ROM nội (PROG).
- **EA /VPP (External Access):** (chân 31) dùng để cho phép thực thi chương trình từ ROM ngoài. Khi nốichân 31 với Vcc, AT89C51 sẽ thực thi chương trình từ ROM nội (tối đa 8KB), ngượclại thì thực thi từ ROM ngoài (tối đa 64KB).
  - Ngoài ra, chân EA được lấy làm chân cấp nguồn 12V khi lập trình cho ROM.
- **RST (Reset)**: (chân 9) cho phép reset AT89C51 khi ngõ vào tín hiệu đưa lên mức 1 trong ít nhất là 2 chu kỳ máy.
- **X1,X2:** Ngõ vào và ngõ ra bộ dao động, khi sử dụng có thể chỉ cần kết nối thêm thạch anh và các tụ như hình vẽ trong sơ đồ. Tần số thạch anh thường sử dụng cho AT89C51 là 12Mhz.



Giá trị $C_1, C_2 = 30\ pF \pm 10\ pF$

1.1.5 TỔ CHỨC BỘ NHỚ



- Bộ nhớ của họ MCS-51 có thể chia thành 2 phần: bộ nhớ trong và bộ nhớ ngoài.
- Bộ nhớ trong bao gồm 4 KB ROM và 128 byte RAM (256 byte trong 8052). Các byte RAM có địa chỉ từ 00h – 7Fh và các thanh ghi chức năng đặc biệt (SFR) có địa chỉ từ 80h – 0FFh có thể truy xuất trực tiếp. Đối với 8052, 128 byte RAM cao (địa chỉ từ 80h – 0FFh) không thể truy xuất trực tiếp mà chỉ có thể truy xuất gián tiếp (xem thêm trong phần tập lệnh).
- Bộ nhớ ngoài bao gồm bộ nhớ chương trình (điều khiển đọc bằng tín hiệu PSEN**)** và bộ nhớ dữ liệu (điều khiển bằng tín hiệu RD hay WR để cho phép đọc hay ghi dữ liệu). Do số đường địa chỉ của MCS-51 là 16 bit (Port 0 chứa 8 bit thấp và Port 2 chứa 8 bit cao) nên bộ nhớ ngoài có thể giải mã tối đa là 64KB.

- **Tổ chức bộ nhớ trong:**
  - Bộ nhớ trong của MCS-51 gồm ROM và RAM. RAM bao gồm nhiều vùng có mục đích khác nhau: vùng RAM đa dụng (địa chỉ byte từ 30h – 7Fh và có thêm vùng 80h – 0FFh ứng với 8052), vùng có thể địa chỉ hóa từng bit (địa chỉ byte từ 20h – 2Fh, gồm 128 bit được định địa chỉ bit từ 00h – 7Fh), các bank thanh ghi (từ 00h – 1Fh) và các thanh ghi chức năng đặc biệt (từ 80h – 0FFh).
  - Các thanh ghi chức năng đặc biệt (SFR – Special Function Registers):

Bảng 1.2 – Các thanh ghi chức năng đặc biệt

Địa chỉ byte	Có thể định địa chỉ bit	Không định địa chỉ bit						
F8h								
F0h	B							
E8h								
E0h	ACC							
D8h								
D0h	PSW							
C8h	(T2CON)		(RCAP2L)	(RCAP2H)	(TL2)	(TH2)		
C0h								
B8h	IP	SADEN						
B0h	P3							
A8h	IE	SADDR						
A0h	P2							
98h	SCON	SBUF	BRL	BDRCON				
90h	P1							
88h	TCON	TMOD	TL0	TH0	TL1	TH1	AUXR	CKCON
80h	P0	SP	DPL	DPH				PCON

- Các thanh ghi có thể định địa chỉ bit sẽ có địa chỉ bit bắt đầu và địa chỉ byte trùng nhau. Ví dụ như: thanh ghi P0 có địa chỉ byte là 80h và có địa chỉ bit bắt đầu từ 80h (ứng với P0.0) đến 87h (ứng với P0.7). Chức năng các thanh ghi này sẽ mô tả trong phần sau.

+RAM nội: chia thành các vùng phân biệt: vùng RAM đa dụng (30h – 7Fh), vùng RAM có thể định địa chỉ bit (20h – 2Fh) và các bank thanh ghi (00h – 1Fh).

Địa chỉ byte	Địa chỉ bit								Chức năng
7F									Vùng RAM đa dụng
30									
2F	7F	7E	7D	7C	7B	7A	79	78	Vùng có thể định địa chỉ bit
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 18	Bank 3								Các bank thanh ghi
17 10	Bank 2								
1F 08	Bank 1								
07 00	Bank thanh ghi 0 (mặc định cho R0-R7)								

Hình 1.6 – Sơ đồ phân bố RAM nội

+RAM đa dụng: có 80 byte từ địa chỉ 30h – 7Fh có thể truy xuất mỗi lần 8 bit bằng cách dùng chế độ địa chỉ trực tiếp hay gián tiếp.

+RAM có thể định địa chỉ bit: vùng địa chỉ từ 20h – 2Fh gồm 16 byte (= 128 bit) có thể thực hiện giống như vùng RAM đa dụng (mỗi lần 8 bit) hay thực hiện truy xuất mỗi lần 1 bit bằng các lệnh xử lý bit. Vùng RAM này có các địa chỉ bit bắt đầu tại giá trị 00h và kết thúc tại 7Fh. Như vậy, địa chỉ bắt đầu 20h (gồm 8 bit) có địa chỉ bit từ 00h – 07h; địa chỉ kết thúc 2Fh có địa chỉ bit từ 78h – Fh.

+Các bank thanh ghi: vùng địa chỉ từ 00h – 1Fh được chia thành 4 bank thanh ghi: bank 0 từ 00h – 07h, bank 1 từ 08h – 0Fh, bank 2 từ 10h – 17h và bank 3 từ 18h – 1Fh. Các bank thanh ghi này được đại diện bằng các thanh ghi từ R0 đến R7. Sau khi khởi động hệ thống thì bank thanh ghi được sử dụng là bank 0. Do có 4 bank thanh ghi nên tại một thời điểm chỉ có một bank thanh ghi được truy xuất

bởi các thanh ghi R0 đến R7. Việc thay đổi bank thanh ghi có thể thực hiện thông qua thanh ghi từ trạng thái chương trình (PSW). Các bank thanh ghi này cũng có thể truy xuất bình thường như vùng RAM đa dụng đã nói ở trên.

- **Tổ chức bộ nhớ ngoài:**

- MCS-51 có bộ nhớ theo cấu trúc Harvard: phân biệt bộ nhớ chương trình và dữ liệu. Chương trình và dữ liệu có thể chứa bên trong nhưng vẫn có thể kết nối với 64KB chương trình và 64KB dữ liệu. Bộ nhớ chương trình được truy xuất thông qua chân PSEN còn bộ nhớ dữ liệu đươc truy xuất thông qua chân WR hay RD. Lưu ý rằng việc truy xuất bộ nhớ chương trình luôn sử dụng địa chỉ 16 bit còn bộ nhớ dữ liệu có thể là 8 bit hay 16 bit tuỳ theo câu lệnh sử dụng. Khi dùng bộ nhớ dữ liệu 8 bit thì có thể dùng Port 2 như Port I/O thông thường còn khi dùng ở chế độ 16 bit thì Port 2 chỉ dùng làm các bit địa chỉ cao. Port 0 được dùng làm địa chỉ thấp/ dữ liệu đa hợp. Tín hiệu ALE để tách byte địa chỉ và đưa vào bộ chốt ngoài.
- Trong chu kỳ ghi, byte dữ liệu sẽ tồn tại ở Port 0 vừa trước khi WR tích cực và được giữ cho đến khi WR không tích cực.Trong chu kỳ đọc, byte nhận được chấp nhận vừa trước khi RD không tích cực.
- Bộ nhớ chương trình ngoài được xử lý 1 trong 2 điều kiện sau:

+ Tín hiệu EA tích cực (= 0).
+ Giá trị bộ đếm chương trình (PC: Program Counter) lớn hơn kích thước bộ nhớ.

1.1.6 CÁC NHÓM LỆNH TRONG 89C51

➢ *Các chi tiết thiết lập lệnh:*

Rn :thanh ghi R0 đến R7 của bank thanh ghi được chọn.

Data :8 bit địa chỉ vùng dữ liệu bên trong. Nó có thể là vung ram dữ liệu trong (0 – 127) hoặc các thanh ghi chức năng đặc biệt.

@Ri : 8 bit vùng ram dữ liệu trong (0-125) được đánh giá địa chỉ gián tiếp qua thanh ghi R0 hoặc R1

#data :hằng 8 bit chức trong câu lệnh

#data 16 :hằng 16 bit chứa trong câu lệnh

Addr 16 :16 bit địa chỉ đích được dung trong câu lệnh LCALL và LJMP

Addr11 :11 bit địa chỉ đích được dung trong câu lênh LCALL và AJMP

Rel :byte offset 8 bit có dấu được dung trong câu lệnh SJMP và những lệnh nhảy có điều kiện.

Bit :Bit được định địa chỉ trực tiếp trong RAM dữ liệu nội hoặc các thanh ghi chức năng đặc biệt.

- *Nhóm lệnh xử lý toán học:* "(**số byte , chu kỳ máy**)"

ADD A,Rn (1,1): cộng thanh ghi Rn vào thanh ghi A.

ADD A,data (2,1):Cộng trực tiếp vào thanh ghi A.

ADD A,@Ri (1,1):Cộng gián tiếp nội dung RAM chứa tại địa chỉ được khai báo trong Ri vào thanh ghi A

ADD A,#data (2,1):Cộng dữ liệu tức thời vào A

ADD A,Rn (1,1):Cộng thanh ghi và cờ nhớ vào A.

ADD A,data (2,1):Cộng trực tiếp byte dữ lệu và cờ nhớ vào A.

ADDC A,@Ri (1,1):Cộng gián tiếp nội dung RAM và cờ nhớ vào A.

ADDC A,#data (2,1):Cộng dữ liệu tức thời và cờ nhớ vào A.

SUBB A,Rn (1,1):Trừ nội dung thanh ghi A cho nội dung thanh ghi Rn và cờ nhớ.

SUBB A,data (2,1):Trừ trực tiếp A cho một số và cờ nhớ.

SUBB A,@Ri (1,1):Trừ gián tiếp A cho một số và cờ nhớ.

SUBB A,data (2,1):Trừ nội dung A cho một số tức thời và cờ nhớ.

INC A (1,1):Tăng nội dung thanh ghi A lên 1.

INC Rn (1,1):tăng nội dung thanh ghi Rn lên 1.

INC data (2,1):Tăng dữ liệu trực tiếp lên 1.

INC @Ri (1,1):Tăng gián tiếp vùng RAM lên 1.

DEC A (1,1):Giảm nội dung thanh ghi A xuống 1.

DEC Rn (1,1);Giảm nội dung thanh ghi Rn xuống 1.

DEC data (2,1):Giảm dữ liệu trực tiếp xuống 1.

DEC @Ri (1,1);Giảm gián tiếp nội dung vùng RAM xuống 1.

INC DPTR (1,2); Tăng nội dung con trỏ dữ liệu lên 1.

MUL AB (1,4):Nhân nội dung thanh ghi A cho nội dung thanh ghi B.

DIV AB (1,4): Chia nội dung thanh ghi A cho nội dung thanh ghi B.

- *Nhóm lệnh luân lý:*

ANL A,Rn (1,1):AND nội dung thanh ghi A với nội dung thanh ghi Rn.

ANL A,data (1,1):AND nội dung thanh ghi A với dữ liệu trực tiếp.

ANL A,@Ri (1,1):AND nội dung thanh ghi A với nội dung gián tiếp trong RAM.

ANL A,#data (2,1):AND nội dung thanh ghi với dữ liệu tức thời.

ANL data,A (2,1):AND một dữ liệu trực tiếp với A

ANL data,#data (3,2):AND một dữ liệu trực tiếp với A một dữ liệu tức thời .

ANL C,bit (2,2):AND cờ nhớ với một bit trực tiếp

ANL C,/bit (2,2):AND cờ nhớ với bù 1 bit trực tiếp

ORL A,Rn (1,1):Or thanh ghi A với thanh ghi Rn.

ORL A,data (2,1):Or thanh ghi A với một dữ liệu trực tiếp

ORL A,@Ri (1,1):Or thanh ghi A với một dữ liệu gián tiếp

ORL A,#data (2,1):OR thanh ghi A với một dữ liệu tức thời.

ORL data,A (2,1):OR một dữ liệu trực tiếp với thanh ghi A

ORL data,#data (2,1): OR một dữ liệu trực tiếp với một dữ liệu tức thời.

ORL C,bit (2,2):OR cờ nhớ với một bit trực tiếp.

ORL C,/bit (2,2):OR cờ nhớ với bù của một bit trực tiếp.

XRL A,Rn (1,1):XOR thanh ghi A với thanh ghi Rn.

XRL A,data (2,1):XOR thanh ghi A với một dữ liệu trực tiếp.

XRL A,@Ri (1,1):XOR thanh ghi A với một dữ liệu gián tiếp.

XRL A,#data (2,1):XOR thanh ghi A với một dữ liệu tức thời.

XRL data,A (2,1): XOR một dữ liệu trực tiếp với thanh ghi A.

XRL data,#data (3,1):): XOR một dữ liệu trực tiếp với một dữ liệu tức thời.

SETB C (1,1):Đặt cờ nhớ.

SETB bit (2,1):Đặt một bit trực tiếp.

CLR A (1,1):Xóa thanh ghi A.

CLR C (1,1):Xóa cờ nhớ.

CPL A (1,1):Bù nội dung thanh ghi A.

CPL C (1,1):Bù cờ nhớ.

CPL bit (2,1):Bù một bit trực tiếp.

RL A (1,1):Quay nội dung thanh ghi A.

RLCA (1,1):Quay trái thanh ghi A qua cờ nhớ.

RR A (1,1):Quay phải nội dung thanh ghi A

RRC A (1,1):Quay trái nội dung thanh ghi A có cờ nhớ.

SWAP (1,1):Quay trái nội dung thanh ghi A 1 nibble(1/2 byte).

- *Nhóm lệnh chuyển dữ liệu:*

MOV A,Rn (1,1):Chuyển nội dung thanh ghi Rn vào thanh ghi A.

MOV A,data (2,1):Chuyển dữ liệu trực tiếp vào thanh ghi A.

MOV A,@Ri (1,1):Chuyển dữ liệu gián tiếp vào thanh ghi A.

MOV A,#data (2,1)chuyển dữ liệu tức thời vào thanh ghi A.

MOV Rn,data (2,2):Chuyển dữ liệu trực tiếp vào thanh ghi Rn.

MOV Rn,#data (2,2):Chuyển dữ liệu tức thời vào thanh ghi Rn.

MOV data,A (2,1):Chuyển nội dung thanh ghi A vào một dữ liệu trực tiếp.

MOV data,Rn (2,2):Chuyển nội dung thanh ghi Rn vào một dữ liệu trực tiếp.

MOV data,data (2,2):Chuyển một dữ liệu trực tiếp vào một dữ liệu trực tiếp.

MOV data,@Ri (2,2):Chuyển một dữ liệu gián tiếp vào một dữ liệu gián tiếp.

MOV data,#data (3,2): Chuyển một dữ liệu tức thời vào một dữ liệu trực tiếp.

MOV @Ri,A (1,1):Chuyển nội dung thanh ghi A vào một dữ liệu gián tiếp.

MOV @Ri,data (1,1):Chuyển một dữ liệu trực tiếp vào một dữ liệu gián tiếp.

MOV @Ri,#data (1,1):Chuyển một dữ liệu tức thời vào một dữ liệu gián tiếp.

MOV DPTR,#data(3,2):Chuyển một hằng 16 bit vào thanh ghi con trỏ dữ liệu.

MOV C,bit (2,1):Chuyển một bit trực tiếp vào cờ nhớ

MOV bit,C (2,2):Chuyển cờ nhớ vào một bit trực tiếp.

MOV A,@A+DPTR(1,2):Chuyển byte bộ nhớ chương trình có địa chỉ là @A+DPTR vào thanh ghi A.

MOVC A,@A+PC(1,2): Chuyển byte bộ nhớ chương trình có địa chỉ là @A+PC vào thanh ghi A.

MOV A,@Ri (1,2):Chuyển dữ liệu ngoài (8 bit địa chỉ) vào thanh ghi A

MOVX A,@DPTR(1,2):Chuyển dữ liệu ngoài (16 bit địa chỉ) vào thanh ghi A

MOVX @Ri,A (1,2):Chuyển nội dung A ra dữ liệu ngoài (8 bit địa chỉ).

MOVX @DPTR,A (1,2):Chuyển nội dung A ra dữ liệu bên ngoài (16 bit địa chỉ)

PUSH data (2,2):Chuyển dữ liệu trực tiếp vào ngăn xếp và tăng SP

POP data (2,2):Chuyển dữ liệu trực tiếp vào ngăn xếp và giảm SP

XCH A,Rn (1,1):Trao đổi dữ liệu giữa thanh ghi Rn Và thanh ghi A

XCH A,data (2,1):trao đổi giữa thanh ghi A với dữ liệu trực tiếp

XCH A,@Ri (1,1):trao đổi giữa thanh ghi A với dữ liệu gián tiếp

XCHD,@R (1,1):trao đổi giữa nibble thấp (LSN) của thanh ghi A và LSN của dữ liệu gián tiếp.

- *Nhóm lệnh chuyển điều khiển:*

ACALL addr11 (2,2):Gọi chương trình con dùng địa chỉ tuyệt đối

LCALL addr16 (3,2):Gọi chương trình con dùng địa chỉ dài

RET (1,2):Trở về từ lệnh gọi chương trình con

RET1 (1,2):Trở về từ lệnh gọi ngắt

AJMP addr11 (2,2):Nhảy tuyệt đối

LJMP addr16 (3,2):Nhảy dài

SJMP rel (2,2):Nhảy ngắn

JMP @A+DPTR (1,2):Nhảy gián tiếp từ con trỏ dữ liệu

JZ rel (2,2):Nhảy nếu A=0

JNZ rel (2,2):Nhảy nếu A không bằng 0

JC rel (2,2):Nhảy nếu cờ nhớ được đặt

JNC rel (2,2):Nhảy nếu cờ nhớ không được đặt

JB bit, rel (3,2):Nhảy tương đối nếu bit trực tiếp được đặt

JNB bit,rel (3,2):Nhảy tương đối nếu bit trực tiếp không được đặt

JBC bit,rel (3,2):Nhảy tương đối nếu bit trực tiếp được đặt rồi xóa bit

CJNE A,data,rel (3,2):So sánh dự liệu trực tiếp với A và nhảy nếu không bằng

CJNE A,#data,rel (3,2):So sánh dữ liệu tức thời với A và nhảy nếu không bằng

CJNE Rn,#data,rel (3,2):So sánh dữ liệu tức thời với nội dung thanh ghi Rn và nhảy nếu không bằng.

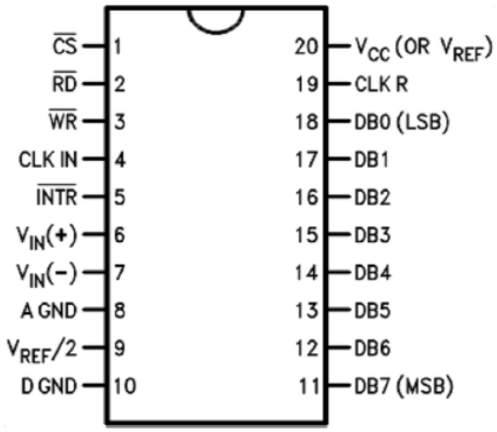
CJNE @Ri,#data,rel(3,2):So sánh dữ liệu tức thời với dữ liệu gián tiếp và nhảy nếu không bằng.

DJNZ Rn,rel (3,2):Giảm thanh ghi Rn và nhảy nếu không bằng

DJNZ data,rel (3,2):Giảm dữ liệu trực tiếp và nhảy nếu không bằng.

1.2 ADC 0804

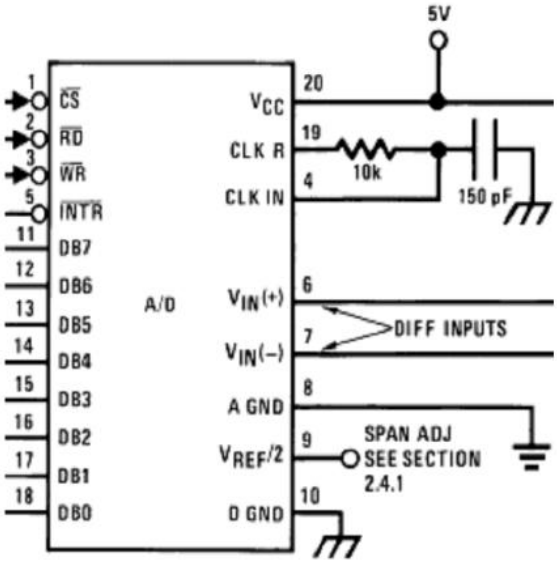
- Chip ADC0804 là bộ chuyển đổi tương tự sang số thuộc họ ADC800 của hãng National Semiconductor. Chip này cũng được nhiều hãng khác sản xuất. Chip có điện áp nuôi +5V v à độ phân giải 8 bit.



- Ngoài độ phân giải thì thời gian chuyển đổi cũng là một tham số quan trọng khi đánh giá bộ ADC. Thời gian chuyển đổi được định nghĩa là thời gian mà bộ ADC cần để chuyển một đầu vào tương tự thành một số nhị phân. Đối với ADC0804 th ì thời gian chuyển đổi phụ thuộc vào tần số đồng hồ được cấp tới chân CLK và CLK IN và không bé hơn 110μs.
- Các chân khác của ADC0804 có chức năng như sau:

+ **CS (Chip select):** (chân số 1) là chân chọn Chip, đầu vào tích cực mức thấp được sử dụng để kích hoạt chip ADC0804. Để truy cập ADC0804 thì chân này phải ở mức thấp.
+ **RD (Read):** (chân số 2) là một tín hiệu vào, tích cực ở mức thấp. Các bộ chuyển đổi đầu v ào tương tự thành số nhị phân và giữ nó ở một thanh ghi trong. RD đ ược sử dụng để có dữ liệu đã được chyển đổi tới đầu ra của ADC0804. Khi CS = 0 nếu có một xung cao xuống thấp áp đến chân RD thì dữ liệu ra dạng số 8 bit được đưa tới các chân dữ liệu (DB0 – DB7).

+ **WR (Write):** (chân số 3) đây là chân vào tích cực mức thấp được dùng để báo cho ADC biết bắt đầu quá trình chuyển đổi. Nếu CS = 0 khi WR tạo ra xung cao xuống thấp thì bộ ADC0804 bắt đầu quá trình chuyển đổi giá trị đầu vào tương tự Vin về số nhị phân 8 bit. Khi việc chuyển đổi hoàn tất thì chân INTR được ADC hạ xuống mức thấp.



+ **CLK IN và CLK R:** (chân số 4) và (chân số 19)
  - CLK IN là chân nối tới đồng hồ ngoài được sử dụng để tạo thời gian.
  - Tuy nhiên ADC0804 cũng có một bộ tạo xung đồng hồ riêng. Để dùng đồng hồ riêng thì các chân CLK IN và CLK R được nối với một tụ điện và một điện trở như hình vẽ. Khi ấy tần số được xác định bằng biểu thức:

$$f = \frac{1}{1.1RC}$$

  - Với R=10 kΩ, C=150pF và tần số f=606 kHz và thời gian chuyển đổi là 110μs.
+ **Ngắt INTR (Interupt):** (chân số 5) là chân ra tích cực mức thấp. Bình thường chân này ở trạng thái cao, khi việc chuyển đổi hoàn tất thì nó được ADC kéo xuống thấp để báo cho CPU biết l là dữ liệu chuyển đổi sẵn sàng để lấy đi. Khi INTR xuống thấp, cần đặt CS = 0V và gửi một xung cao xuống thấp tới chân RD để lấy dữ liệu.
+ **Vin (+) và Vin (-):** (chân số 6) và (chân số 7) đây là 2 đầu vào tương tự vi sai, trong đó Vin = Vin (+) – Vin (–). Thông thường Vin (–) được nối tới đất và Vin (+) được dùng làm đầu vào tương tự và sẽ được chuyển đổi về dạng số.
+ **Vcc:** (chân số 20) là chân nguồn nuôi +5V. Chân này còn được dùng làm điện áp tham chiếu khi đầu vào Vref/2 để hở.
+ **Vref/2:** (chân số 9) là chân điện áp đầu vào được dùng làm điện áp tham chiếu. Nếu để hở thì đầu vào tương tự có điện áp tham chiếu 0→5V. Khi ứng dụng có đầu vào

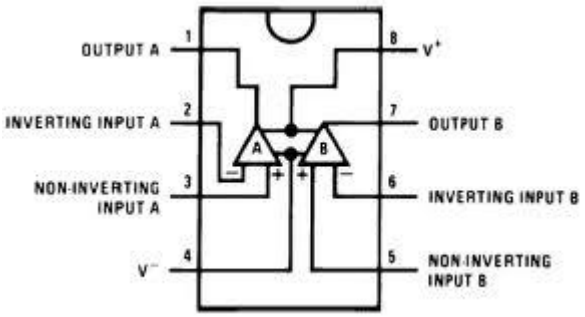
tương tự cần tham chiếu với dải điện áp khác thì chân Vref/2 được dùng để thực hiện các điện áp tham chiếu tùy chọn 0→Vref.

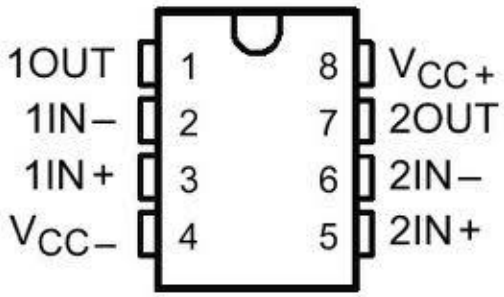
Bảng quan hệ điện áp $V_{ref/2}$ với V_{in}

$V_{ref/2}$ (V)	V_{in} (V)	Kích thước bước (mV)
Hở	0 – 5	5/256 = 19.53
2.0	0 – 4	4/256 = 15.62
1.5	0 – 3	3/256 = 11.71
1.28	0 – 2.56	2.56/256 = 10
1.0	0 – 2	2/256 = 7.81
0.5	0 – 1	1/256 = 3.90

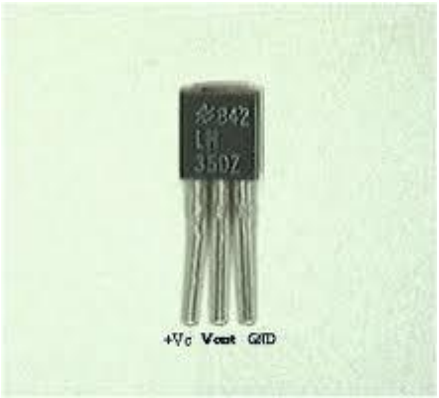
+ **D0 - D7:** (chân 18 – 11) là các chân ra dữ liệu số (D7 là bit cao nhất MSB và D0 là bit thấp nhất LSB). Các chân này được đệm ba trạng thái và dữ liệu đã được chuyển đổi chỉ được truy cập khi chân CS = 0 và chân RD đưa xuống mức thấp. Để tính điện áp đầu ra ta tính theo công thức: V_{in} D_{out} = Kích thước bước : (với D_{out} là số bít ngõ ra)

1.3 TL082





- TL082 bên trong bao gồm 2 opam với các chân V_+ , V_- , nguồn dương, nguồn âm , V_{out} như hình trên.



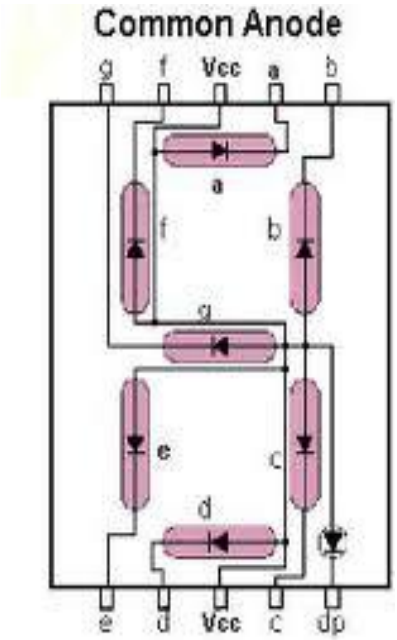
1.4 LM35

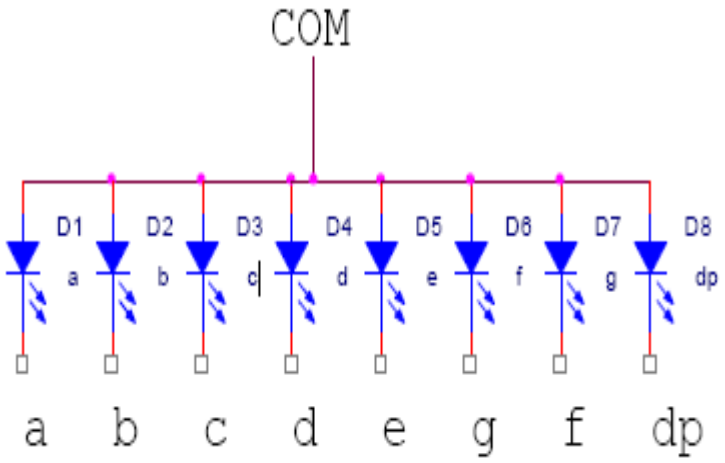
- Để đo nhiệt độ được chính xác ,tất nhiên cần có một đầu dò thích hợp. Đầu dò là một cảm biến nhiệt độ có nhiệm vụ chuyển đổi từ nhiệt độ sang tín hiệu điện. Có rất nhiều loại cảm biến nhiệt nhưng dựa vào lý thuyết và thực tế của mạch cần thiết kế ta dùng IC cảm biến nhiệt độ.

- Các IC cảm biến nhiệt độ có độ chính xác cao, dễ tìm, giá thành rẻ. Một trong số đó là IC LM35, là loại thông dụng trên thị trường hiện nay, đồng thời nó có những đặc tính làm việc phù hợp mạch cần thiết kế. Một số đặc tính làm việc của LM35:
  + Có độ biến thiên điện áp ra (V_{out}) theo nhiệt độ là 10mV/1^{o}C.
  + Điện áp 0V tương ứng 0^{o}C.
  + Có độ chính xác cao và nhạy. Ở nhiệt độ 25^{o}C có sai số không quá 0,5^{o}C.
  + Có tầm đo từ - 55^{o}C đến 150^{o}C ,tín hiệu ra tuyến tính liên tục với những thay đổi của nhiệt độ.
  + Tiêu tán công suất thấp.
  + Điện áp làm việc từ 4V-30V và đã được tích hợp bộ định dòng làm việc bên trong nên cần điện trở hạn dòng bên ngoài.

1.5 LED 7 ĐOẠN

- Led Anode chung





- Đối với dạng Led anode chung, chân COM phải có mức logic 1 và muốn sáng Led nào thì tương ứng các chân a – f, dp sẽ ở mức logic 0.
- Bảng mã cho Led Anode chung:

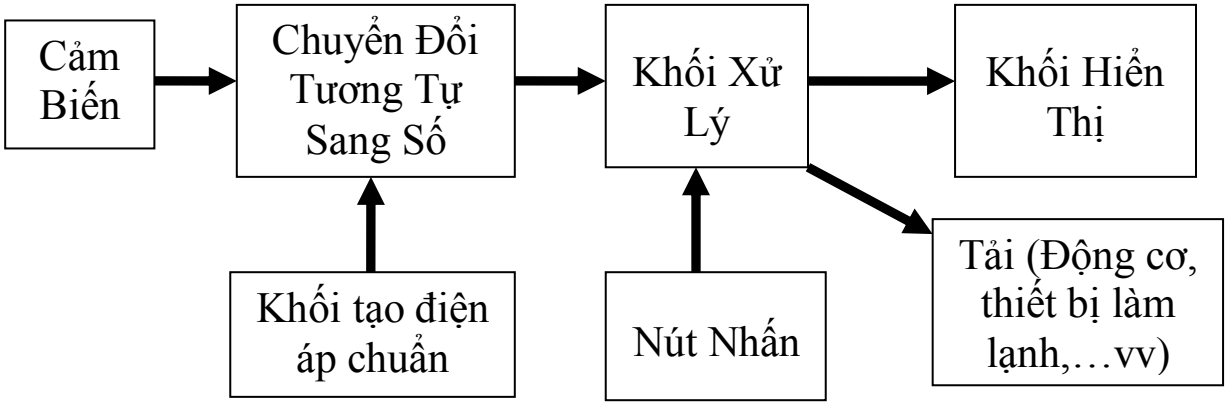
Led / Ký tự	a	b	c	d	e	f	g	dp	a là MSB, dp là LSB	a là LSB, dp là MSB
0	0	0	0	0	0	0	1	1	03h	0C0h
1	1	0	0	1	1	1	1	1	9Fh	0F9h
2	0	0	1	0	0	1	0	1	25h	0A4h
3	0	0	0	0	1	1	0	1	0Dh	0B0h
4	1	0	0	1	1	0	0	1	99h	99h
5	0	1	0	0	1	0	0	1	49h	92h
6	0	1	0	0	0	0	0	1	41h	82h
7	0	0	0	1	1	1	1	1	1Fh	0F8h
8	0	0	0	0	0	0	0	1	01h	80h
9	0	0	0	0	1	0	0	1	09h	90h
º	0	0	1	1	1	0	0	1	h	h
C	0	1	1	0	0	0	1	1	h	h
.	1	1	1	1	1	1	1	0	h	h

Chương 2

TÍNH TOÁN THIẾT KẾ VÀ THI CÔNG MẠCH

******

2.1 THIẾT KẾ SƠ ĐỒ KHỐI



2.2 NGUYÊN LÝ HOẠT ĐỘNG VÀ NHIỆM VỤ TỪNG KHỐI

2.2.1 NGUYÊN LÝ HOẠT ĐỘNG:

- Khối cảm biến chuyển đổi giá trị nhiệt độ thành giá trị điện áp đưa đến khối chuyển đổi tương tự sang số. Khối chuyển đổi tương tự sang số chuyển đổi giá trị điện áp thành giá trị số dựa vào việc so sánh với giá trị điện áp chuẩn mà khối tạo điện áp chuẩn tạo ra và số bít đầu ra. Giá trị số của nhiệt độ được đưa đến khối xử lý để so sánh với giá trị cài đặt (mặc định của chương trình hoặc có thể thay đổi giá trị cài đặt bằng các nút nhấn) để điều khiển tải và chuyển đổi giá trị giá trị số của nhiệt độ thành mã led để hiển thị.

2.2.2 KHỐI CẢM BIẾN:

- Có nhiệm vụ cảm biến nhiệt độ môi trường xung quanh và biến nhiệt độ đó thành đại lượng điện áp ổn định và thay đổi tuyến tính theo nhiệt độ môi trường.

2.2.3 KHỐI TẠO ĐIỆN ÁP CHUẨN:

- Có nhiệm vụ tạo ra một điện áp ổn định để cấp đến khối chuyển đổi tương tự sang số làm thước đo cho tín hiệu điện áp ra cua khối cảm biến.

2.2.4 KHỐI CHUYỂN ĐỔI TƯƠNG TỰ SANG SỐ:

- Làm nhiệm vụ so sánh giá trị tương tự mà khối cảm biến đưa ra với điện áp thước đo chuẩn mà khối tạo điện áp chuẩn đưa đến. Và sau đó dựa trên số bít đầu ra mà qui đổi giá trị đầu ra của khối cảm biến thành giá trị dưới dạng số nhị phân tương ứng cấp đến khối vi xử lý.

2.2.5 KHỐI XỬ LÝ VÀ KHỐI NÚT NHẤN:

- Khối xử lý làm nhiệm vụ đọc giá trị nhị phân mà khối chuyển đổi tương tự sang số đưa đến và so sánh nó với giá trị nhiệt độ cài đặt để điều khiển tải, đồng thời đổi giá trị nhị phân đó thành giá trị dưới dạng mã BCD để đưa đến khối hiển thị. Khối xử lý còn đọc các tín hiệu từ các nút nhấn để điều khiển quá trình xử lý như: bắt đầu lại quá trình xử lý trở về với các giá trị mặc định, hoặc thay đổi và cài đặt giá trị nhiệt độ chuẩn khác với giá trị mặc định.
- Khối nút nhấn làm nhiệm vụ đưa tín hiệu tướng ứng với phím nhấn tương ứng đến khối xử lý để điều khiển quá trình xử lý.

2.2.6 KHỐI HIỂN THỊ:

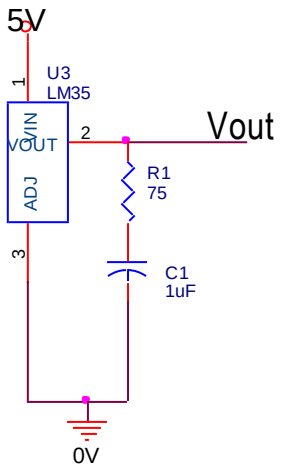
- Làm nhiệm vụ hiển thị giá trị nhiệt độ mà khối cảm biến đọc được.

2.2.7 KHỐI TẢI:

- Làm nhiệm vụ mở hoặc tắt quạt theo điều khiển của khối xử lý.

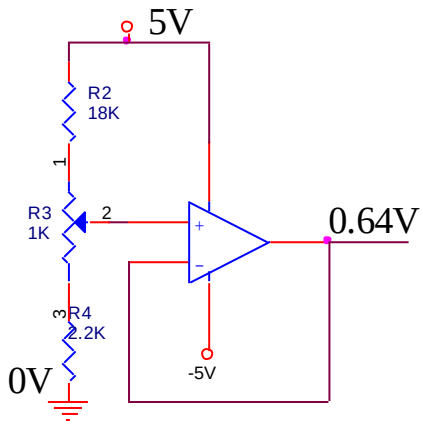
2.3 TÍNH TOÁN THIẾT KẾ MẠCH

2.3.1 KHỐI CẢM BIẾN:



- Do đặc điểm yêu cầu của mạch nên ta dùng IC cảm biến nhiệt độ LM35. Nó có khoảng đo từ - 55^{o}C đến 150^{o}C nếu ta cấp nguồn dương 5V cho chân 1và nguồn âm 5V cho chân 3. Nhưng yêu cầu của mạch nên chân 3 của LM35 ta nối mass ⇒khoảng đo từ 0^{o}C đến 150^{o}C
- Điện trở 75 ohm và tụ 1uF tạo thành mạch lọc thông thấp với tần số cắt $f_c = \frac{1}{RC} = \frac{1}{75x10^{-6}}$ = 13 Khz dùng để chống nhiễu cho tín hiệu đầu ra của LM35 (Datasheet của LM35 hướng dẫn).

2.3.2 KHỐI TẠO ĐIỆN ÁP CHUẨN:



- Để tạo điện áp chuẩn cho Vref ta sử dụng sơ đồ mạch bên:
- Để điện áp đầu ra ổn định ta chọn tổng trở R = R2+R3+R4 lớn do đó ta chọn R=20k
- Để dễ điều chỉnh ta chọn biến trở là 1k
- Ta có điện áp nguồn V=5V

⇨ $I_R=\frac{5}{20}=0{,}25$ mA

Gọi $R_{dưới}$ là điện trở từ chân 2 của biến trở xuống mass

⇨ $R_{dưới}=\frac{0.64}{0.25}=2{,}56$ k

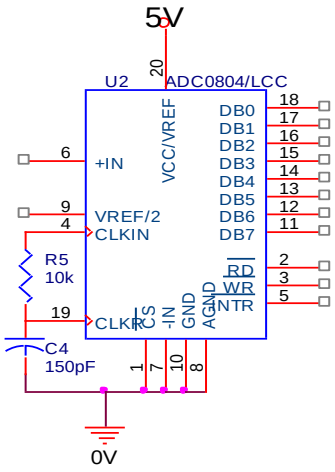
⇨ Chọn R4=2,2k

⇨ Chọn R2=18k

- Opamp ta dùng làm bộ đệm không đảo khống chế điện áp, ra chống tụt áp ⇒ vì vậy ta nối ngõ ra với V- để hồi tiếp.

2.3.3 KHỐI CHUYỂN ĐỔI TƯƠNG TỰ SANG SỐ:

- Sơ đồ nối chân cho ADC:



Ta có công thức tính xung clock cấp vào cho ADC:

$F=\frac{1}{1.1RC}$

Mà tần số hoạt động của ADC là 600 – 700 Khz.

Chọn f= 660 Khz ;R =10K

$=>C=\frac{1}{1{,}1.10000.660000}=137.10^{-12}F=137\text{pF}$

Chọn C = 150 pF

Để có thể truy cập ADC thì chân CS phải ở mức thấp do đó ta nói chân CS xuống mass.

- Điện áp cấp vào chân V_{ref} :

+Ta có nhiệt độ đo là khoảng từ 0^oC đến 150^oC nên ta có điện áp ra của LM35 là từ 0V đến 1.5V (vì LM35 có điện áp ra là $10mV/1^oC$).

+Ta có ngõ ra của ADC là 8 bit dạng số nhị phân ⇒ có 2^8=256 mức giá trị

+Mức giá trị cao nhất 255 ứng với V_{ref} =1.5V, mức giá trị thấp nhất 0 ứng với 0V

+Mỗi mức giá trị ứng với $\frac{V_{ref}}{256} = \frac{1.5}{256} = 5.86*10^{-3}V = 5.86mV$

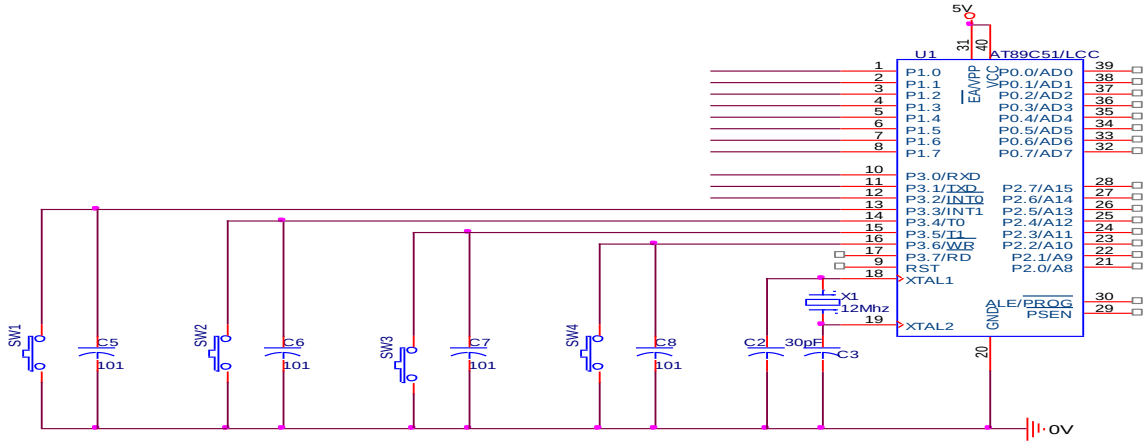
+Vậy ta chọn mỗi mức giá trị ứng với 5mV để dễ tính toán nhiệt độ khi lập trình ⇒ vậy mỗi độ ứng với 2 mức giá trị (5mv 1 giá trị, 1 độ ứng với 10mV).

⇒Mức điện áp ứng giá trị cao nhất của ADC là $\frac{V_{ref}}{256}$ =5mV⇒V_{ref}=256*5*10^{-3}=1.28 V

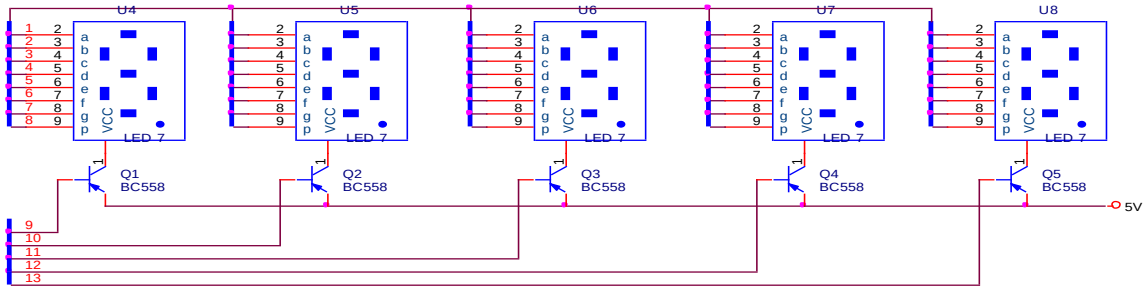
+Mà điện áp cấp đến ADC để so sánh là $V_{ref}/2$ nên điện áp cấp đến chân 9 của là $\frac{1.28}{2}$ =0.64 V ⇒ Khoảng nhiệt độ ta có thể đo được là $0^oC \rightarrow 128^oC$.

2.3.4 KHỐI XỬ LÝ VÀ KHỐI NÚT NHẤN

- Tụ 101 dùng để ngắn mạch các xung dao động nhiễu từ vi xử lý.



2.3.5 KHỐI HIỂN THỊ:



- Chọn V_{led}=2V , V_γ = 0,7V , V_{CE} = 0,2V (vì ta sử dụng transistor ở chế độ bão hòa), V_{CC} =4,75V.
- Điện trở hạn dòng vào vi điều khiển.

$$I_B=\frac{(V_{CC}-V_\gamma)}{R}$$

Chọn R=150 Ω =>I_B=27 mA

- Điện trở hạn dòng cho led :

Chọn $I_C \approx$ 10mA

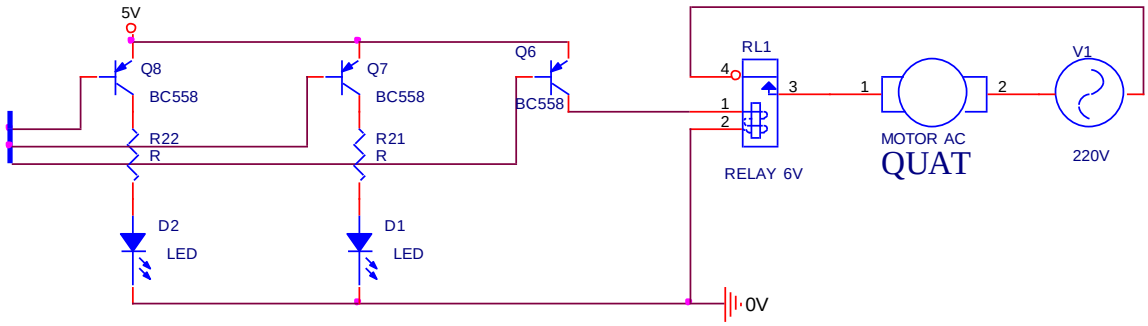
Ta có: $V_{CC}=V_{CE}+V_{LED}+V_R$ (với $V_R=I_C*R$)

⇔ 4,75V=0,2V +2V+0.01A*R

=> R=255 Ω

Chọn R=150 Ω (để led sáng hơn khi này I_C=17mA, dòng mà led chịu được lên tới 20mA).

2.3.6 KHỐI TẢI:



- Để điều khiển tải 220V ta sử dụng transistor để kích cho relay tắt/mở tải.
- Chọn V_{led}=2V , V_γ = 0,7V , V_{CE} = 0,2V (vì ta sử dụng transistor ở chế độ bão hòa), V_{CC} =4,75V.
- Điện trở hạn dòng vào vi điều khiển.

$$I_B=\frac{(V_{CC}-V_\gamma)}{R} \Leftrightarrow \text{Chọn R=150 Ω =>}I_B\text{=27 mA}$$

- Ở đây chỉ dùng 1 chân minh họa cách để kích relay điều khiển tải. Còn 2 chân còn lại chỉ điều khiển led.
- Điện trở hạn dòng cho led :

Chọn $I_C \approx 10mA$

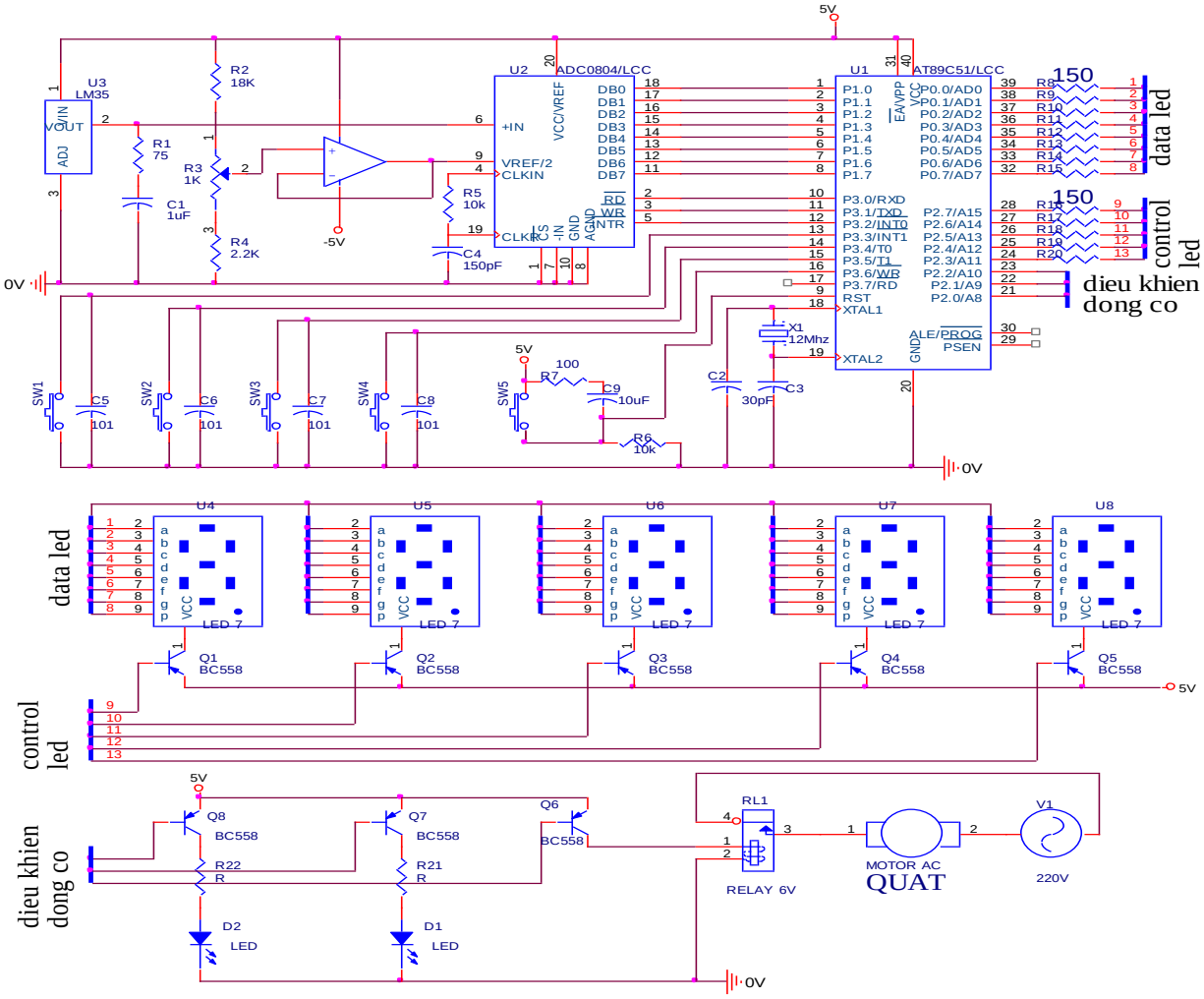
Ta có: $V_{CC}=V_{CE}+V_{LED}+V_R$ (với $V_R=I_C*R$)

$\Leftrightarrow$ 4,75V=0,2V +2V+0.01A*R

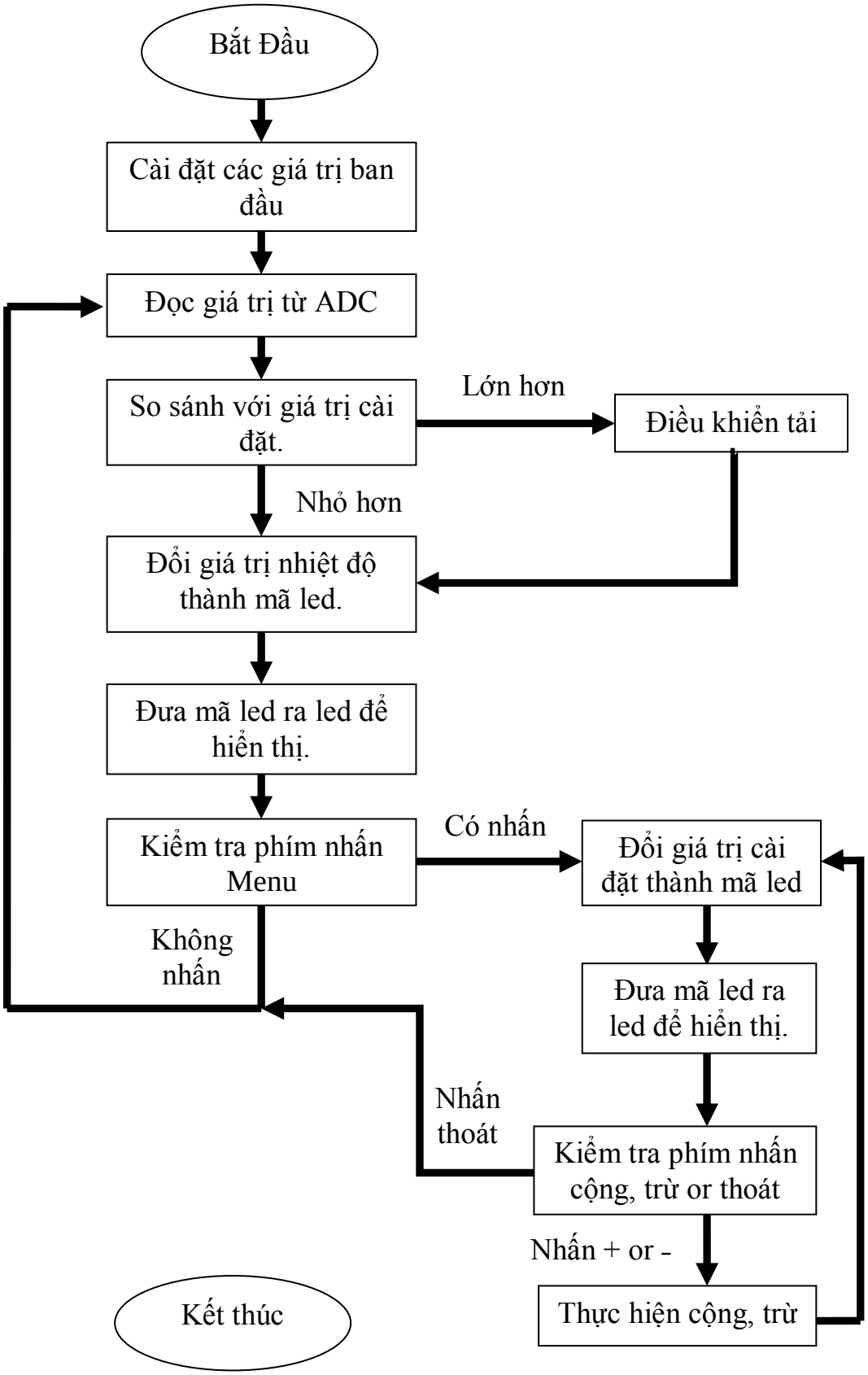
=> R=255 Ω

- Chọn R=150 Ω (để led sáng hơn khi này I_C=17mA).

2.4 TỔNG HỢP SƠ ĐỒ NGUYÊN LÝ CỦA MẠCH



2.5 SƠ ĐỒ THUẬT TOÁN CHO CHƯƠNG TRÌNH ĐIỀU KHIỂN



2.6 CHƯƠNG TRÌNH VI ĐIỀU KHIỂN

```
include reg_51.pdf
led     equ     p0 ;==>> data led
adc     equ     p1      ;==>> data adc
gtle    equ     127 ;==> =0 or =1 de xac dinh gia tri 0.0 or 0.5
chuc    equ     126 ;==> gia tri hang chuc cua nhiet do
donvi   equ     125 ;==> gia tri hang don vi cua nhiet do
temp    equ     124 ;==> chua gia tri cua thanh ghi a khi cat thanh ghi a vao ngan sep
value   equ     123 ;==>        gia tri cai dat
intr    bit     p3.2 ;==>> dieu khien adc
write   bit     p3.1  ;==>> dieu khien adc
read    bit     p3.0  ;==>> dieu khien adc
l1      bit     p2.7 ;==>> dieu khien led
l2      bit     p2.6 ;==>> dieu khien led
l3      bit     p2.5 ;==>> dieu khien led
l4      bit     p2.4 ;==>> dieu khien led
l5      bit     p2.3 ;==>> dieu khien led
  mov p2,#0ffh ;==> tat het cac led va dong co
  mov p3,#0ffh ;==> dat cac chan nut nhan va cac chan dieu khien adc len 1
  mov adc,#0ffh ;==> dat gia tri ban dau cho port data adc(luon la 0ffh) (neu la 00h thi
du lieu lay vao se luon la 00h=>sai)
  mov dptr,#ma ;==> dua gia tri vao bang
  mov value,#60 ;==> dat gia tri nhiet do chuan ban dau
main:
        call read_adc ;==> doc gia tri tu adc
        call sosanh   ;==> so sanh gia tri tu adc voi gia tri cai dat
        call doiso    ;==> doi gia tri tu adc ra chuc, don vi, le
        call hienthi  ;==> hien thi ra led
```

```
        jnb p3.3,x1   ;==> kiem tra nut nhan neu co nhan nut thi nhay den x1 => nhay den
cai dat
        call hienthi  ;==> lap lai hien thi dong thoi co kiem tra phim nhan
        jnb p3.3,x1   ;==> de lam cham qua trinh lay du lieu moi tu adc
        call hienthi  ;==> gia tri hien thi ra se de doc hon
        jnb p3.3,x1
        call hienthi
        jnb p3.3,x1
        call hienthi
        jnb p3.3,x1
        call hienthi
        jnb p3.3,x1
        call hienthi
        jnb p3.3,x1
        ljmp main     ;==> nhay tro lai main thuc hien xoay vong
x1:
        ljmp caidat   ;==> nhay den cai dat khi co nhan phim
read_adc:
        setb read     ;==> dat cac chan RD, WR, INTR len muc cao
        setb write
        setb intr
        call delay    ;==> cho adc xu ly => vi adc xu ly cham hon chip vi dieu khien
        call delay
        clr write     ;==> xoa chan WR cua adc xuong thap
        call delay    ;==> cho adc xu ly
        call delay
        setb write    ;==> sau do dat WR len muc cao de adc bat dau chuyen doi gia tri
x2:             ;==> cho adc chuyen doi gia tri khi nao chuyen doi xong thi chan INTR se
duoc adc keo xuong muc thap
```

```
        jb intr,x2    ;==> khi chan INTR duoc keo xuong muc thap thi se thoat khoi vong
lap va thuc hien lenh ke tiep
        clr read      ;==> xoa chan RD cua adc xuong thap de adc dua du lieu ra
        call delay    ;==> cho adc xu ly
        call delay
        mov a,adc     ;==> di chuyen du lieu ma adc dua den port 1 vao thanh ghi a
        ret
sosanh:
        cjne a,value,x3 ;==> so sanh thanh ghi a voi gia tri cai dat '=' thuc hien lenh ke
tiep, '#' nhay den x3
        setb p2.0 ;==> khi gia tri trong thanh ghi a = gia tri cai dat ==> tat dong co
        setb p2.1 ;==> tat dong co
        setb p2.2 ;==> tat dong co
        ljmp doiso ;==> sau khi so sanh xong thi nhay toi buoc tiep theo la doi so
x3:
        jnc open  ;==> khi so sanh neu a > value thi co carry = 0 ( nguoc lai = 1 ) ta dung
lenh kiem tra co carry neu 0 nhay den open => mo dong co
        setb p2.0 ;==> neu co carry =1 thi thuc hien lenh ke tiep la tat cac dong co
        setb p2.1
        setb p2.2
        ljmp doiso
open:                         ;==> mo dong co de lam giam nhiet do
        clr p2.0
        clr p2.1
        clr p2.2
        ret
doiso:        ;==> doi gia tri sang chuc, don vi, le de hien thi len led
        mov temp,a   ;==> di chuyen gia tri cua a vao temp
        push acc     ;==> thuc hien cat thanh ghi a vao ngan sep
```

```
        push b      ;==> cat thanh ghi b vao ngan sep
        mov a,temp  ;==> dem gia tri cua tra lai cho a
        mov b,#2
        div ab      ;==> chia 2 de lay gia tri nhiet do
        mov gtle,b  ;==> dem so du sau khi chia bo vao o nho gia tri le de them '.0' or '.5'
        mov b,#10
        div ab      ;==> chia 10 de lay ra gia tri hang chuc va gia tri hang don vi
        mov chuc,a  ;==> sau khi chia a chua gia tri hang chuc => dem gia tri cua a den o
nho chuc
        mov donvi,b ;==> sau khi chia b chua gia tri hang don vi => dem gia tri cua b den
o nho donvi
        pop b       ;==> lay noi dung chua trong ngan sep tra lai cho thanh ghi
        pop acc
        ret
hienthi:    ;==> hien thi gia tri nhiet do
        push 05
        mov r5,#50  ;==> lap lai qua trinh hien thi 50 lan
x4:
        mov a,chuc  ;==> dua gia tri hang chuc vao a
        movc a,@a+dptr  ;==> dua vao thanh ghi a ma led de hien thi gia tri hang chuc
        mov led,a       ;==> dua ma led chua trong thanh ghi a ra port 0
        clr l1          ;==> hien thi gia tri hang chuc tren led 1
        call delay      ;==> thuc hien cho de mat luu hinh anh hang chuc
        setb l1         ;==> tat led 1
        mov a,donvi     ;==> dua vao thanh ghi a gia tri hang don vi
        add a,#10       ;==> cong gia tri hang don vi them 10 dua vao thanh ghi a ma led
hien thi gia tri hang don vi co them dau '.'
        movc a,@a+dptr  ;==> dua vao thanh ghi a ma led de hien thi gia tri hang don vi
        mov led,a       ;==> dua ma led chua trong thanh ghi a ra port 0
```

```
        clr l2          ;==> hien thi gia tri hang don vi tren led 2
        call delay      ;==> thuc hien cho de mat luu hinh anh hang chuc
        setb l2         ;==> tat led 2
        mov a,gtle      ;==> tuong tu ta hien thi gia tri le la 0 or 5
        mov b,#5
        mul ab
        movc a,@a+dptr
        mov led,a
        clr l3
        call delay
        setb l3
        mov led,#9ch    ;==> hien thi ki tu do: 'o'
        clr l4
        call delay
        setb l4
        mov led,#0c6h   ;==> hien thi ky tu 'C'
        clr l5
        call delay
        setb l5
        jnb p3.3,caidat
        djnz r5,x4
        pop 05
        ret
caidat:
        mov a,value             ;==> dua gia tri cai dat vao thanh ghi a
        call doiso      ;==> thuc hien doi so de hien thi
        call hienthi    ;==> hien thi gia tri cai dat
        jnb p3.4,cong   ;==> kiem tra phim nhan neu bit p3.4=0 co nhan phim => cong de
thuc hien lenh cong nguoc lai thuc hien lenh ke tiep
```

```
        jnb p3.5,tru    ;==> kiem tra phim nhan neu bit p3.5=0 co nhan phim => tru de
thuc hien lenh tru nguoc lai thuc hien lenh ke tiep
        jnb p3.6,thoat  ;==> kiem tra phim nhan neu bit p3.6=0 co nhan phim => thoat de
thuc hien lenh thoat nguoc lai thuc hien lenh ke tiep
        ljmp caidat     ;==> nhay tro lai cai tiep tuc cai dat den khi phim thoat duoc nhan
cong:
        mov a,value     ;==> dua gia tri cai dat vao thanh ghi a
        add a,#1        ;==> vi co nhan phim nen cong them don vi tuong ung +0.5 oC
        mov value,a     ;==> dem gia tri da duoc cong bo vao value
        ljmp caidat     ;==> nhay tro lai cai dat de cho nhan phim tiep theo
tru:            ;==> qua trinh cung thuc hien tuong tu
        mov a,value
        subb a,#1       ;==> tru di 1don vi tuong ung -0.5 oC
        mov value,a
        ljmp caidat
thoat:
        ljmp main       ;==> nhay tro lai main de thoat va bat dau lai qua trinh tu dau
        ret
delay:          ;==> xoay vong dem nguoc 255 den 0 de cho
        push 06
        mov r6,#255
        djnz r6,$
        pop 06
        ret
ma:     db
        0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0f8H,80H,90H,40h,79h,24h,30h,19h,12h,
02h,78h,00h,10h
end main        ;==> bang tren co 10 gia tri dau <=> ma led tu 0 den 9 ; 10 gia tri sau
tuong ung ma led 0 den 9 nhung co them dau cham '.'
```

Chương 3

KẾT QỦA

******

3.1 KẾT QUẢ SAU KHI THI CÔNG MẠCH

- Mạch sau khi thi công chạy đúng như yêu cầu và ổn định nhưng sai số vẫn còn lớn ($\pm 1^oC$) đồng thời do không thực hiện biện pháp lấy giá trị trung bình của nhiều lần đo nên số hiện thị không ổn định.

3.2 HƯỚNG PHÁT TRIỂN MẠCH

- Ta cần thực hiện lấy giá trị trung bình của khoảng 5 đến 10 lần đo (hoặc hơn thì càng chính xác nhưng sẽ làm chậm tiến độ xử lý của toàn bộ công việc) thì giá trị dùng để so sánh và hiển thị sẽ chính xác hơn ít biến động hơn. Đồng thời có thể dùng ADC 9 bít để ngõ ra của ADC cho giá trị số chính xác hơn $\Rightarrow$ giảm sai số.
- Có thể chuyển sang hiển thị dùng LCD để làm cho mạch gọn hơn hiển thị đẹp hơn.

3.3 ỨNG DỤNG CỦA MẠCH

- Dùng để điều khiển động cơ (hoặc tải bất kì) theo nhiệt độ ví dụ :máy nước nóng , máy điều hòa nhiệt độ, tắt/mở động cơ khi nhiệt quá nóng hoặc quá lạnh,….vv.

TÀI LIỆU THAM KHẢO

- Cuốn " Kỹ Thuật Xung " Ths Nguyễn Trọng Hải trường Đại Học Kỹ Thuật Công Nghệ TPHCM.
- Cuốn " Kỹ Thuật Số " Ths Nguyễn Trọng Hải trường Đại Học Kỹ Thuật Công Nghệ TPHCM.
- Cuốn " Điện Tử Tương Tự " Ths Nguyễn Thị Ngọc Anh trường Đại Học Kỹ Thuật Công Nghệ TPHCM.
- Cuốn " Vi Điều Khiển " Ths Trần Viết Thắng trường Đại Học Kỹ Thuật Công Nghệ TPHCM.
- Trang Web Tham khảo:
  + Dientuvienthong.net
  + Tailieu.vn
  + Datasheetcatalog.com
  + hanhtrangsinhvien.net