BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

ISO 9001:2008

THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG ĐIỀU KHIỂN ĐỘNG CƠ THEO NHIỆT ĐỘ

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP

HẢI PHÒNG - 2017

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

ISO 9001:2008

THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG ĐIỀU KHIỂN ĐỘNG CƠ THEO NHIỆT ĐỘ

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

NGÀNH ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP

Sinh viên: Kiều Công Hòa
Người hướng dẫn: Ths. Nguyễn Đoàn Phong

HẢI PHÒNG - 2017

Cộng hoà xã hội chủ nghĩa Việt Nam

Độc lập – Tự Do – Hạnh Phúc

----------o0o----------

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên **:** Kiều Công Hòa – MSV : 1312102017

Lớp **:** ĐC1201- Ngành Điện Tự Động Công Nghiệp

Tên đề tài : Thiết kế và xây dựng hệ thống điều khiển động cơ theo nhiệt độ.

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp (về lý luận, thực tiễn, các số liệu cần tính toán và các bản vẽ).

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Các số liệu cần thiết để thiết kế, tính toán

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Địa điểm thực tập tốt nghiệp.....:

CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên	:	Nguyễn Đoàn Phong
Học hàm, học vị	:	Thạc sĩ
Cơ quan công tác	:	Trường Đại học dân lập Hải Phòng
Nội dung hướng dẫn	:	Toàn bộ đề tài

Người hướng dẫn thứ hai:

Họ và tên	:	
Học hàm, học vị	:	
Cơ quan công tác	:	
Nội dung hướng dẫn	:	

Đề tài tốt nghiệp được giao ngày tháng năm 2017.
Yêu cầu phải hoàn thành xong trước ngày.....tháng.....năm 2017.

Đã nhận nhiệm vụ Đ.T.T.N	Đã giao nhiệm vụ Đ.T.T.N
Sinh viên	Cán bộ hướng dẫn Đ.T.T.N
Kiều Công Hòa	Th.S Nguyễn Đoàn Phong

Hải Phòng, ngày.....tháng.....năm 2017

HIỆU TRƯỞNG

GS.TS.NGƯT TRẦN HỮU NGHỊ

PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1.Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp.

.....

.....

.....

.....

.....

2. Đánh giá chất lượng của Đ.T.T.N (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận thực tiễn, tính toán giá trị sử dụng, chất lượng các bản vẽ..)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Cho điểm của cán bộ hướng dẫn

(Điểm ghi bằng số và chữ)

Ngày……tháng…….năm 2017

Cán bộ hướng dẫn chính

(Ký và ghi rõ họ tên)

NHẬN XÉT ĐÁNH GIÁ CỦA NGƯỜI CHẤM PHẢN BIỆN
ĐỀ TÀI TỐT NGHIỆP

1. Đánh giá chất lượng đề tài tốt nghiệp về các mặt thu thập và phân tích số liệu ban đầu, cơ sở lý luận chọn phương án tối ưu, cách tính toán chất lượng thuyết minh và bản vẽ, giá trị lý luận và thực tiễn đề tài.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Cho điểm của cán bộ chấm phản biện
(Điểm ghi bằng số và chữ)

Ngày……tháng…….năm 2017
Người chấm phản biện
(Ký và ghi rõ họ tên)

MỤC LỤC

LỜI MỞ ĐẦU

Ngày nay với những ứng dụng của khoa học kĩ thuật tiên tiến, thế giới của chúng ta đã và đang ngày một thay đổi, văn minh và hiện đại hơn. Trong đó sự phát triển của kĩ thuật tự động hóa đã đóng góp vai trò quan trọng, tạo ra hàng loạt những thiết bị với các đặc điểm nổi bật như: sự chính xác, an toàn, tốc độ nhanh, gọn nhẹ ... Ý tưởng đề tài xuất phát từ bài toán thực tế là thiết kế hệ thống đo nhiệt độ phòng, từ đó dựa vào nhiệt độ đặt để điều khiển động cơ phù hợp với sự thay đổi nhiệt độ.

Đề tài "***Thiết kế và xây dựng hệ thống điều khiển động cơ theo nhiệt độ***", do Thạc sĩ Nguyễn Đoàn Phong hướng dẫn. Là sự kết hợp của nhiều mạch điện tử cơ bản cũng như sử dụng phần tử vi điều khiển trong chương trình giảng dạy, là sự tổng hợp kiến thức các môn cơ sở ngành và kĩ năng thực hành trong môn Vi điều khiển. Đề tài của em gồm 3 chương:

Chương 1. Tổng quan về các phần tử

Chương 2. Thiết kế hệ thống điều khiển động cơ DC theo nhiệt độ

Chương 3. Chương trình điều khiển

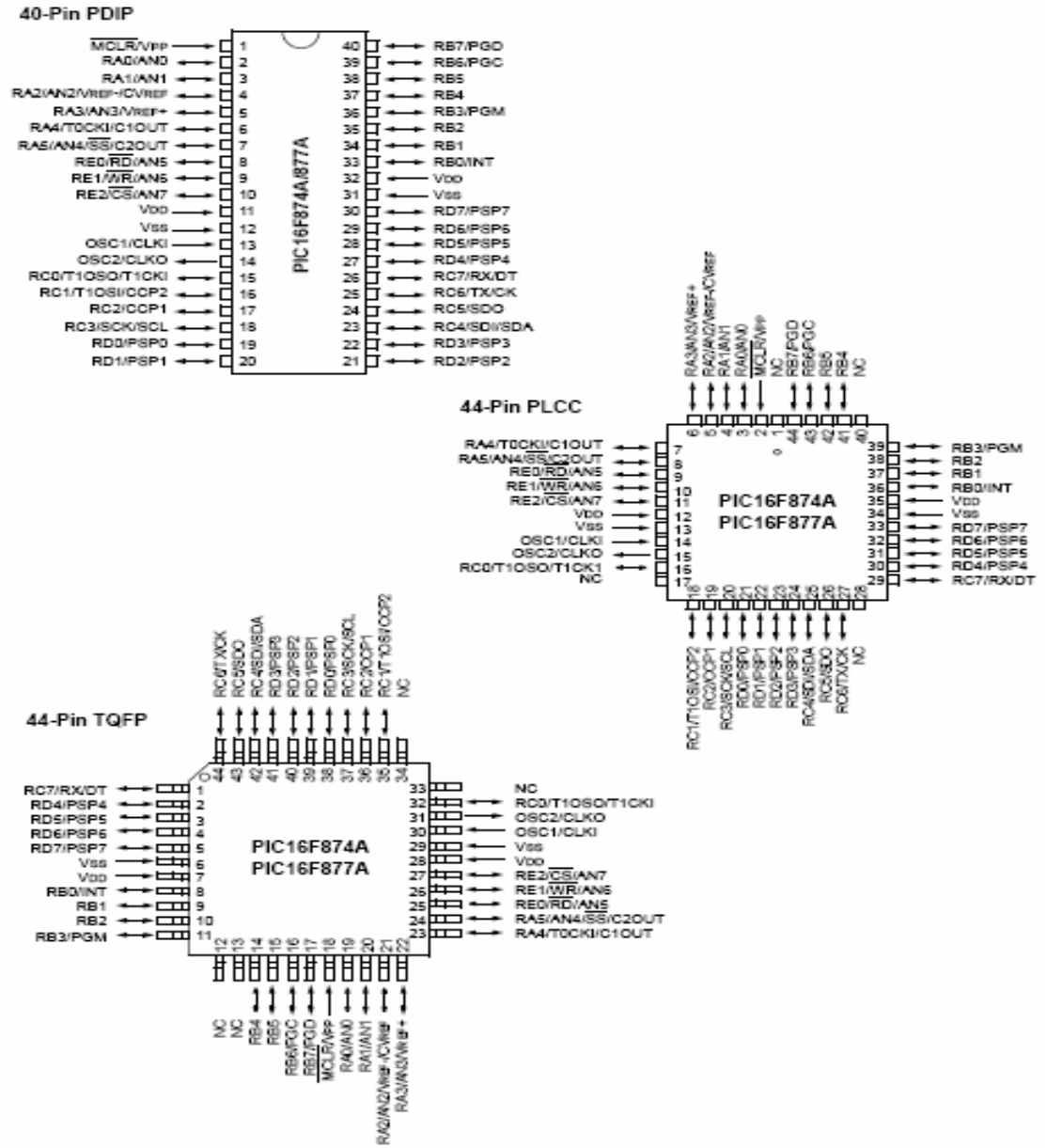
CHƯƠNG 1.

TỔNG QUAN VỀ CÁC PHẦN TỬ

1.1. TỔNG QUAN VỀ PIC16F877A

1.1.1. Chức năng và sơ đồ chân vi điều khiển PIC16F877A

PIC16F877A là dòng PIC phổ biến nhất hiện nay, đủ mạnh về tính năng, bộ nhớ đủ cho hầu hết các ứng dụng thông thường.



Hình 1.1: Vi điều khiển PIC 16F877A/PIC16F874A và các dạng sơ đồ chân.

Chức năng của PIC16F877A:

- Có khả năng xử lí ngắt từ nhiều nguồn ngắt khác nhau như ngắt ngoài, ngắt tràn Timer, ngắt ngoại vi như ngắt ADC...
- Chức năng CCP gồm Comporator (bộ so sánh), capture và PWM (điều chế độ rộng xung).
- Chức năng giao tiếp đồng bộ nối tiếp SSP gồm 2 giao tiếp SPI và I2C.
- Chức năng bộ truyền phát không đồng bộ đa năng nối tiếp USART ở dạng module phần cứng phục vụ cho giao tiếp theo chuẩn RS-232.
- Bộ ADC 10 bit chuyển đổi tín hiệu tượng tự sang tín hiệu số.
- Chức năng giao tiếp song song PSP.

1.1.2. Một vài thông số về vi điều khiển PIC16877A

Đây là vi điều khiển thuộc họ PIC16Fxxx với tập lệnh gồm 37 lệnh có độ dài 14 bit. Mỗi lệnh đều được thực thi trong một chu kì xung clock. Tốc độ hoạt động tối đa cho phép là 20MHz với một chu kì lệnh là 200ms. Bộ nhớ chương trình 8Kx14 bit, bộ nhớ dữ liệu 368x8 byte RAM và bộ nhớ dữ liệu EEPROM với dung lượng 256x8 byte. Số PORT I/O là 5 với 33 pin I/O.

Các đặc tính ngoại vi bao gồm các khối chức năng sau:

-Timer0: bộ đếm 8 bit với bộ chia tần số 8 bit.

-Timer1: bộ đếm 16 bit với bộ chia tần số, có thể thực hiện chức năng đếm dựa vào xung clock ngoại vi ngay khi vi điều khiển hoạt động ở chế độ sleep.

-Timer2: bộ đếm 8 bit với bộ chia tần số, bộ postcaler.

- Hai bộ Capture/so sánh/điều chế độ rộng xung.

- Các chuẩn giao tiếp nối tiếp SSP(Synchronous Serial Port) với các chân điều khiển RD, WR, CS ở bên ngoài.

Các đặc tính Analog:

- 8 kênh chuyển đổi ADC 10 bit.

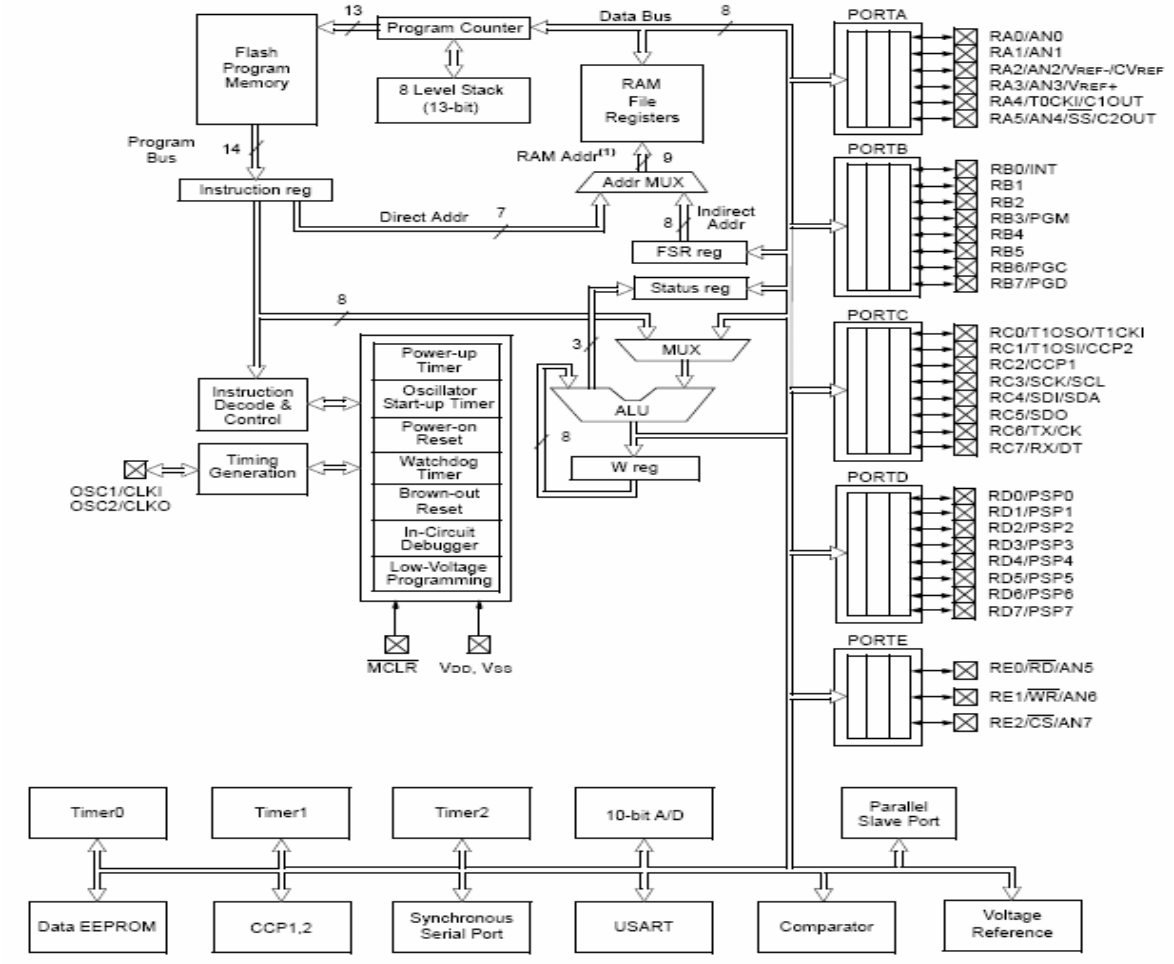
- Hai bộ so sánh.

Bên cạnh đó là một vài đặc tính khác của vi điều khiển như:

- Bộ nhớ flash với khả năng ghi xóa được 100.000 lần.
- Bộ nhớ EEPROM với khả năng ghi xóa được 1.000.000 lần.
- Dữ liệu bộ nhớ EEPROM có thể lưu trữ trên 40 năm.
- Khả năng tự nạp chương trình với sự điều khiển của phần mềm.
- Nạp được chương trình ngay trên mạch điện ICSP (In Circuit Serial Programming) thông qua 2 chân.
- Watchdog Timer với bộ dao động trong.
- Chức năng bảo mật mã chương trình.
- Có thể hoạt động với nhiều dạng Oscillator.

1.1.3. Sơ đồ khối vi điều khiển PIC16F877A

Sơ đồ khối của PIC16F877A:



Hình 1.2: Sơ đồ khối vi điều khiển PIC16F877A.

1.1.4. Tổ chức bộ nhớ

Cấu trúc bộ nhớ vi điều khiển PIC16F877A bao gồm bộ nhớ chương trình (Program memory) và bộ nhớ dữ liệu (Data Memmory).

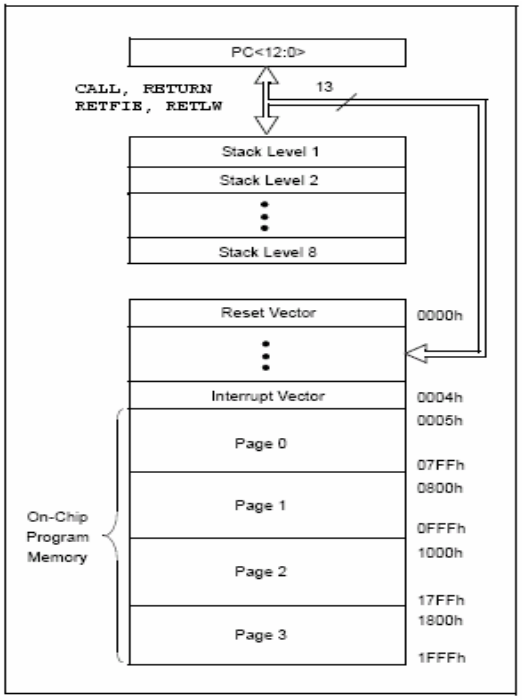
1.1.4.1. Bộ nhớ chương trình

Bộ nhớ chương trình của vi điều khiển PIC16F877A là bộ nhớ flash, dung lượng bộ nhớ 8K word (1 word = 14 bit) và được phân thanh nhiều trang (từ page 0 đến page 3). Như vậy bộ nhớ chương trình có khả năng chứa được 8x1024 = 8192 lệnh (vì một lệnh sau khi mã hóa sẽ có dung lượng 1 word = 14 bit).

Để mã hóa được địa chỉ của 8K word bộ nhớ chương trình có dung lượng 3 bit (PC<12:0>).

Khi vi điều khiển được reset, bộ đếm chương trình sẽ chỉ đến địa chỉ 0000h (Reset vector). Khi có ngắt xảy ra, bộ đếm chương trình sẽ chỉ đến địa chỉ 0004h (Interrupt vector).

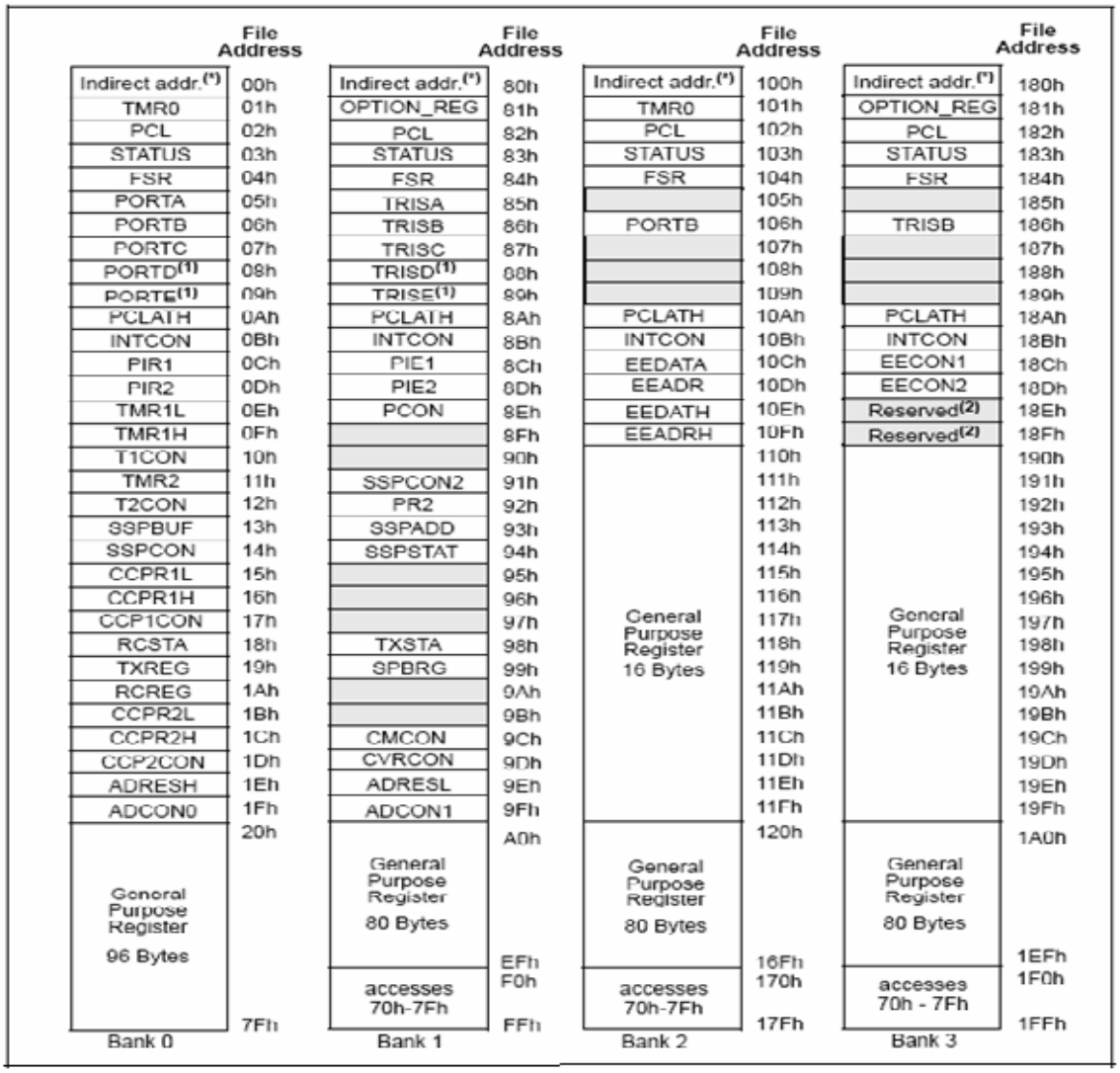
Bộ nhớ chương trình không bao gồm bộ nhớ stack và không được địa chỉ hóa bởi bộ đếm chương trình.



Hình 1.3: Bộ nhớ chương trình PIC16F877A.

1.1.4.2. Bộ nhớ dữ liệu

Bộ nhớ dữ liệu của PIC là bộ nhớ EEPROM được chia ra làm nhiều bank. Đối với PIC16F877A bộ nhớ dữ liệu được chia ra làm 4 bank. Mỗi bank có dung lượng 28 byte, bao gồm các thanh ghi có chức năng đặc biệt SFR (Special Function Register) nằm ở các vùng địa chỉ thấp và các thanh ghi mục đích chung GPR (General Purpose Register) nằm ở vùng địa chỉ còn lại trong bank. Các thanh ghi SFR thường xuyên được sử dụng (ví dụ như thanh ghi STATUS) sẽ được đặt ở tất cả các bank của bộ nhớ dữ liệu giúp thuận tiện trong quá trình truy xuất và làm giảm bớt lệnh của chương trình. Sơ đồ cụ thể của bộ nhớ dữ liệu PIC16F877A như sau:



Hình 1.4: Sơ đồ bộ nhớ dữ liệu PIC16F877A.

1.1.4.2.1. Thanh ghi chức năng đặc biệt SFR

Đây là các thanh ghi được sử dụng bởi CPU hoặc được dùng để thiết lập và điều khiển các khối chức năng được tích hợp bên trong vi điều khiển. Có thể phân thanh ghi SFR thành 2 loại: thanh ghi SFR liên quan đến các chức năng bên trong (CPU) và thanh ghi SFR dùng để thiết lập và điều khiển các khối chức năng bên ngoài (ADC, PWM,...).

- Thanh ghi STATUS (03h, 83h, 103h, 183h): thanh ghi chứa kết quả thực hiện phép toán của khối ALU, trạng thái reset và các bit chọn bank cần truy xuất trong bộ nhớ dữ liệu.

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

- Thanh ghi OPTION_REG (81h, 181h): thanh ghi này cho phép đọc và ghi, cho phép điều khiển chức năng pull-up của các chân trong PORTB, xác lập các tham số về xung tác động, cạnh tác động của ngắt ngoại vi và bộ đếm Timer0.

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

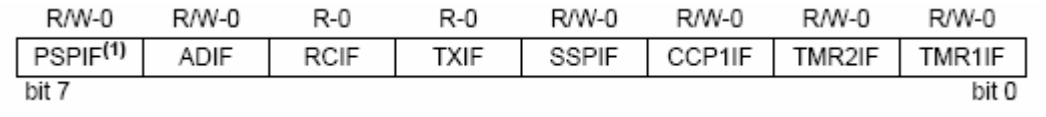
- Thanh ghi INTCON (0Bh, 8Bh, 10Bh,18Bh): thanh ghi cho phép đọc và ghi, chứa các bit điều khiển và các bit cờ hiệu khi Timer0 bị tràn, ngắt ngoại vi RB0/INT và ngắt interrupt-on-change tại các chân của PORTB.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

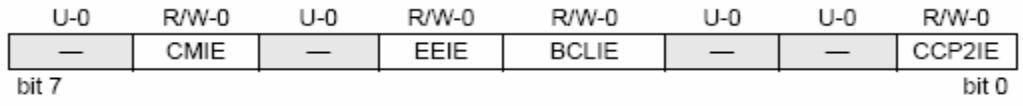
- Thanh ghi PIE1 (8Ch): chứa các bit điều khiển chi tiết các ngắt của các khối chức năng ngoại vi.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

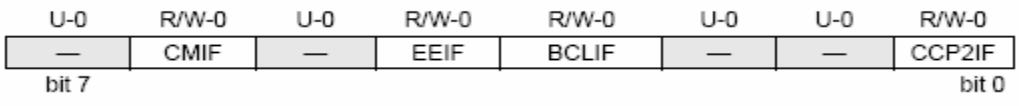
- Thanh ghi PIR1 (0Ch): chứa cờ ngắt của các khối chức năng ngoại vi, các ngắt này được cho phép bởi các bit điều khiển chứa trong thanh ghi PIE1.



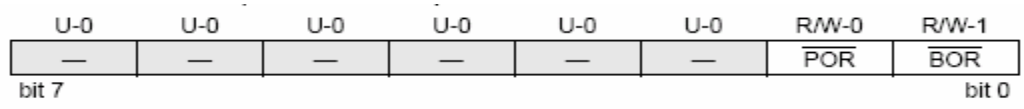
- Thanh ghi PIE2 (8Dh): chứa các bit điều khiển ngắt của các khối chức năng CCP2, SSP bus, ngắt của bộ so sánh và ngắt ghi vào bộ nhớ EEPROM.



- Thanh ghi PIR2 (0Dh): chứa các cờ ngắt của các khối chức năng ngoại vi, các ngắt này được cho phép bởi các bit điều khiển chứa trong thanh ghi PIE2.



- Thanh ghi PCON (8Eh): chứa các cờ hiệu cho biết trạng thái các chế độ reset của vi điều khiển.



1.1.4.2.2.Thanh ghi mục đích chung GPR

Các thanh ghi này có thể được truy xuất trực tiếp hoặc gián tiếp thông qua thanh ghi FSR (File Select Register). Đây là các thanh ghi dữ liệu thông thường, người sử dụng có thể tùy theo mục đích chương trình mà có thể dùng các thanh ghi này để chứa các biến số, hằng số, kết quả hoặc các tham số phục vụ cho chương trình.

1.1.4.3. Stack

Stack không nằm trong bộ nhớ chương trình hay bộ dữ liệu mà là một vùng nhớ đặc biệt không cho phép đọc hay ghi. Khi lệnh CALL được thực hiện hay khi một ngắt xảy ra làm chương trình bị rẽ nhánh, giá trị của bộ đếm chương trình PC tự động được vi điều khiển cất vào trong stack. Khi một trong các lệnh RETURN, RETLW hay RETFIE được thực thi, giá trị PC sẽ tự

động được lấy ra từ trong stack, vi điều khiển sẽ thực hiện tiếp chương trình theo đúng quy trình định trước.

Bộ nhớ stack trong vi điều khiển PIC họ 16Fxxxx có khả năng chứa được 8 địa chỉ và hoạt động theo cơ chế xoay vòng. Nghĩa là giá trị cất vào bộ nhớ stack lần thứ 9 sẽ ghi đè lên giá trị cất vào stack lần đầu tiên và giá trị cất vào stack lần thứ 10 sẽ ghi đè lên giá trị cất vào stack lần thứ 2.

Cần chú ý là không có cờ hiệu nào cho biết trạng thái stack, do đó ta không biết được khi nào stack tràn. Bên cạnh đó tập lệnh của vi điều khiển dòng PIC cũng không có lệnh POP hay PUSH, các thao tác với bộ nhớ stack sẽ hoàn toàn được điều khiển bởi CPU.

1.1.5. Các cổng xuất nhập của PIC16F877A

Cổng xuất nhập (I/O port) chính là phương tiện mà vi điều khiển dùng để tương tác với thế giới bên ngoài. Sự tương tác này rất đa dạng và thông qua quá trình tương tác đó, chức năng của vi điều khiển được thể hiện một cách rõ ràng.

Vi điều khiển PIC16F877A có 5 cổng xuất nhập, bao gồm PORTA, PORTB, PORTC, PORTD, PORTE.

1.1.5.1. Cổng PORTA

PORTA (RPA) gồm 6 I/O pin. Đây là các chân “hai chiều” (bidirectional pin), nghĩa là có thể xuất và nhập được. Chức năng I/O này được điều khiển bởi thanh ghi TRISA (địa chỉ 85h). Muốn xác lập chức năng của một chân trong PORTA là input, ta “set” bit điều khiển tương ứng với chân đó trong thanh ghi TRISA và ngược lại, muốn xác lập chức năng của một chân trong PORTA là output, ta “clear” bit điều khiển tương ứng với chân đó trong thanh ghi TRISA. Thao tác này hoàn toàn tương đối với các PORT và các thanh ghi điều khiển tương ứng TRIS (đối với PORTA là TRISA, PORTB là TRISB, PORTC là TRISC,....). Bên cạnh đó PORTA còn là ngõ ra của bộ ADC, bộ so

sánh, ngõ vào analog ngõ vào xung clock của Timer0 và ngõ vào của bộ giao tiếp MSSP (Master Synchronous Serial Port).

Các thanh ghi SFR liên quan đến PORTA gồm:

- PORTA (địa chỉ 05h): chứa giá trị các pin trong PORTA.
- TRISA (địa chỉ 85h): điều khiển xuất nhập.
- CMCON (địa chỉ 9Ch): thanh ghi điều khiển bộ so sánh.
- CVRCON (địa chỉ 9Dh): thanh ghi điều khiển bộ so sánh điện áp.
- ADCON1 (địa chỉ 9Fh): thanh ghi điều khiển bộ ADC.

1.1.5.2. Cổng PORTB

PORTB (RPB) gồm 8 pin I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISB. Bên cạnh đó một số chân của PORTB còn được sử dụng trong quá trình nạp chương trình cho vi điều khiển với các chế độ nạp khác nhau. PORTB còn liên quan đến ngắt ngoại vi và bộ Timer0. PORTB còn được tích hợp chức năng điện trở kéo lên được điều khiển bởi chương trình.

Các thanh ghi SFR liên quan đến PORTB gồm:

- PORTB (địa chỉ 06h, 106h): chứa giá trị các pin trong PORTB.
- TRISB (địa chỉ 86h, 186h): điều khiển xuất nhập.
- OPTION_REG (địa chỉ 81h, 181h): điều khiển ngắt ngoại vi và bộ Timer0.

1.1.5.3. Cổng PORTC

PORTC (RPC) gồm 8 pin I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISC. Bên cạnh đó PORTC còn chứa các chân chức năng của bộ so sánh, bộ Timer1, bộ PWM và các chuẩn giao tiếp nối tiếp I2C, SPI, SSP, USART.

Các thanh ghi điều khiển liên quan đến PORTC:

- PORTC (địa chỉ 07h): chứa giá trị các pin trong PORTC.
- TRISC (địa chỉ 87h): điều khiển xuất nhập.

1.1.5.4. Cổng PORTD

PORTD (RPD) gồm 8 chân I/O, thanh ghi điều khiển xuất nhập tương ứng là TRISD. PORTD còn là cổng xuất dữ liệu của chuẩn giao tiếp PSP (Parallel Slave Port).

Các thanh ghi liên quan đến PORTD gồm:

- Thanh ghi PORTD: chứa giá trị các pin trong PORTD.
- Thanh ghi TRISD : điều khiển xuất nhập.
- Thanh ghi TRISE: điều khiển xuất nhập PORTE và chuẩn giao tiếp PSP.

1.1.5.5. PORTE

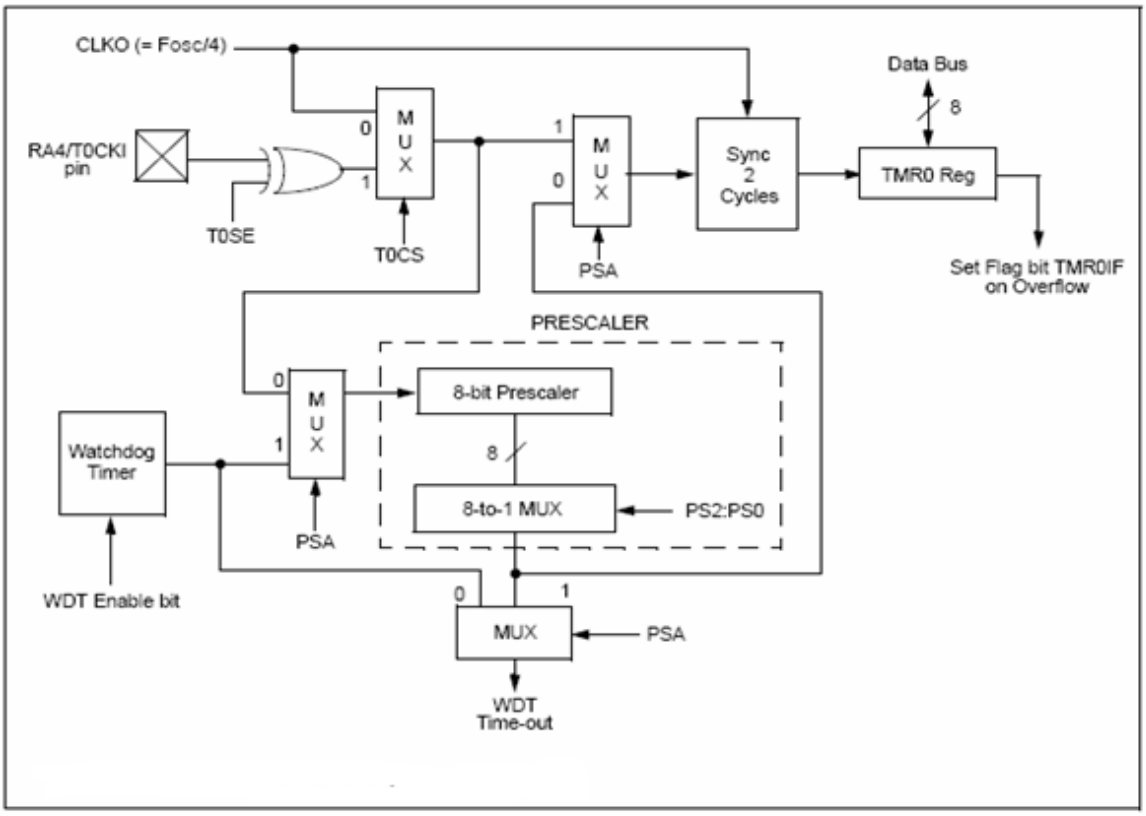
PORTE (RPE) gồm 3 chân I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISE. Các chân của PORTE có ngõ vào analog. Bên cạnh đó PORTE còn có các chân điều khiển của chuẩn giao tiếp PSP.

Các thanh ghi liên quan đến PORTE gồm:

- PORTE: chứa giá trị các chân trong PORTE.
- TRISE: điều khiển xuất nhập và xác lập các thông số cho chuẩn giao tiếp PSP.
- SDCON1: thanh ghi điều khiển khối ADC.

1.1.6. Timer0

Đây là một trong ba bộ đếm hoặc bộ định thời của vi điều khiển PIC16F877A. Timer0 là bộ đếm 8 bit được kết nối với bộ chia tần số (prescaler) 8 bit. Cấu trúc của Timer0 cho phép ta lựa chọn xung clock tác động và cạnh tích cực của xung clock. Ngắt Timer0 sẽ xuất hiện khi Timer0 bị tràn. Bit TMR0IE (INTCIN<5>) là bit điều khiển của Timer0. TMR0IE=1 cho phép ngắt Timer0 tác động, TMR0IF=0 không cho phép ngắt Timer0 tác động. Sơ đồ khối của Timer0 như sau:



Hình 1.5: Sơ đồ khối của Timer0.

Muốn Timer0 hoạt động ở chế độ Timer ta clear bit TOSC (OPTION_REG<5>), khi đó giá trị thanh ghi TMR0 sẽ tăng theo từng chu kì xung đồng hồ (tần số vào Timer0 bằng ¼ tần số oscillator). Khi giá trị thanh ghi TMR0 từ FFh trở về 00h, ngắt Timer0 sẽ xuất hiện. Thanh ghi TMR0 cho phép ghi và xóa được giúp ta ấn định thời điểm ngắt Timer0 xuất hiện một cách linh động.

Muốn Timer0 hoạt động ở chế độ counter ta set bit TOSC (OPTION_REG<5>). Khi đó xung tác động lên bộ đếm được lấy từ chân RA4/TOCK1. Bit TOSE (OPTION_REG<4>) cho phép lựa chọn cạnh tác động vào bộ đếm. Cạnh tác động sẽ là cạnh lên nếu TOSE=0 và cạnh tác động sẽ là cạnh xuống nếu TOSE=1.

Khi thanh ghi TMR0 bị tràn, bit TMR0IF (INTCON<2>) sẽ được set. Đây chính là cờ ngắt của Timer0. Cờ ngắt này phải được xóa bằng chương

trình trước khi bộ đếm bắt đầu thực hiện lại quá trình đếm. Ngắt Timer0 không thể "đánh thức" vi điều khiển từ chế độ sleep.

Bộ chia tần số (prescaler) được chia sẻ giữa Timer0 và WDT (Watchdog Timer). Điều đó có nghĩa là nếu prescaler được sử dụng cho Timer0 thì WDT sẽ không có được hỗ trợ của prescaler và ngược lại. Prescaler được điều khiển bởi thanh ghi OPTION_REG. Bit PSA (OPTION_REG<3>) xác định đối tượng tác động của prescaler. Các bit PS2:PS0 (OPTION_REG<2:0>) xác định tỉ số chia tần số của prescaler. Xem lại thanh ghi OPTION_REG để xác định lại một cách chi tiết về các bit điều khiển trên.

Các lệnh tác động lên giá trị thanh ghi TMR0 sẽ xóa chế độ hoạt động của prescaler. Khi đối tượng tác động là Timer0, tác động lên giá trị thanh ghi TMR0 sẽ xóa prescaler nhưng làm thay đổi đối tượng tác động của prescaler. Khi đối tượng là WDT, lệnh CLRWDT sẽ xóa prescaler, đồng thời prescaler sẽ ngưng tác vụ hỗ trợ cho WDT.

Các thanh ghi điều khiển liên quan đến Timer0 bao gồm:

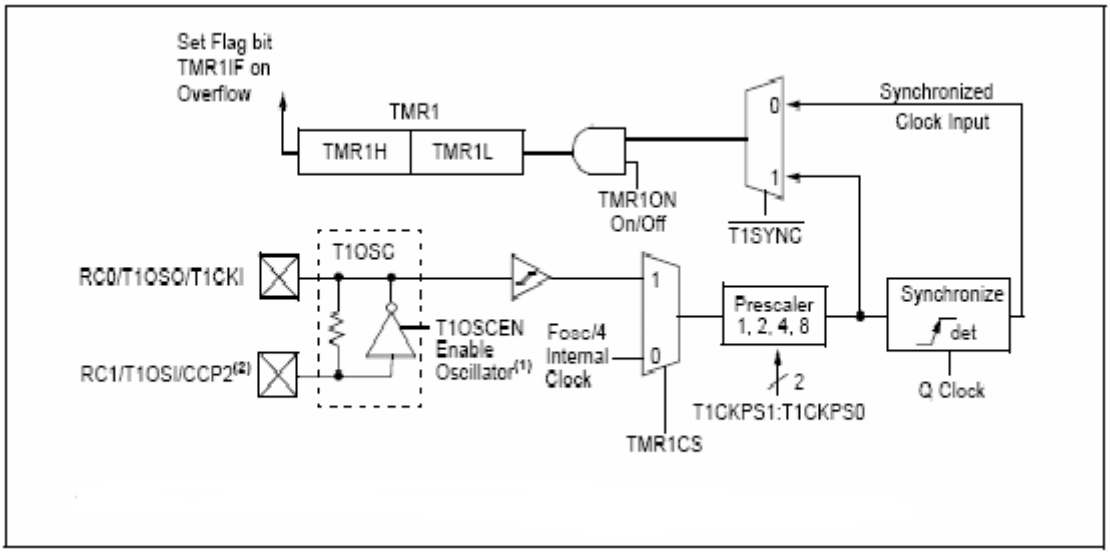
- TMR0 (địa chỉ 01h, 101h): chứa giá trị đếm của Timer0.
- INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): cho phép ngắt hoạt động (GIE và PEIE).
- OPTION_REG (địa chỉ 81h, 181h): điều khiển prescaler.

1.1.7. Timer1

Timer1 là bộ định thời 16 bit, giá trị của Timer1 sẽ được lưu trong hai thanh ghi (TMR1H:TMR1L). Cờ ngắt của Timer1 là bit TMR1IF(PIR1<0>). Bit điều khiển của Timer1 sẽ là TMR1IE (PIE<0>).

Tương tự như Timer0, Timer1 cũng có hai chế độ hoạt động: chế độ định thời (timer) với xung kích là xung clock của oscillator (tần số của timer bằng ¼ tần số của oscillator) và chế độ đếm (counter) với xung kích là xung phản ánh các sự kiện cần đếm lấy từ bên ngoài thông qua chân RC0/T1OSO/T1CKI (cạnh tác động là cạnh lên). Việc lựa chọn xung tác

động (tương ứng với việc lựa chọn chế độ hoạt động là timer hay counter) được điều khiển bởi bit TMR1CS (TICON<1>). Sau đây là sơ đồ khối của Timer1:



Hình 1.6: Sơ đồ khối của Timer1.

Ngoài ra Timer1 còn có chức năng reset input bên trong được điều khiển bởi một trong hai khối CCP(Capture/Compare/PWM).

Khi bit T1OSCEN (T1CON<3>) được set, Timer1 sẽ lấy xung clock từ hai chân RC1/T1OSI/CCP2 và RC0/T1OSO/T1CKI làm xung đếm. Timer sẽ bắt đầu đếm sau cạnh xuống đầu tiên của xung ngõ vào. Khi đó PORTC sẽ bỏ qua sự tác động của hai bit TRISC<1:0> và PORTC<2:1> được gán giá trị 0. Khi clear bit T1OSCEN Timer1 sẽ lấy xung đếm từ oscillator hoặc từ chân RC0/T1OSO/T1CKI.

Timer1 có hai chế độ đếm là đồng bộ (Synchronous) và bất đồng bộ (Asynchronous). Chế độ đếm được quyết định bởi bit điều khiển $\overline{\text{T1SYNC}}$ (T1CON<2>).

Khi $\overline{\text{T1SYNC}}$=1 xung đếm lấy từ bên ngoài sẽ không được đồng bộ hóa với xung clock bên trong, Timer1 sẽ tiếp tục quá trình đếm khi vi điều khiển đang ở chế độ sleep và ngắt do Timer1 tạo ra khi bị tràn có khả năng "đánh thức" vi điều khiển. Ở chế độ đếm đồng bộ, Timer1 không thể được sử dụng

để làm nguồn xung clock cho khối CCP (Capture/Compare/Pulse width modulation).

Khi $\overline{\text{T1SYNC}}$=0 xung đếm vào Timer1 sẽ được đồng bộ hóa với xung clock bên trong. Ở chế độ này Timer1 sẽ không hoạt động khi vi điều khiển đang ở chế độ sleep.

Các thanh ghi liên quan đến Timer1 gồm:

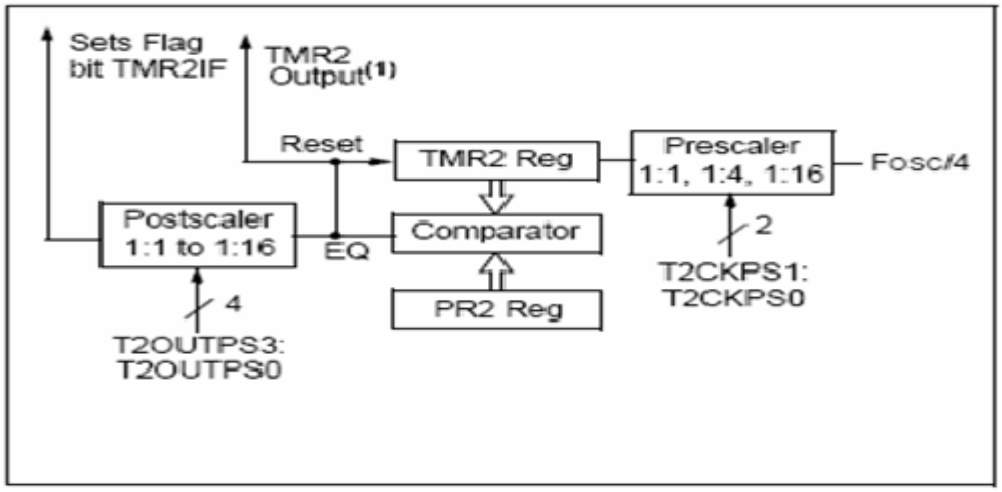
- INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): cho phép ngắt hoạt động (GIE và PEIE).
- PIR1 (địa chỉ 0Ch): chứa cờ ngắt Timer1 (TMR1IF).
- PIE1 (địa chỉ 8Ch): cho phép ngắt Timer1 (TMR1IE).
- TMR1L (địa chỉ 0Eh): chứa giá trị 8 bit thấp của bộ đếm Timer1.
- TMR1H (địa chỉ 0Eh): chứa giá trị 8 bit cao của bộ đếm Timer1.
- T1CON (địa chỉ 10h): xác lập các thông số cho Timer1.

1.1.8. Timer2

Timer2 là bộ định thời 8 bit và được hỗ trợ bởi hai bộ chia tần số prescaler và postcaler. Thanh ghi chứa giá trị đếm của Timer2 là TMR2. Bit cho phép ngắt Timer2 tác động là TMR2ON (T2CON<2>). Cờ ngắt của Timer2 là bit TMR2IF (PIR1<1>). Xung ngõ vào (tần số bằng ¼ tần số oscillator) được đưa qua bộ chia tần số prescaler 4 bit (với các tỉ số chia tần số là 1:1, 1:4 hoặc 1:6 và được điều khiển bởi các bit T2CKPS1:T2CKPS0 (T2CON<1:0>)).

Timer2 còn được hỗ trợ thanh ghi PR2. Giá trị đếm trong thanh ghi TMR2 sẽ tăng từ 00h đến giá trị chứa trong thanh ghi PR2, sau đó được reset về 00h. Khi reset thanh ghi PR2 được nhận giá trị mặc định FFh.

Ngõ ra của Timer2 được đưa qua bộ chia tần số postcaler với các mức chia từ 1:1 đến 1:16. Postcaler được điều khiển bởi 4 bit T2OUTPS3: T2OUTPS0. Ngõ ra của postcaler đóng vai trò quyết định trong việc điều khiển cờ ngắt.



Hình 1.7:Sơ đồ khối Timer2.

Ngoài ra ngõ ra của Timer2 còn được kết nối với khối SSP, do đó Timer2 còn đóng vai trò tạo ra xung clock đồng bộ cho khối SSP.

Các thanh ghi liên quan đến Timer2 gồm:

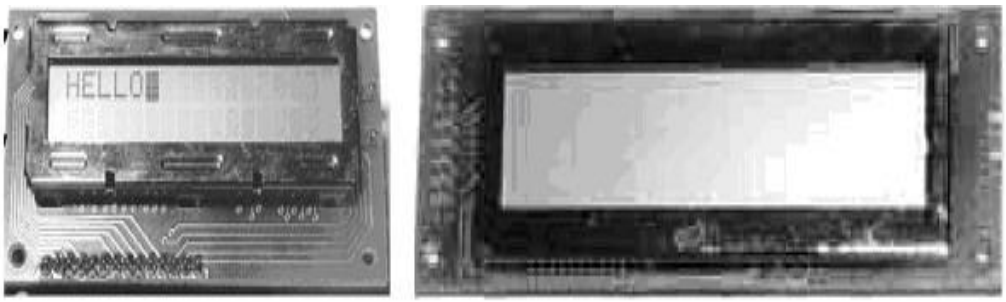
- INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): cho phép toàn bộ các ngắt (GIE và PEIE).
- PIR1 (địa chỉ 0Ch): chứa cờ ngắt Timer2 (TMR2IF).
- PIE1 (địa chỉ 8Ch): chứa bit điều khiển Timer2 (TMR2IE).
- TMR2 (địa chỉ 11h): chứa giá trị đếm của Timer2.
- T2CON (địa chỉ 12h): xác lập các thông số cho Timer2.
- PR2 (địa chỉ 92h): thanh ghi hỗ trợ cho Timer2.

1.2. THIẾT BỊ LCD

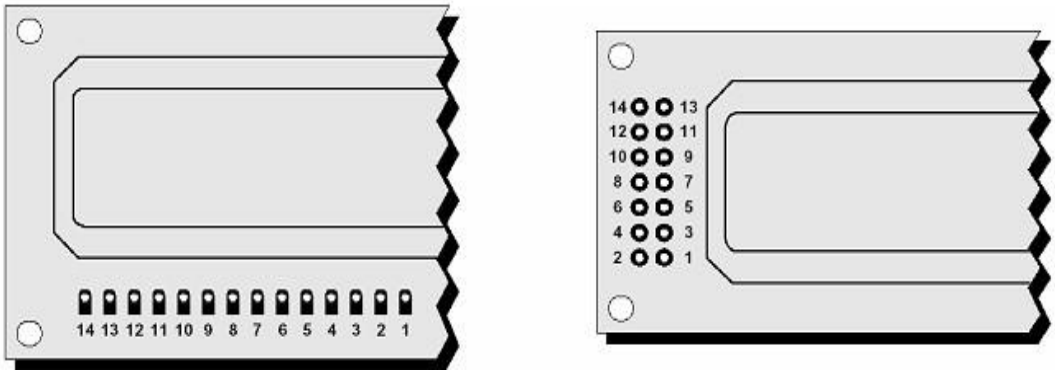
Ngày nay, thiết bị hiển thị LCD (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng của VĐK. LCD có rất nhiều ưu điểm so với các dạng hiển thị khác như nó có khả năng hiển thị kí tự đa dạng, trực quan (chữ, số và kí tự đồ họa), dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn rất ít tài nguyên hệ thống và giá thành rẻ ... Trong đề tài này em sử dụng HD44780 của Hitachi, một loại thiết bị hiển thị LCD rất thông dụng ở nước ta.

1.2.1. Hình dáng kích thước

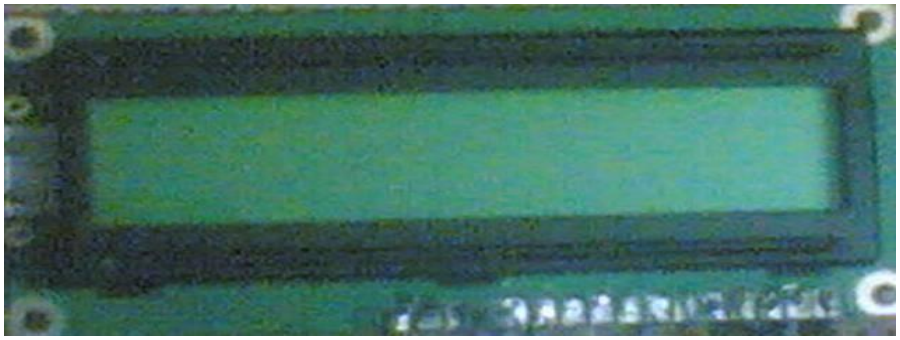
Có rất nhiều loại LCD với nhiều hình dáng và kích thước khác nhau, trên hình 1.8 là hai loại LCD thông dụng.



Hình 1.8: Hình hai loại LCD thông dụng.



Hình 1.9: Sơ đồ chân của LCD.



Hình 1.10: LCD loại DM 1602A.

Khi sản xuất LCD, nhà sản xuất đã tích hợp chíp điều khiển (HD44780) bên trong lớp vỏ và chỉ đưa các chân giao tiếp cần thiết. Các chân này được đánh số thứ tự và đặt tên như hình 1.9.

1.2.2.Các chân chức năng

Bảng 1.1: Các chân chức năng của HD44780.

Chân số	Tên	Chức năng
1	Vss	Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển.
2	Vdd	Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với 5V của mạch điều khiển.
3	Vo	Chân này dùng để điều chỉnh độ tương phản của LCD.
4	RS	Chân chọn thanh ghi (Register select). Nối chân RS với logic "0" (GND) hoặc logic "1" (Vcc) để chọn thanh ghi. + Logic "0": Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ "ghi" - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ "đọc" -read). + Logic "1": Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD.
5	RW	Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic "0" để LCD hoạt động ở chế độ ghi, hoặc nối với logic "1" để LCD ở chế độ đọc.
6	E	Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E. + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào (chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (low-to-high transition) của tín hiệu chân E. +Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện sườn lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp.
7÷14	DB0÷DB7	8 đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này: + Chế độ 8 bit: Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7. + Chế độ 4 bit: Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7.
15	A	15 là Catot, điện áp khoảng Uak=4,2V
16	K	Chân nối đất của đèn Back light.

1.2.3. Sơ đồ khối của HD44780

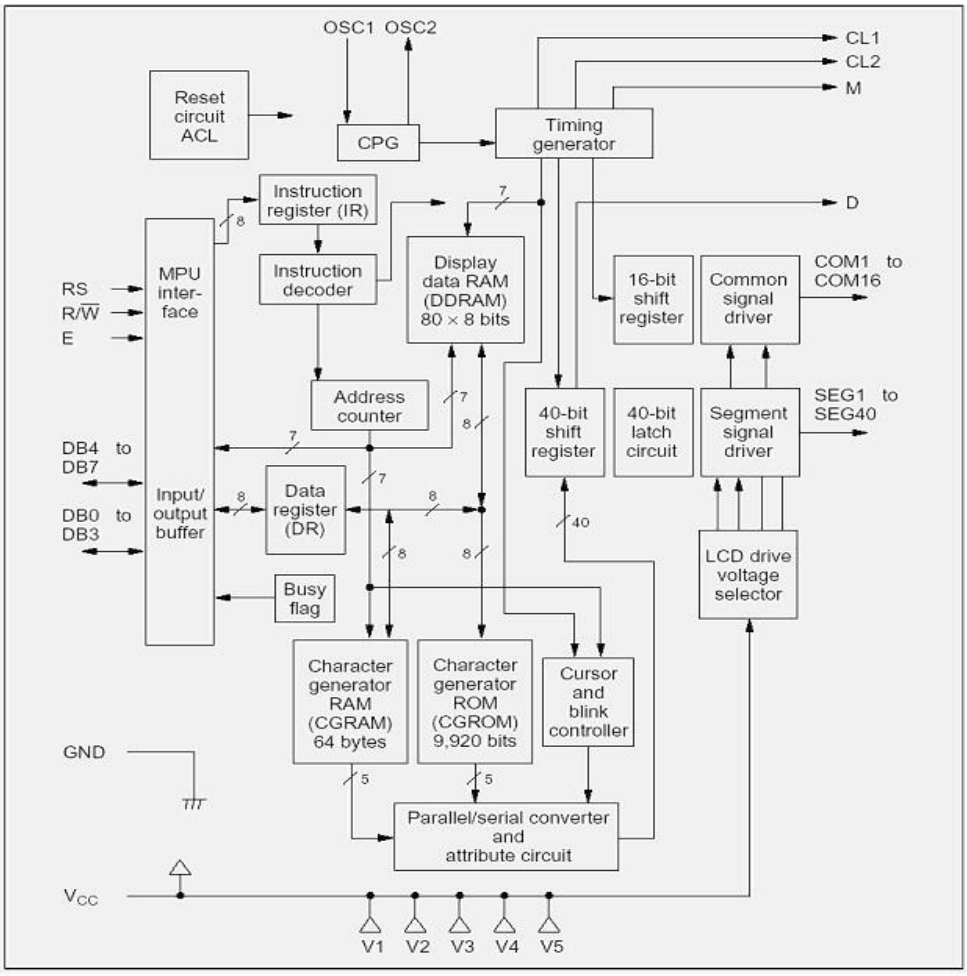
Để hiểu rõ hơn chức năng các chân và hoạt động của chúng, ta tìm hiểu sơ qua chíp HD44780 thông qua các khối cơ bản của nó.

a) Các thanh ghi:

Chíp HD44780 có 2 thanh ghi 8 bit quan trọng là: Thanh ghi lệnh IR (Instructor Register) và thanh ghi dữ liệu DR (Data Register).

- Thanh ghi IR: Để điều khiển LCD, người dùng phải "ra lệnh" thông qua tám đường bus DB0-DB7. Mỗi lệnh được nhà sản xuất LCD đánh địa chỉ rõ ràng. Người dùng chỉ việc cung cấp địa chỉ lệnh bằng cách nạp vào thanh ghi IR. Nghĩa là, khi ta nạp vào thanh ghi IR một chuỗi 8 bit, chíp HD44780 sẽ tra bảng mã lệnh tại địa chỉ mà IR cung cấp và thực hiện lệnh đó.

Ví dụ: Lệnh "hiển thị màn hình" có địa chỉ lệnh là 00001100 (DB7…DB0).



Hình 1.11:Sơ đồ khối của HD44780.

- Thanh ghi DR: Thanh ghi DR dùng để chứa dữ liệu 8 bit để ghi vào vùng RAM, DDRAM hoặc CGRAM (ở chế độ ghi) hoặc dùng để chứa dữ liệu từ 2 vùng RAM này gửi ra cho MPU (ở chế độ đọc). Nghĩa là, khi MPU ghi thông tin vào DR, mạch nội bên trong chíp sẽ tự động ghi thông tin này vào DDRAM hoặc CGRAM. Hoặc khi thông tin về địa chỉ được ghi vào IR, dữ liệu ở địa chỉ này trong vùng RAM nội của HD44780 sẽ được chuyển ra DR để truyền cho MPU. Vậy bằng cách điều khiển chân RS và R/W chúng ta có thể chuyển qua lại giữa 2 thanh ghi này trong khi giao tiếp với MPU. Bảng 1.2 tóm tắt lại các thiết lập đối với hai chân RS và R/W theo mục đích giao tiếp.

Bảng 1.2*:* Bảng chức năng chân RS và R/W theo mục đích sử dụng.

RS	RW	Ý nghĩa
0	0	Ghi vào thanh ghi IR để ra lệnh cho LCD (VD: cần display clear,...).
0	1	Đọc cờ bận ở DB7 và giá trị của bộ đếm địa chỉ ở DB0-DB6.
1	0	Ghi vào thanh ghi DR.
1	1	Đọc dữ liệu từ DR.

b) Cờ báo bận BF (Busy Flag):

Khi thực hiện các hoạt động bên trong chíp, mạch nội bên trong cần một khoảng thời gian để hoàn tất. Khi đang thực thi các hoạt động bên trong chíp như thế, LCD bỏ qua mọi giao tiếp với bên ngoài và bật cờ BF (thông qua chân DB7 khi có thiết lập RS=0, R/W=1) lên để báo cho MPU biết nó đang "bận". Dĩ nhiên, khi xong việc, nó sẽ đặt cờ BF lại mức 0.

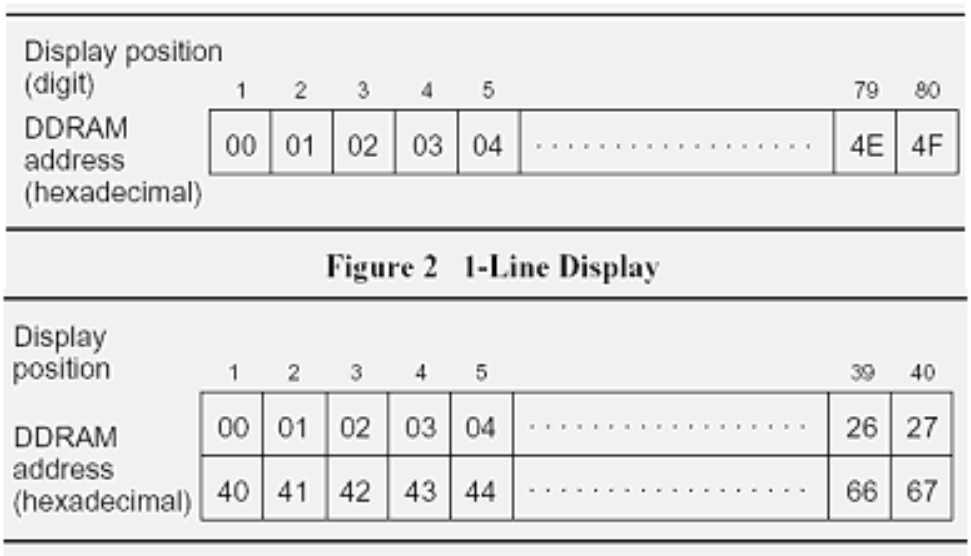
c) Bộ đếm địa chỉ AC (Address Counter):

Như trong sơ đồ khối, thanh ghi IR không trực tiếp kết nối với vùng RAM (DDRAM và CGRAM) mà thông qua bộ đếm địa chỉ AC. Bộ đếm này lại nối với 2 vùng RAM theo kiểu rẽ nhánh. Khi một địa chỉ lệnh được nạp

vào thanh ghi IR, thông tin được nối trực tiếp cho 2 vùng RAM nhưng việc chọn lựa vùng RAM tương tác đã được bao hàm trong mã lệnh. Sau khi ghi vào (hoặc đọc từ) RAM, bộ đếm AC tự động tăng lên (hoặc giảm đi) 1 đơn vị và nội dung của AC được xuất ra cho MPU thông qua DB0-DB6 khi có thiết lập RS=0 và R/W=1 (xem bảng 1.2). Lưu ý: Thời gian cập nhật AC không được tính vào thời gian thực thi lệnh mà được cập nhật sau khi cờ BF lên mức cao (not busy), cho nên khi lập trình hiển thị, bạn phải delay một khoảng tADD khoảng 4µS-5µS (ngay sau khi BF=1) trước khi nạp dữ liệu mới.

d) Vùng RAM hiển thị DDRAM (Display Data RAM):

Đây là vùng RAM dùng để hiển thị, nghĩa là ứng với một địa chỉ của RAM là một ô kí tự trên màn hình và khi bạn ghi vào vùng RAM này một mã 8 bit, LCD sẽ hiển thị tại vị trí tương ứng trên màn hình một kí tự có mã 8 bit đã cung cấp như hình 1.12.



Hình 1.12: Mối liên hệ giữa địa chỉ của DDRAM và vị trí hiển thị của LCD.

Vùng RAM này có 80x8 bits nhớ, nghĩa là chứa được 80 kí tự mã 8 bits. Những vùng RAM còn lại không dùng cho hiển thị có thể dùng như vùng RAM đa mục đích. Lưu ý là để truy cập vào DDRAM, ta phải cung cấp địa chỉ cho AC theo mã HEX.

e) Vùng ROM chứa kí tự CGROM (Character Generator ROM):

Vùng ROM này dùng để chứa các mẫu kí tự loại 5x8 hoặc 5x10 điểm ảnh/kí tự, và định địa chỉ bằng 8 bit. Tuy nhiên, nó chỉ có 208 mẫu kí tự 5x8 và 32 mẫu kí tự kiểu 5x10 (tổng cộng là 240 thay vì 256 mẫu kí tự). Người dùng không thể thay đổi vùng ROM này.

Table 2 **Example of Correspondence between EPROM Address Data and Character Pattern (5 × 8 Dots)**

EPROM Address												Data					
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	O4	O3	O2	O1	LSB O0	
								0	0	0	0	1	0	0	0	0	
								0	0	0	1	1	0	0	0	0	
								0	0	1	0	1	0	1	1	0	
								0	0	1	1	1	1	0	0	1	
								0	1	0	0	1	0	0	0	1	
								0	1	0	1	1	0	0	0	1	
								0	1	1	0	1	1	1	1	0	
0	1	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0	← Cursor position
								1	0	0	0	0	0	0	0	0	
								1	0	0	1	0	0	0	0	0	
								1	0	1	0	0	0	0	0	0	
								1	0	1	1	0	0	0	0	0	
								1	1	0	0	0	0	0	0	0	
								1	1	0	1	0	0	0	0	0	
								1	1	1	0	0	0	0	0	0	
								1	1	1	1	0	0	0	0	0	

Hình 1.13: Mối liên hệ giữa địa chỉ của ROM và dữ liệu tạo mẫu kí tự.

f) Vùng RAM chứa kí tự đồ họa CGRAM (Character Generator RAM):

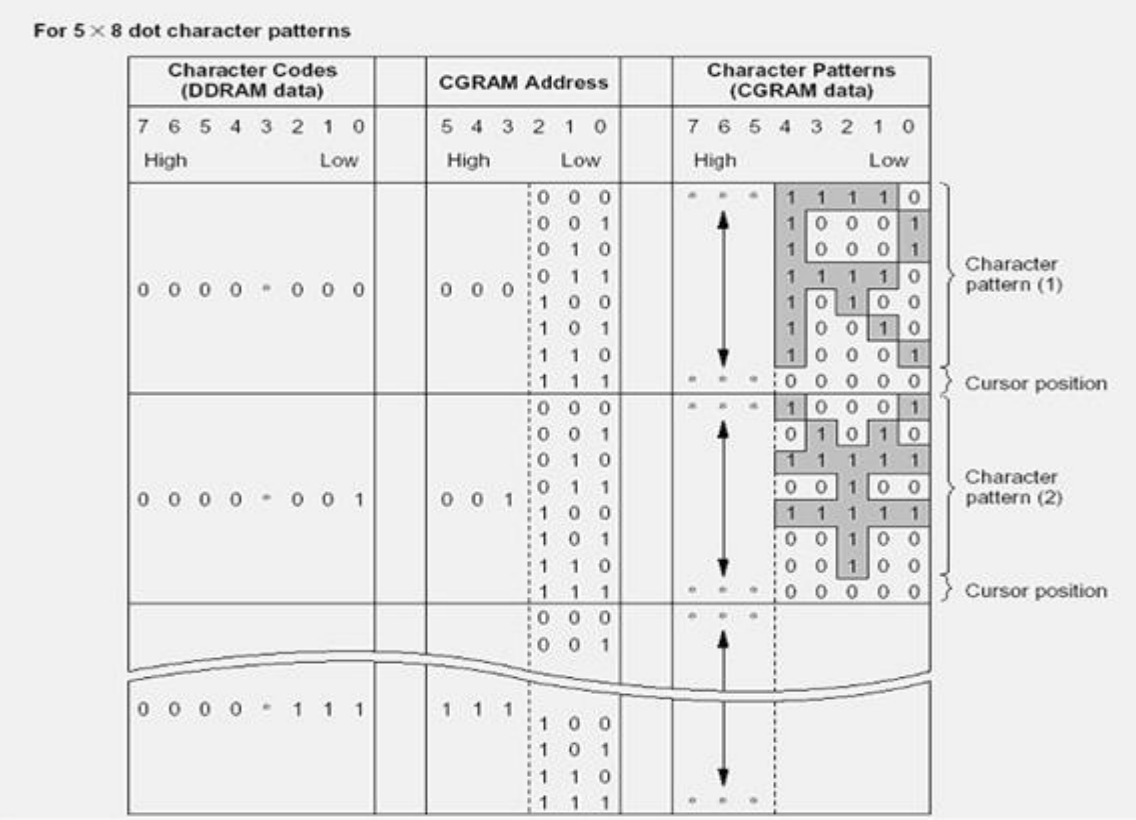
Như trên bảng mã kí tự, nhà sản xuất dành vùng có địa chỉ byte cao là 0000h để người dùng có thể tạo các mẫu kí tự đồ họa riêng. Tuy nhiên dung lượng vùng này rất hạn chế: Ta chỉ có thể tạo 8 kí tự loại 5x8 điểm ảnh, hoặc 4 kí tự loại 5x10 điểm ảnh. Để ghi vào CGRAM, xem hình1.13.

1.2.4. Tập lệnh của LCD

Trước khi tìm hiểu tập lệnh của LCD, sau đây là một vài chú ý khi giao tiếp với LCD:

Tuy trong sơ đồ khối của LCD có nhiều khối khác nhau, nhưng khi lập trình điều khiển LCD ta chỉ có thể tác động trực tiếp được vào 2 thanh ghi DR

và IR thông qua các chân DBx, và ta phải thiết lập chân RS, R/W phù hợp để chuyển qua lại giữa 2 thanh ghi này (xem bảng1.2).



Hình 1.14: Mối liên hệ giữa địa chỉ của CGRAM, dữ liệu CGRAM, và mã kí tự.

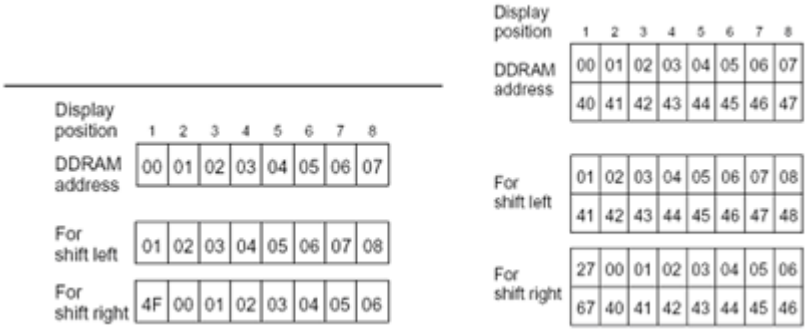
Với mỗi lệnh, LCD cần một khoảng thời gian để hoàn tất, thời gian này có thể khá lâu đối với tốc độ của MPU, nên ta cần kiểm tra cờ BF hoặc đợi (delay) cho LCD thực thi xong lệnh hiện hành mới có thể ra lệnh tiếp theo.

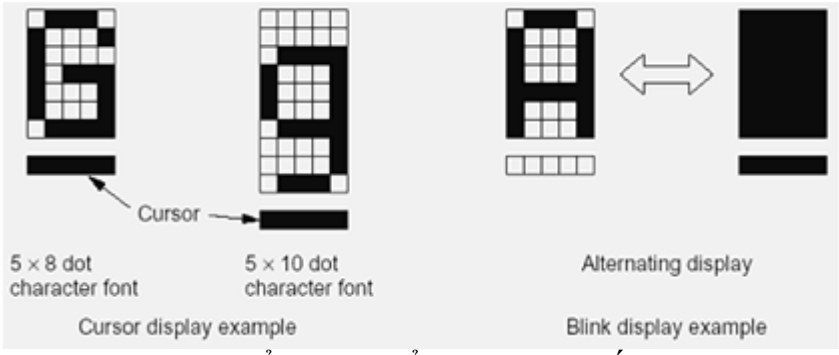
Địa chỉ của RAM (AC) sẽ tự động tăng (giảm) 1 đơn vị, mỗi khi có lệnh ghi vào RAM (Điều này giúp chương trình gọn hơn).

Các lệnh của LCD có thể chia thành 4 nhóm như sau:

- Các lệnh về kiểu hiển thị. VD : Kiểu hiển thị (1 hàng/2 hàng), chiều dài dữ liệu (8 bit/4 bit),…
- Chỉ định địa chỉ RAM nội.
- Nhóm lệnh truyền dữ liệu trong RAM nội.
- Các lệnh còn lại.

Bảng 1.3: Các tập lệnh của LCD.

Tên lệnh	Hoạt động	Thời gian chạy
Clear Display	Mã lệnh: DBx=DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx= 0 0 0 0 0 0 0 1 Lệnh Clear Display (xóa hiển thị) sẽ ghi một khoảng trống (mã hiển thị kí tự 20H) vào tất cả ô nhớ trong DDRAM, sau đó trả bộ đếm địa chỉ AC=0, trả lại hiển thị gốc nếu nó bị thay đổi, nghĩa là: Tắt hiển thị, con trỏ dời về góc trái (hàng đầu tiên), chế độ tăng AC.	
Return home	Mã lệnh:DBx=DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx= 0 0 0 0 0 0 0 *	1.52ms
Entry mode set	Mã lệnh:DBx=DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx= 0 0 0 0 0 1 [I/D] [S] I/D: Tăng (I/D=1) hoặc giảm (I/D=0) bộ đếm địa chỉ hiển thị AC 1 đơn vị mỗi khi có hành động ghi hoặc đọc vùng DDRAM. Vị trí con trỏ cũng di chuyển theo sự tăng giảm này. S: Khi S=1 toàn bộ nội dung hiển thị bị dịch sang phải (I/D=0) hoặc sang trái (I/D=1) mỗi khi có hành động ghi vùng DDRAM. Khi S=0: không dịch nội dung hiển thị. Nội dung hiển thị không dịch khi đọc DDRAM hoặc đọc/ghi vùng CGRAM.  ***Hình 1.15***: Hoạt động dịch trái và dịch phải nội dung hiển thị.	37µs
Display on/off control	Mã lệnh:DBx=DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx= 0 0 0 0 1 [D] [C] [B] D: Hiển thị màn hình khi D=1 và ngược lại. Khi tắt hiển thị, nội dung DDRAM không thay đổi. C: Hiển thị con trỏ khi C=1 và ngược lại. Vị trí và hình dạng con trỏ, xem hình1.16. B: Nhấp nháy kí tự tại vị trí con trỏ khi B=1 và ngược lại. Chu kì nhấp nháy khoảng 409,6ms khi mạch dao động nội LCD là 250kHz.	37µs

<table>
<tr>
<td></td>
<td> 

Hình 1.16: Kiểu con, kiểu kí tự và nhấp nháy kí tự.</td>
<td></td>
</tr>
<tr>
<td>Cursor or display shift</td>
<td>Mã lệnh:DBx=DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0
DBx= 0 0 0 1 [S/C][R/L] * *
Lệnh Cursor or display shift dịch chuyển con trỏ hay dữ liệu hiển thị sang trái mà không cần hành động ghi/đọc dữ liệu. Khi hiển thị kiểu 2 dòng, con trỏ sẽ nhảy xuống dòng dưới khi dịch qua vị trí thứ 40 của hàng đầu tiên. Dữ liệu hàng đầu và hàng 2 dịch cùng một lúc. Chi tiết sử dụng xem hình sau:
<table>
<tr><th>S/C</th><th>R/L</th><th>Hoạt động</th></tr>
<tr><td>0</td><td>0</td><td>Dịch vị trí con trỏ sang trái (Nghĩa là giảm AC một đơn vị).</td></tr>
<tr><td>0</td><td>1</td><td>Dịch vị trí con trỏ sang phải (Tăng AC lên 1 đơn vị).</td></tr>
<tr><td>1</td><td>0</td><td>Dịch toàn bộ nội dung hiển thị sang trái, con trỏ cũng dịch theo.</td></tr>
<tr><td>1</td><td>1</td><td>Dịch toàn bộ nội dung hiển thị sang phải, con trỏ cũng dịch theo.</td></tr>
</table>
Hình 1.17: Chi tiết sử dụng lệnh.</td>
<td>37µs</td>
</tr>
<tr>
<td>Funtion set</td>
<td>Mã lệnh:DBx=DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0
DBx= 0 0 1 [DL] [N] [F] * *
DL: Khi DL=1, LCD giao tiếp với MPU bằng giao thức 8 bit (từ bit DB7 đến DB0). Ngược lại, giao thức giao tiếp là 4 bit (từ bit DB7 đến bit DB0). Khi chọn giao thức 4 bit, dữ liệu được truyền/nhận 2 lần liên tiếp với 4 bit cao gửi/nhận trước, 4 bit thấp gửi/nhận sau.
N: Thiết lập số hàng hiển thị. Khi N=0: hiển thị 1 hàng, N=1: hiển thị 2 hàng.
F: Thiết lập kiểu kí tự. Khi F=0: kiểu kí tự 5x8 điểm ảnh, F=1: kiểu kí tự 5x10 điểm ảnh.
* Chú ý:
• Chỉ thực hiện thay đổi Function set ở đầu chương trình. Và sau khi được thực thi 1 lần, lệnh thay đổi Function set không được LCD chấp nhận nữa ngoại trừ thiết lập chuyển đổi giao thức giao tiếp.
Không thể hiển thị kiểu kí tự 5x10 điểm ảnh ở kiểu hiển</td>
<td>37µs</td>
</tr>
</table>

	thị 2 hàng.	
Set CGRAM address	Mã lệnh:DBx=DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx=0 1[ACG][ACG][ACG][ACG][ACG][ACG][ACG] Lệnh này ghi vào AC địa chỉ của CGRAM. Kí hiệu [ACG] chỉ 1 bit của chuỗi dữ liệu 6 bit. Ngay sau lệnh này là lệnh đọc/ghi dữ liệu từ CGRAM tại địa chỉ đã được chỉ định.	
Set DDRAM address	Mã lệnh:DBx=DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx= 1 [AD][AD][AD][AD] [AD] [AD] [AD] Lệnh này ghi vào AC địa chỉ của DDRAM, dùng khi cần thiết lập tọa độ hiển thị mong muốn. Ngay sau lệnh này là lệnh đọc/ghi dữ liệu từ DDRAM tại địa chỉ đã được chỉ định. Khi ở chế độ hiển thị 1 hàng, địa chỉ có thể từ 00H đến 4FH. Khi ở chế độ hiển thị 2 hàng, địa chỉ từ 00h đến 27H cho hàng thứ nhất, và từ 40h đến 67h cho hàng thứ 2.	37µs
Read BF and address	Mã lệnh:DBx=DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx= [BF][AD][AD][AD][AD][AD][AD][AD] RS=0; R/W=1 Như đã đề cập trước đây, khi cờ BF bật, LCD đang làm việc và lệnh tiếp theo (nếu có) sẽ bị bỏ qua nếu cờ BF chưa về mức thấp. Cho nên, khi lập trình điều khiển, bạn phải kiểm tra cờ BF trước khi ghi dữ liệu vào LCD. Khi đọc cờ BF, giá trị của AC cũng được xuất ra các bit [AC]. Nó là địa chỉ của CG hay DDRAM là tùy thuộc vào lệnh trước đó.	0µs
Write data to CG or DDRAM	Mã lệnh:DBx=DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx= [Write data] (RS=1, R/W=0) Khi thiết lập RS=1, R/W=0, dữ liệu cần ghi được đưa vào các chân DBx từ mạch ngoài sẽ được LCD chuyển vào trong LCD tại địa chỉ được xác định từ lệnh ghi địa chỉ trước đó. Sau khi ghi, bộ đếm địa chỉ AC tự động tăng/giảm 1 tùy theo thiết lập Entry mode. Lưu ý là thời gian cập nhật AC không tính vào thời gian thực thi lệnh.	37µs tADD 4µs
Read data from CG or DDRAM	Mã lệnh:DBx=DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 DBx= [Read data] (RS=1, R/W=1) Khi thiết lập RS=1, R/W=1, dữ liệu từ CG/DDRAM được chuyển ra MPU thông qua các chân DBx (địa chỉ và vùng RAM đã được xác định bằng lệnh ghi địa chỉ trước đó). Sau khi đọc, AC tự động tăng/giảm 1 tùy theo thiết lập Entry mode, tuy nhiên nội dung hiển thị không bị dịch bất chấp chế độ Entrymode.	37µs tADD 4µs

1.2.5. Đặc tính của các chân giao tiếp

LCD sẽ bị hỏng nghiêm trọng, hoặc hoạt động sai lệch nếu bạn vi phạm khoảng đặc tính điện sau đây:

Bảng 1.4: Đặc tính làm việc điển hình.

Chân cấp nguồn (Vcc-GND)	Min:-0.3V , Max+7V
Các chân ngõ vào (DBx,E,…)	Min:-0.3V , Max:(Vcc+0.3V)
Nhiệt độ hoạt động	Min:-30C , Max:+75C
Nhiệt độ bảo quản	Min:-55C , Max:+125C

Đặc tính điện làm việc điển hình: (Đo trong điều kiện hoạt động V_{cc} = 4.5V đến 5.5V, T = -30 đến $+75^0$C).

Bảng 1.5: Miền làm việc bình thường.

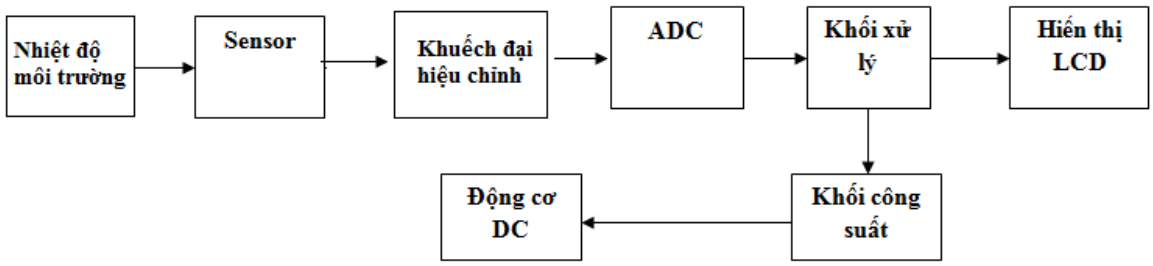
Chân cấp nguồn Vcc-GND	2.7V đến 5.5V
Điện áp vào mức cao V_{IH}	2.2V đến Vcc
Điện áp vào mức thấp V_{IL}	-0.3V đến 0.6V
Điện áp ra mức cao (DB0-DB7)	Min 2.4V (khi I_{OH} = -0.205mA)
Điện áp ra mức thấp (DB0-DB7)	Max 0.4V (khi I_{OL} = 1.2mA)
Dòng điện ngõ vào (input leakage current) I_{LI}	-1uA đến 1uA (khi V_{IN} = 0 đến Vcc)
Dòng điện cấp nguồn I_{CC}	350uA(typ.) đến 600uA
Tần số dao động nội f_{OSC}	190kHz đến 350kHz (điển hình là 270kHz)

CHƯƠNG 2.

THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN ĐỘNG CƠ DC THEO NHIỆT ĐỘ

2.1. SƠ ĐỒ KHỐI

Với yêu cầu của đề tài là thiết kế hệ thống điều khiển động cơ DC theo nhiệt độ, tức là từ nhiệt độ đo được trong môi trường, hệ thống điều khiển tốc độ động cơ DC quay nhanh hay chậm. Ta có sơ đồ khối hệ thống trong hình 2.1.



Hình 2.1: Sơ đồ khối hệ thống điều khiển động cơ DC theo nhiệt độ.

Với sơ đồ này ta sử dụng cảm biến đo nhiệt độ môi trường. Điện áp ra của cảm biến được khuếch đại, hiệu chỉnh để phù hợp với đầu vào của ADC. Khối ADC làm nhiệm vụ chuyển đổi tín hiệu tương tự thành tín hiệu số đưa vào khối xử lý. Khối xử lý làm nhiệm vụ nhận giá trị đo, từ đó điều khiển động cơ DC quay với tốc độ phù hợp. Trên sơ đồ sử dụng khối hiển thị để người sử dụng có thể theo dõi được các thông số và thao tác thực hiện.

2.2. THIẾT KẾ CÁC KHỐI

2.2.1. Mạch đo nhiệt độ

Nhiệt độ là một đại lượng vật lý vô hướng. Để đo đạc và tính toán giá

trị của nó ta phải dùng các bộ cảm biến. Cảm biến đo nhiệt độ ở đây em chọn loại phổ biến là LM335. Bộ cảm biến LM335 đưa ra điện áp 10mV cho mỗi sự thay đổi 1^oK.

❖ Thông số kĩ thuật:

- Là cảm biến nhiệt độ cho ra các mức điện áp tương ứng với độ Kelvin

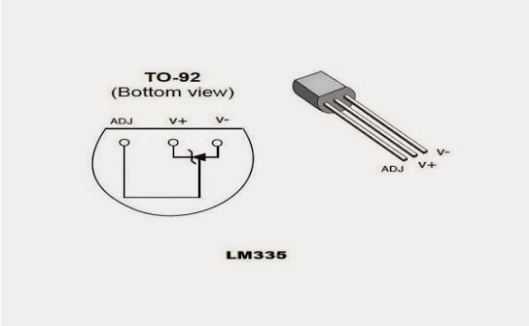
Sai số +/- 1^0C

- Hoạt động trong dải từ 400uA - 5 mA

- Trở kháng nội < 1 Ω

- Dải nhiệt độ đo được -55^0C đến 150^0C

❖ Sơ đồ chân và mạch đo:

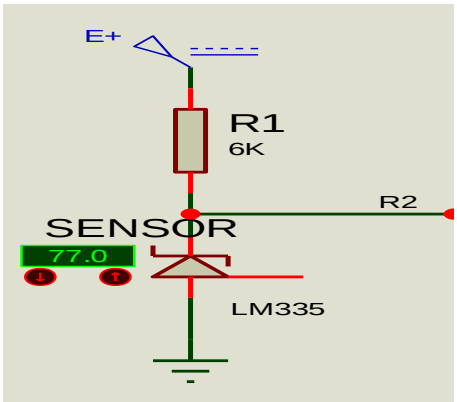


Hình 2.2: Sơ đồ chân của LM335.

ADJ - Chân hiệu chuẩn.

V+ - Chân cấp nguồn 15V.

V- - Chân GND.



Hình 2.3: Kết nối VĐK.

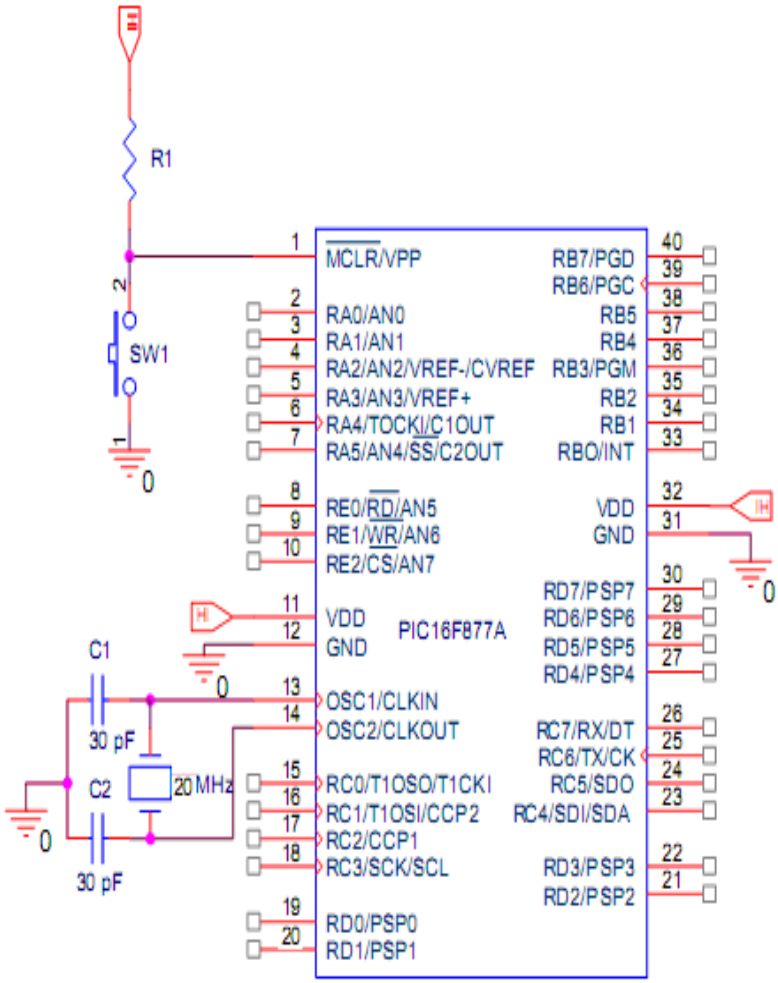
- Nhiệt độ tính theo độ Kelvin T = V_{out} /(10mV).

Từ độ Kelvin ta có thể đổi ra độ C theo công thức:

$$T_{(độ\ C)} = T_{(độ\ Kenvin)} - 273,15\ .$$

2.2.2. Khối xử lý

Đây là khối quan trọng và quản lý toàn bộ hoạt động của mạch. Nhiệm vụ chính của khối là nhận về giá trị nhiệt độ, điều khiển động cơ. Giá trị đo từ LM335 được cho qua bộ ADC, mạch này ta có thể dùng IC ngoài hoặc được tích hợp trong một số dòng vi điều khiển. Ở đây em dùng PIC16F877A, vừa chuyển đổi ADC, hiển thị LCD, vừa điều khiển động cơ bằng phương pháp PWM.



Hình 2.4: Sơ đồ nguyên lý của PIC16F877A trong mạch.

2.2.3. Khối ADC (tích hợp trong PIC16F877A)

ADC (Analog to Digital Converter) là bộ chuyển đổi tín hiệu giữa hai dạng tương tự và số. PIC16F877A có 8 ngõ vào analog (RA4:RA0 và RE2:RE0). Hiệu điện thế chuẩn V_{REF} có thể được lựa chọn là V_{DD}, V_{SS} hay hiệu điện thế chuẩn được xác lập trên hai chân RA2 và RA3. Kết quả chuyển đổi từ tín hiệu tương tự sang tín hiệu số là 10 bit số tương ứng và được lưu trong hai thanh ghi ADRESH : ADRESL. Khi không sử dụng bộ chuyển đổi ADC, các thanh ghi này có thể được sử dụng như các thanh ghi thông thường khác. Khi quá trình chuyển đổi hoàn tất, kết quả sẽ được lưu vào hai thanh ghi ADRESH : ADRESL, bit GO/$\overline{\text{DONE}}$ (ADCON0<2>) được xóa về số 0 và cờ ngắt ADIF được set.

Quá trình chuyển đổi từ tương tự sang số bao gồm các bước:

+ Thiết lập các thông số cho bộ chuyển đổi ADC:
  - Chọn ngõ vào analog, chọn điện áp lấy mẫu (dựa trên các thông số của thanh ghi ADCON1).
  - Chọn kênh chuyển đổi AD (thanh ghi ADCON0).
  - Chọn xung clock cho kênh chuyển đổi AD (thanh ghi ADCON0).
  - Cho phép bộ chuyển đổi AD hoạt động (thanh ghi ADCON0).
+ Thiết lập các cờ ngắt cho AD:
  - Clear bit ADIF.
  - Set bit ADIE.
  - Set bit PEIE.
  - Set bit GIE.
+ Đợi cho tới khi quá trình lấy mẫu hoàn tất.
+ Bắt đầu quá trình chuyển đổi (set bit GO/$\overline{\text{DONE}}$).
+ Đợi cho tới khi quá trình chuyển đổi hoàn tất bằng cách :
  - Kiểm tra bit GO/$\overline{\text{DONE}}$. Nếu GO/$\overline{\text{DONE}}$ =0, quá trình chuyển đổi đã hoàn tất.

- Kiểm tra cờ ngắt.

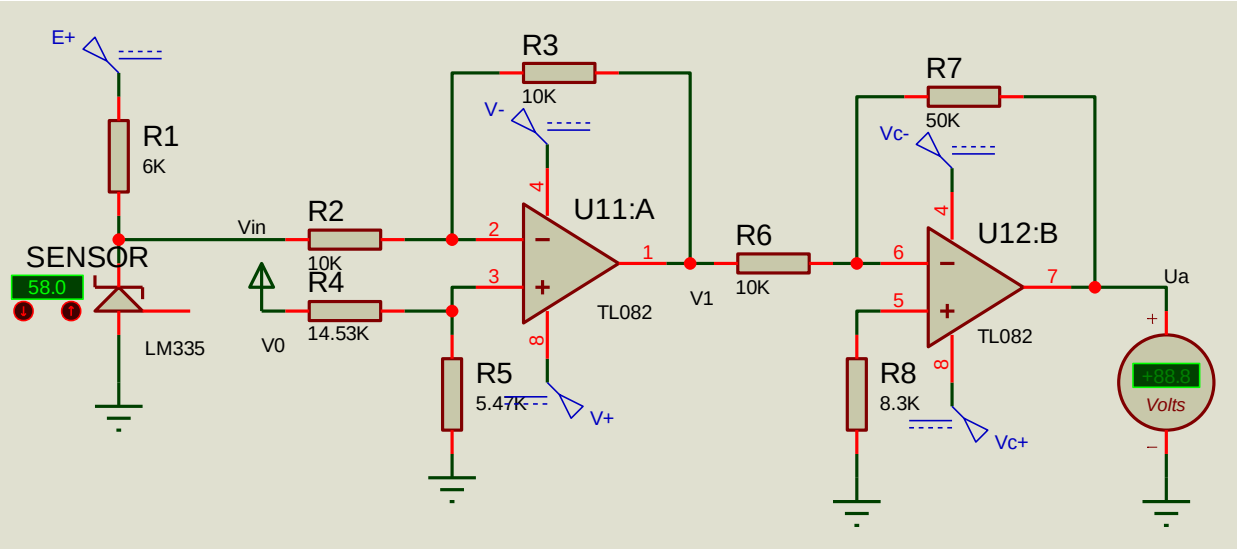
+ Đọc kết quả chuyển và xóa cờ ngắt, set bit GO/$\overline{\text{DONE}}$ (nếu cần tiếp tục chuyển đổi).

+ Tiếp tục thực hiện các bước 1 và 2 cho quá trình chuyển đổi tiếp theo.

Các thanh ghi liên quan đến bộ chuyển đổi ADC gồm:

- INTCON (địa chỉ 0Bh, 8Bh, 10Bh,18Bh): cho phép ngắt (các bit GIE, PEIE).
- PIR1(địa chỉ 0Ch): chứa cờ ngắt AD (bit ADIF).
- PIE1 (địa chỉ 8Ch): chứa bit điều khiển AD (ADIE).
- ADRESH (địa chỉ 1Eh) và ADRESL (địa chỉ 9Eh) : các thanh ghi chứa kết quả chuyển đổi AD.
- ADCON0 (địa chỉ 1Fh) và ADCON1 (địa chỉ 9Fh): xác lập các thông số cho bộ chuyển đổi AD.
- PORTA (địa chỉ 05h) và TRISA (địa chỉ 85h): liên quan đến các ngõ vào analog ở PORTA.
- PORTE (địa chỉ 09h) và TRISE (địa chỉ 89h): liên quan đến các ngõ vào analog ở PORTE.

2.2.4. Khối khuếch đại hiệu chỉnh



Hình 2.5: Mạch khuếch đại hiệu chỉnh.

Ở đây, mạch khuếnh đại hiệu chỉnh được dùng để chuyển dải điện áp trong LM335 (V_{out}=2,73÷3,73V tương ứng với dải nhiệt độ đo 0÷100^{0}C) thành dải từ 0÷5V rồi đưa vào bộ ADC tích hợp trong PIC16F877A.

Mạch khuếch đại hiệu chỉnh được tích hợp từ mạch trừ và mạch khuếch đại. Cụ thể, dải điện áp ra từ LM335 (V_{in}=2,73÷3,73V) qua mạch trừ thành dải V_1=(0÷(-1))V, sau đó qua mạch khuếch đại với hệ số khuếch đại (-5) chuyển thành dải 0÷5V.

❖ Tính chọn giá trị điện trở trong mạch:

$$V_1 = \frac{R_2 + R_3}{R_4 + R_5}\frac{R_5}{R_2}V_0 - \frac{R_3}{R_2}V_{in}$$

Chọn R_2+R_3=R_4+R_5

$$V_1 = \frac{R_5}{R_2}V_0 - \frac{R_3}{R_2}V_{in}$$

Thay V_{in}=2,73 và V_{in}=3,73, V_1=0 và V_1=(-1) vào ta được:

$$\frac{R_5}{R_2}V_0 - 2{,}73\frac{R_3}{R_2} = 0$$

$$\frac{R_5}{R_2}V_0 - 3{,}73\frac{R_3}{R_2} = -1$$

=> R_3=R_2 ; R_5.V_0 - 2,73.R_3=0

Chọn R_2=10kΩ =>R_3=10kΩ.

Chọn V_0=5V =>R_4=14,54kΩ.

Có :

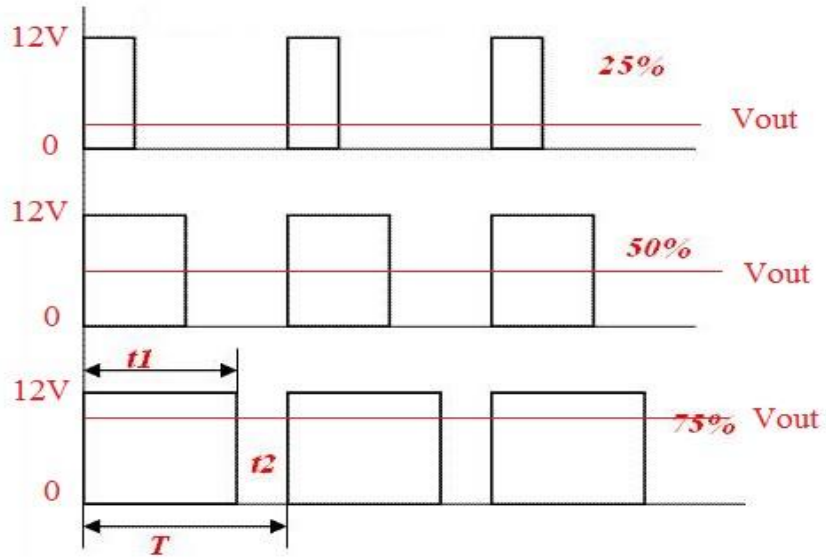
$$\frac{R_7}{R_6} = 5$$

Chọn R_6=10kΩ =>R_7=50kΩ; R_8=8,3kΩ.

2.2.5. Khối công suất

Ở đây sử dụng phương pháp PWM (Pulse Width Modulation) để điều khiển tốc độ của động cơ DC. Phương pháp điều chế PWM là phương pháp

điều chỉnh điện áp ra tải hay nói cách khác là phương pháp điều chế dựa trên sự thay đổi độ rộng của chuỗi xung vuông dẫn đến sự thay đổi điện áp ra. Các xung PWM khi biến đổi thì có cùng 1 tần số và khác nhau về độ rộng của sườn dương hay là sườn âm. Điều khiển động cơ sử dụng phương thức điều chế xung PWM là một trong các phương thức được sử dụng rất rộng rãi trong điều khiển động cơ ứng dụng trong công nghiệp, dân dụng cũng như trong nhiều ứng dụng khác, ngoài ra PWM còn tham gia và điều chế các mạch nguồn như là: boot, buck, nghịch lưu 1 pha và 3 pha ... Điều đặc biệt là PWM chuyên dùng để điều khiển các phần tử điện tử công suất có đường đặc tính là tuyến tính khi có sẵn 1 nguồn 1 chiều cố định. Như vậy PWM được ứng dụng rất nhiều trong các thiết bị điện, điện tử.



Hình 2.6: Xung PWM và điện áp đầu ra

Hình 2.6. là sơ đồ đặc tả xung PWM và cách thức tính điện áp đầu ra đưa tới động cơ. Nhìn vào sơ đồ ta có

- Chu kỳ của xung PWM là thời gian T.
- Thời gian phát xung PWM là t1
- Thời gian nghỉ không phát xung là t2

Công thức tính giá trị trung bình của điện áp ra tải:

$$V_{out} = V_{in}.duty$$

Trong đó:

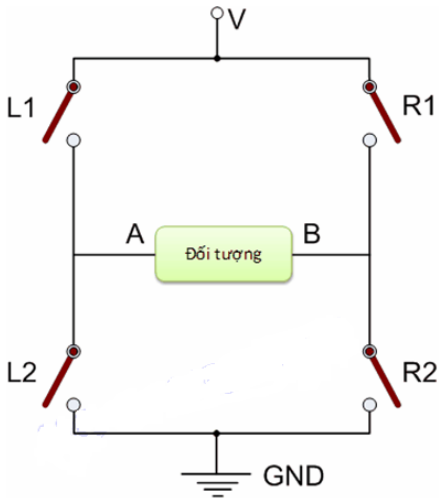
- Vout là điện áp ra
- Vin là điện áp đầu vào
- Duty là % thời gian phát xung được tính bằng:

$$Duty = t_1/T.100\%$$

Trong khối điều khiển, PIC16F877A điều khiển động cơ thông qua quá trình tạo xung PWM rồi đưa vào mạch cầu H tích hợp trong IC Driver L293D.

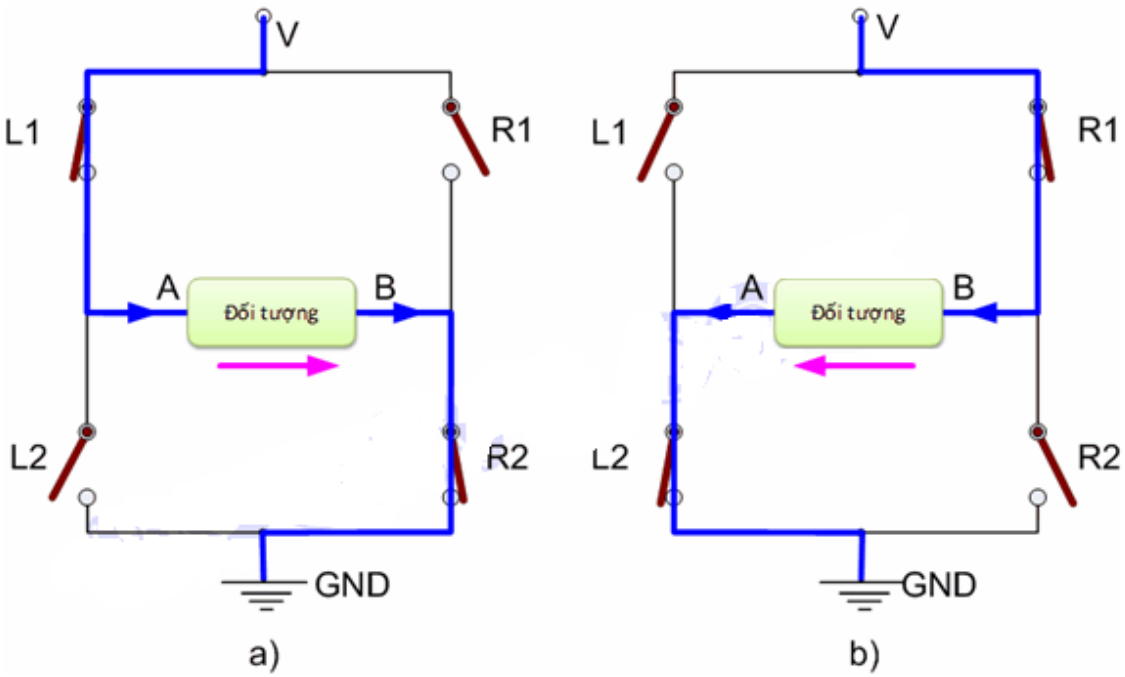
❖ Nguyên lý hoạt động mạch cầu H:

Mạch cầu H là một trong những mạch được sử dụng rộng rãi cho việc điều khiển động cơ.



Hình 2.7: Mạch cầu H.

- Trong hình 2.7, "đối tượng" là động cơ DC mà ta cần điều khiển, "đối tượng" này có 2 đầu A và B, mục đích điều khiển là cho phép dòng điện qua "đối tượng" theo chiều A đến B hoặc B đến A. Thành phần chính tạo nên mạch cầu H chính là 4 "khóa" L1, L2, R1 và R2 (L: Left, R: Right). Ở điều kiện bình thường 4 khóa này "mở", mạch cầu H không hoạt động. Hoạt động của mạch cầu H được mô tả trong hình 2.8a và 2.8b.



Hình 2.8: Nguyên lý hoạt động mạch cầu H.

+ Ở hình 2.8a L1 và R2 được "đóng lại" , L2 và R1 vẫn mở, dòng điện sẽ chạy từ V qua khóa L1 và đi qua đối tượng đến đầu B của nó trước khi qua R2 về GND.

+ Ở hình 2.8b L2 và R1 được "đóng lại", L1 và R2 mở, dòng điện sẽ chạy từ V qua khóa R1 và đi qua đối tượng đến đầu B của nó trước khi qua L2 về GND.

Như vậy, có thể dùng mạch cầu H để đảo chiều dòng điện qua một "đối tượng" (hay cụ thể, đảo chiều quay động cơ).

❖ IC Driver L293D: là hai bộ mạch cầu H được tích hợp trong cùng IC.

Thông số kĩ thuật L293D:

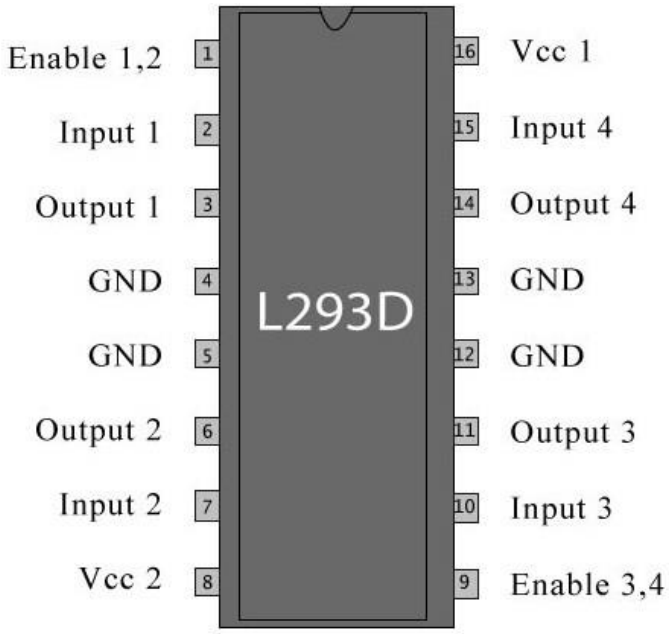
+ Điện áp cực đại: 36V.

+ Dòng ra cực đại: 1.2A.

+ Dải nhiệt độ hoạt động: -40 ~ 150^{o}C.

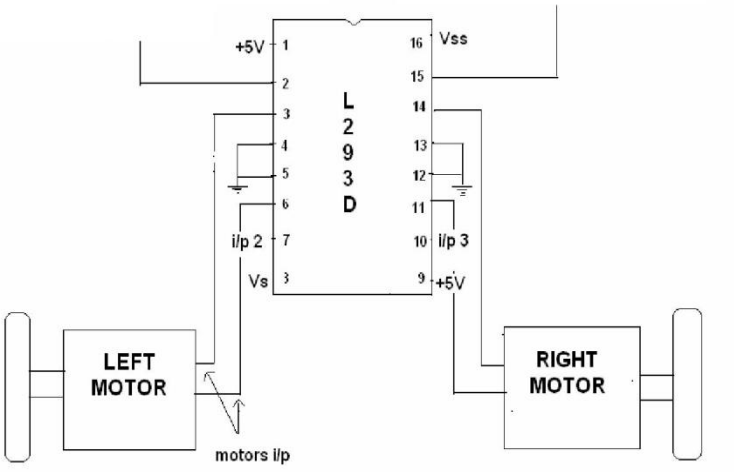
L293D là một chip tích hợp 2 mạch cầu H trong gói 16 chân. Tất cả các mạch kích, mạch cầu đều được tích hợp sẵn. L293D có điện áp danh nghĩa

cao (lớn nhất 36V) và dòng điện danh nghĩa lớn nhất 1.2A nên rất thích hợp cho các các ứng dụng công suất nhỏ như các động cơ DC loại nhỏ và vừa.



Hình 2.9: Sơ đồ chân L293D.

Có 2 mạch cầu H trên mỗi chip L293D nên có thể điều khiển 2 đối tượng chỉ với 1 chip này. Mỗi mạch cầu bao gồm 1 đường nguồn Vs (thật ra là đường chung cho 2 mạch cầu), một đường current sensing (cảm biến dòng), phần cuối của mạch cầu H không được nối với GND mà bỏ trống cho người dùng nối một điện trở nhỏ gọi là sensing resistor.

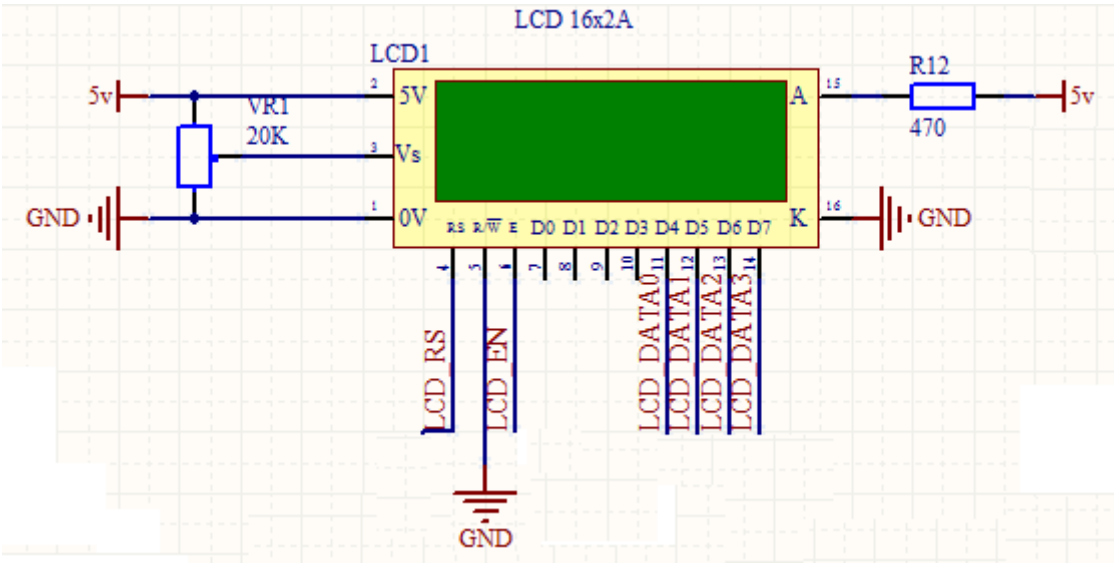


Hình 2.10: Sơ đồ kết nối L293D.

Động cơ sẽ được nối với 2 đường OUT1, OUT2 (hoặc OUT3, OUT4 nếu dùng mạch cầu bên phải). Một chân En (EnA và EnB cho 2 mạch cầu) cho phép mạch cầu hoạt động, khi chân En được đặt lên mức cao, mạch cầu sẵn sàng hoạt động.

2.2.6. Khối hiển thị

Để thuận tiện cho việc hiển thị kí tự và chế độ cài đặt trạng thái điều khiển, ở em đây sử dụng LCD_DM 16x2A.



Hình 2.11: Sơ đồ nguyên lí của LCD16x2A.

LCD16x2A là loại 2 dòng, 16 kí tự, sử dụng nguồn nuôi thấp (từ 2,5 đến 5V). Có thể hoạt động ở hai chế độ 4 bit hoặc 8 bit.

2.2.7. Motor DC

Ngày nay, động cơ điện một chiều được sử dụng rộng rãi trong các ứng dụng công nghiệp vì nó cung cấp công suất cơ không đổi hoặc moment không đổi, tốc độ động cơ được điều chỉnh trong phạm vi rộng, điều khiển tốc độ hoặc vị trí một cách chính xác, vận hành hiệu quả ở dải tốc độ rộng, tăng tốc và giảm tốc nhanh, và đáp ứng nhanh với các tín hiệu phản hồi.

Ứng dụng của động cơ điện một chiều rất rộng rãi. Như với động cơ công suất nhỏ, động cơ được sử dụng trong các thiết bị điều khiển, động cơ

cần gạt nước, động cơ quạt, động cơ của bộ khởi động và một số động cơ chấp hành khác.Với động cơ công suất lớn: động cơ truyền động trong các băng tải, máy bơm, cần trục, cần cẩu, xe nâng, quạt, máy cán thép và nhôm,....

2.2.7.1. Cấu tạo và nguyên lí làm việc của động cơ điện một chiều

a) Cấu tạo:

Cấu tạo của động cơ gồm: Stato (phần tĩnh), rôto (phần động) và phần chỉnh lưu (chổi than và cổ góp).

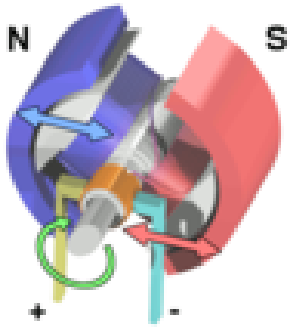
- Stator thường là một hay nhiều cặp nam châm vĩnh cửu, hay nam châm điện.
- Rotor có các cuộn dây quấn và được nối với nguồn điện một chiều.
- Bộ phận chỉnh lưu có nhiệm vụ là đổi chiều dòng điện trong khi chuyển động quay của rotor là liên tục. Bộ phận này gồm bộ cổ góp và bộ chổi than tiếp xúc với cổ góp. Vành góp nằm trên phần ứng và bao gồm một số phiến góp hình cung tròn được lắp ráp vào một đầu hình trụ gắn và cách điện với trục.

b) Nguyên lí làm việc của động cơ điện một chiều:

Khi có dòng điện chạy qua cuộn dây quấn xung quanh một lõi sắt non, cạnh phía bên cực dương sẽ bị tác động bởi một lực hướng lên, trong khi cạnh đối diện lại bị tác động bằng một lực hướng xuống theo nguyên lí bàn tay trái của Fleming. Các lực này gây tác động quay lên cuộn dây, và làm cho rotor quay.

Để làm cho rotor quay liên tục và đúng chiều, một bộ cổ góp điện sẽ làm chuyển mạch dòng điện sau mỗi vị trí ứng với ½ chu kì. Khi mặt của cuộn dây song song với các đường sức từ trường, tức lực quay của động cơ bằng 0 khi cuộn dây lệch 90^o so với phương ban đầu của nó, khi đó rotor sẽ quay theo quán tính.

Trong các máy điện một chiều lớn, người ta có nhiều cuộn dây nối ra nhiều phiến góp khác nhau trên cổ góp. Nhờ vậy dòng điện và lực quay được liên tục và hầu như không bị thay đổi theo các vị trí khác nhau của rotor.



Hình 2.12: Động cơ DC.

Dòng điện I đi qua mạch phần ứng của động cơ được biểu diễn bằng phương trình sau:

$$I_{ư} = \frac{U - E_{ư}}{R_{ư}}$$

U là điện áp đặt vào mạch phần ứng, $R_{ư}$ là điện trở mạch phần ứng, và $E_{ư}$ là sức điện động phần ứng.

2.2.7.2. Các phương pháp điều khiển motor DC

Từ phương trình tính tốc độ:

$$\omega = \frac{U - I_{ư}.R_{ư}}{k.\phi}$$

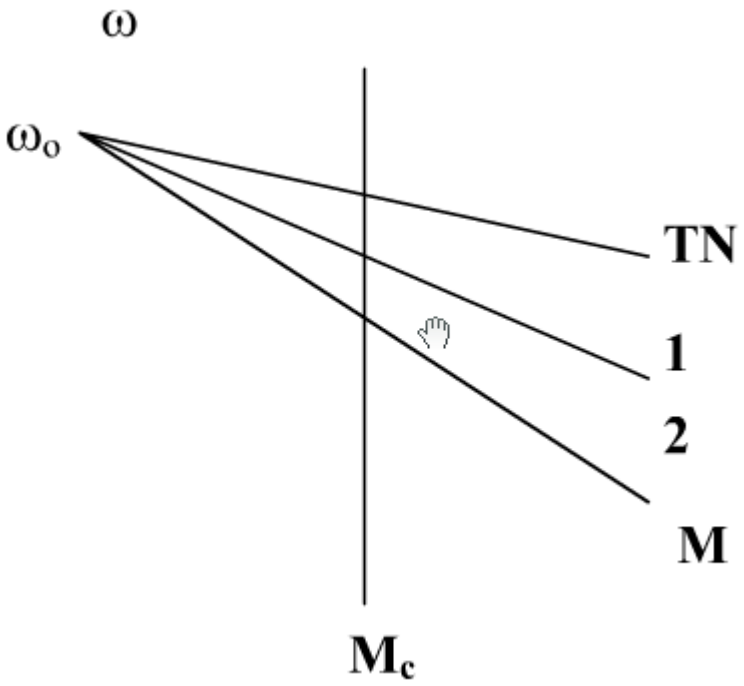
Để điều chỉnh ω :

- Điều chỉnh U.
- Điều chỉnh $R_{ư}$ bằng cách thêm R_p vào mạch phần ứng.
- Điều chỉnh từ thông ϕ.

a) Điều chỉnh tốc độ bằng phương pháp dùng thêm R_p:

Mắc nối tiếp R_p vào phần ứng làm $R_{ư}$ tăng lên, ω giảm, độ dốc của đường đặc tính giảm. Các đường 1, 2 là đường đặc tính sau khi tăng $R_{ư}$, đường TN là đường đặc tính tự nhiên của động cơ ban đầu.

Ưu điểm của phương pháp này là đơn giản, tốc độ điều chỉnh liên tục, nhưng do thêm R_p nên tổn hao tăng, không kinh tế.



Hình 2.13: Đặc tính cơ của motor DC khi thêm R_p.

b) Điều khiển từ thông:

Điều chỉnh từ thông kích thích của động cơ điện một chiều là điều chỉnh moment điện từ của động cơ M=K.ϕ.$I_ư$ và sức điện động quay của động cơ $E_ư$ =K.ϕ.ω. Khi từ thông giảm thì tốc độ quay của động cơ tăng lên trong phạm vi giới hạn của việc thay đổi từ thông. Nhưng theo công thức trên khi ϕ thay đổi thì moment, dòng điện I cũng thay đổi nên khó tính được chính xác dòng điều khiển và moment tải.

c) Điều khiển điện áp phần ứng:

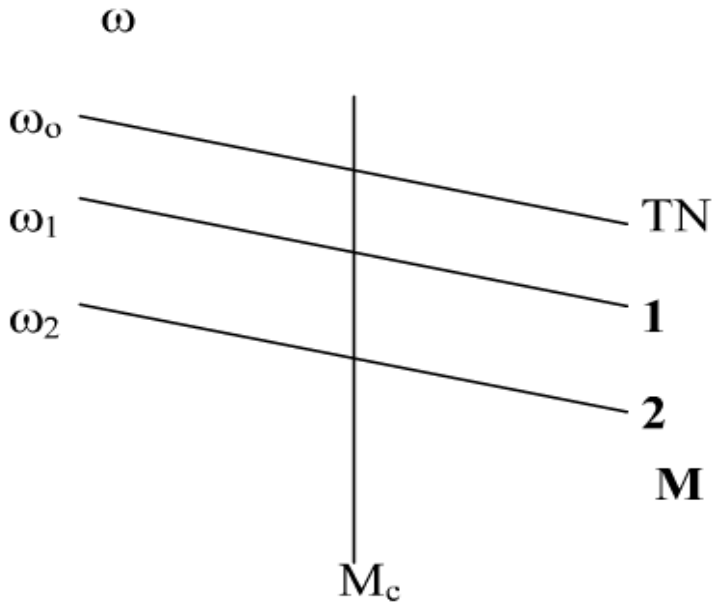
Có hai phương pháp điều khiển tốc độ động cơ điện một chiều bằng điện áp:

- Điều chỉnh điện áp cấp cho mạch phần ứng của động cơ.
- Điều chỉnh điện áp cấp cho mạch kích từ của động cơ.

Trong đó thông thường sử dụng phương pháp điều chỉnh điện áp cấp cho mạch phần ứng. Khi thay đổi điện áp phần ứng thì tốc độ động cơ điện thay đổi theo phương trình sau:

$$\omega = \frac{U_{ư} - I_{ư}.R_{ư}}{k.\phi}$$

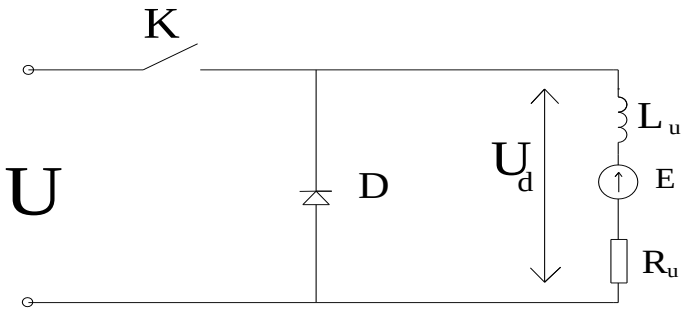
Vì từ thông của động cơ không đổi nên độ dốc đặc tính cơ cũng không đổi, còn tốc độ không tải lí tưởng thì tùy thuộc vào giá trị điện áp điều khiển $U_{ư}$ của hệ thống, do đó có thể nói phương pháp điều khiển này là triệt để. Đặc tính thu được khi điều khiển là một họ đường song song:



Hình 2.14: Đặc tính cơ của motor DC khi thay đổi điện áp phần ứng.

Nguyên lí điều khiển: Dùng phương pháp điều chế độ rộng xung PWM để thay đổi điện áp động cơ.

- Mạch nguyên lí:



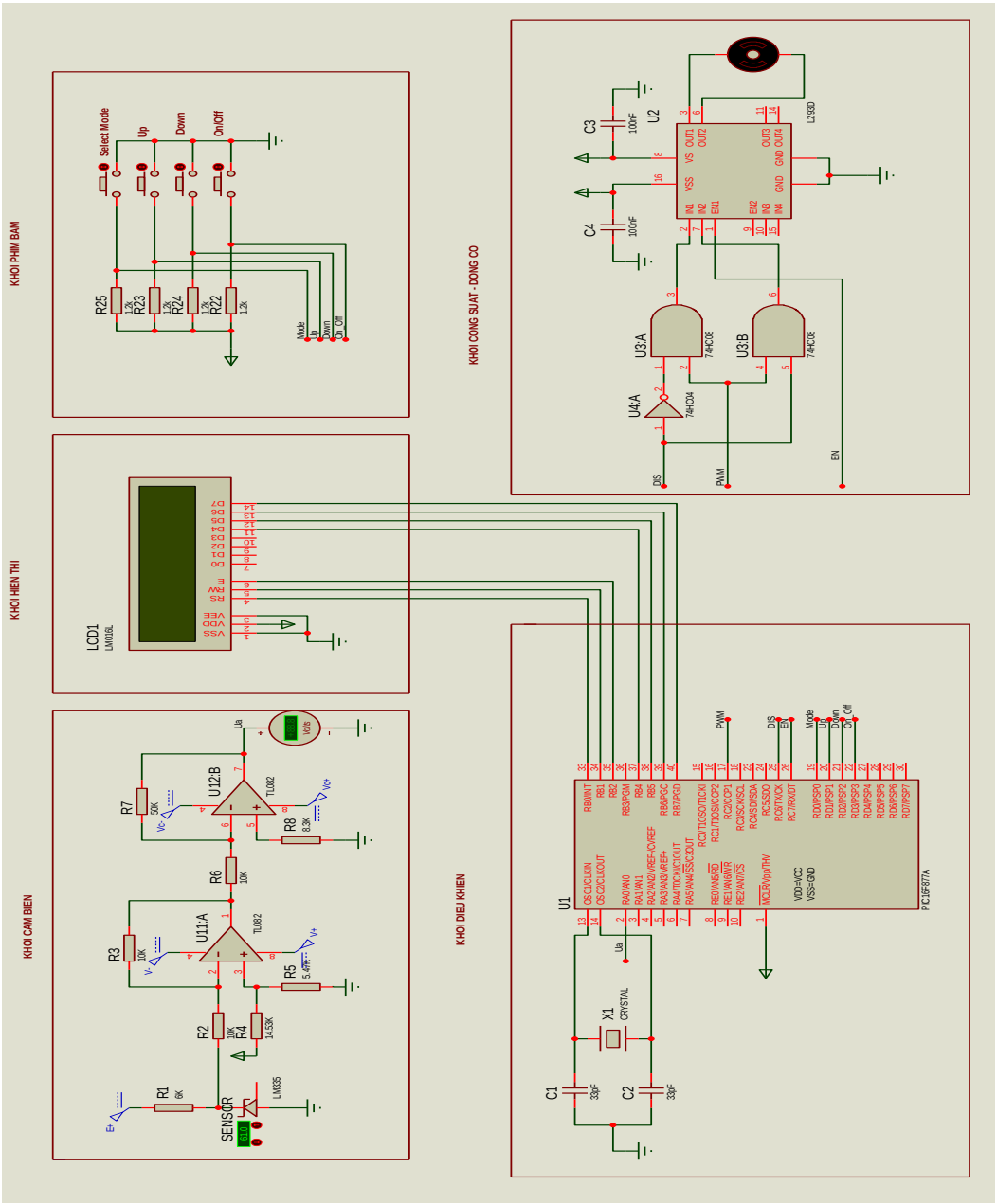
Hình 2.15: Mạch nguyên lí phương pháp điều chế độ rộng xung.

Khi K=1 (đóng): có dòng điện.

Khi K=0 (mở): không có dòng điện.

2.2.8. Sơ đồ mạch nguyên lý hệ thống

Sơ đồ mạch nguyên lý hệ thống như hình 2.12.

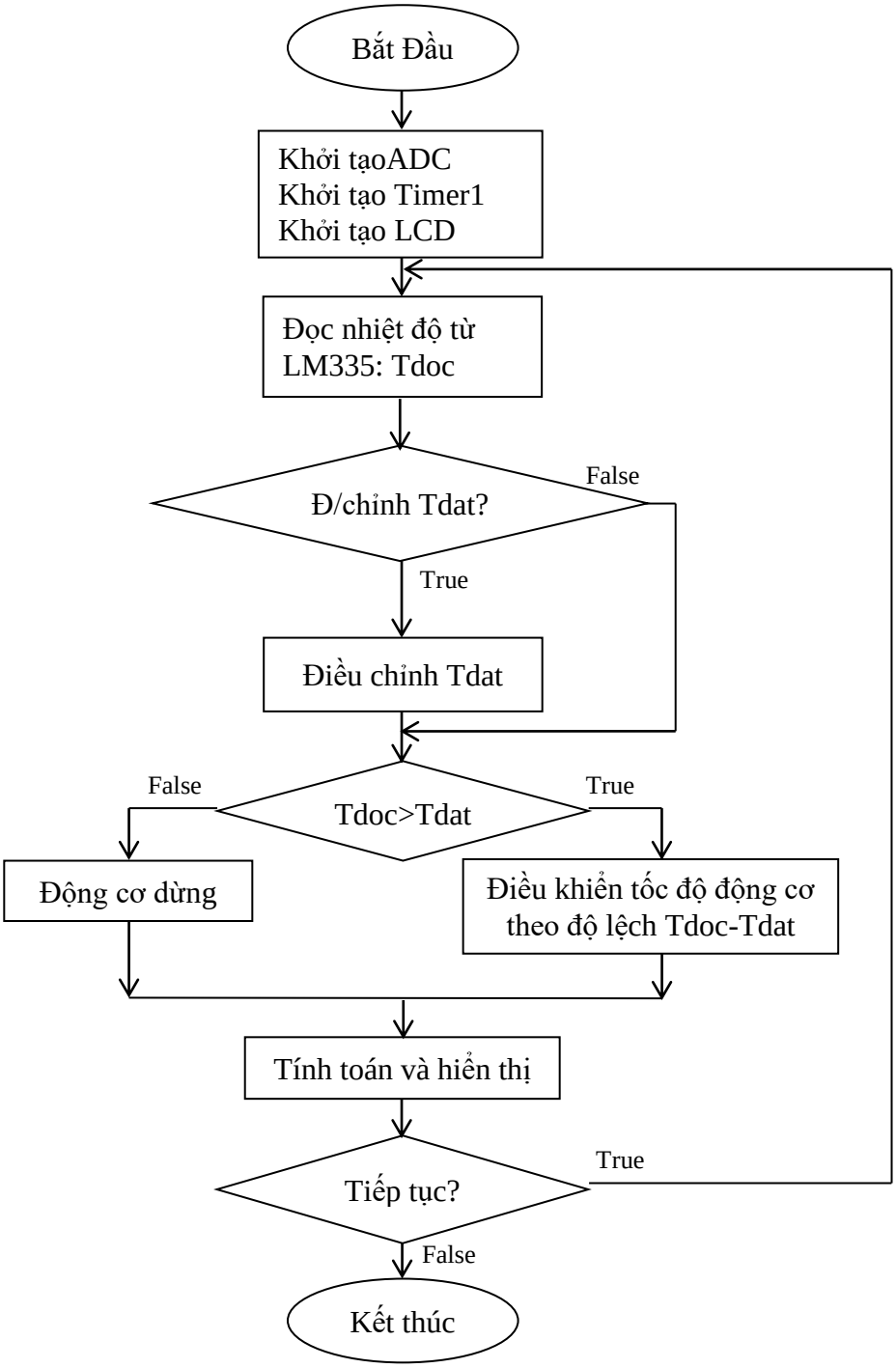


Hình 2.15: Sơ đồ mạch nguyên lý hệ thống.

CHƯƠNG 3.

CHƯƠNG TRÌNH ĐIỀU KHIỂN

3.1. LƯU ĐỒ THUẬT TOÁN



Hình 3.1: Lưu đồ thuật toán.

- Bước 1:
    + Khởi tạo ADC.
    + Khởi tạo Timer1.
    + Khởi tạo LCD.
- Bước 2: Đọc nhiệt độ từ LM335: Tdoc.
- Bước 3: Điều chỉnh Tdat.
    + Nếu không cần điều chỉnh Tdat, bỏ qua bước 4 đi đến bước 5.
    + Nếu cần điều chỉnh Tdat, đi đến bước 4.
- Bước 4: Điều chỉnh Tdat.
- Bước 5: So sánh Tdoc > Tdat:
    + False: dừng động cơ, đi đến bước 6.
    + True: Điều khiển tốc độ theo độ lệch nhiệt độ Tdoc-Tdat, đi đến bước 6.
- Bước 6: Tính toán và hiển thị nhiệt đo được, độ rộng xung PWM.
- Bước 7: Tiếp tục chương trình.

+ Nếu sai, kết thúc chương trình.

+ Đúng thì quay lại bước 2 thực hiện lặp chương trình, thực hiện vòng lặp while1.

3.2. CHƯƠNG TRÌNH ĐIỀU KHIỂN

Dưới đây là chương trình điều khiển động cơ theo nhiệt độ:

```
#include <16F877A.h>
#include <def_877a.h>
#device *=16 ADC=8
#include <LCD4bit.h>

#FUSES NOWDT                    //No Watch Dog Timer
#FUSES HS                       //High speed Osc (> 4mhz for PCM/PCH)
(>10mhz for PCD)
#FUSES NOPUT                    //No Power Up Timer
```

```
#FUSES NOPROTECT             //Code not protected from reading
#FUSES NODEBUG               //No Debug mode for ICD
#FUSES NOBROWNOUT            //No brownout reset
#FUSES NOLVP                 //No low voltage prgming, B3(PIC16) or
B5(PIC18) used for I/O
#FUSES NOCPD                 //No EE protection
#FUSES NOWRT                 //Program memory not write protected
#FUSES RESERVED              //Used to set the reserved FUSE bits

#use delay(clock=20000000)

int8 Duty,Mode,Tdat; // Khai bao cac bien dung trong ct
int16 Read_T, dT;
int1 On_Off,KeyPress,ReadTStatus;
int8 Count=0;
/*****************************************************/

void ReadKB() // Ham phat hien phim bam
{
   KeyPress=0;
   if(INPUT(PIN_D0)==0) // Neu phim 1 duoc an (phim chon mode)
     {
       delay_ms(150);           // Ham delay chong rung phim
       while(INPUT(PIN_D0)==0);   // Cho den khi nut duoc tha ra
           Mode++;  // Tang mode len 1 de chuyen sang mode ke tiep
       if(Mode==4) Mode=0;  // Lap lai mode 0 khi da chuyen qua het cac
mode
       KeyPress=1;
```

```
    }
  else
    {
      if(INPUT(PIN_D1)==0) // Neu phim 2 duoc an (phim tang)
            {
             delay_ms(150);       // Chong rung phim (cho phim duoc an trong
1 luc)
             if(INPUT(PIN_D1)==0)
                if(Mode==1) { if(Tdat<100) Tdat++; KeyPress=1; }      // Giam
nguong tuy theo mode
            }
      else if(INPUT(PIN_D2)==0) // Neu phim 3 duoc an (phim giam)
            {
             delay_ms(150);       // Cho phim duoc an trong 1 luc
             if(INPUT(PIN_D2)==0)
                if(Mode==2) { if(Tdat>0) Tdat--; KeyPress=1; }
            }
      else if(INPUT(PIN_D3)==0) // Neu phim 4 duoc an (phim Start)
            {
             delay_ms(150);
             if(INPUT(PIN_D3)==0); // Cho den khi nut duoc tha ra
                if(Mode==3)
                  {                  // thi nut 4 co chuc nang bat/tat
                    On_Off=~On_Off;
                    if(On_Off==1) Duty=100;
                    if(On_Off==0) Duty=0;
                    KeyPress=1;
                  }
```

```
            }
      }
}
/************************************************/

void Check()  // Ham kiem tra nguong
{
  if (Read_T<Tdat) {Duty=0; On_Off=0;}
  else {dT=Read_T-Tdat; On_Off=1;
        if(Read_T<60) {output_bit(PIN_C6,0);

        if(dT<=2) Duty=40;
        else if(dT<=4) Duty=55;
        else if(dT<=6) Duty=70;
        else if(dT<=8) Duty=85;
        else  Duty=100;}

        else{output_bit(PIN_C6,1);
        if(dT<=2) Duty=40;
        else if(dT<=4) Duty=55;
        else if(dT<=6) Duty=70;
        else if(dT<=8) Duty=85;
        else  Duty=100;}

       }
}
/************************************************/
```

```
void HienNhietDo() // Ham hien thi nhiet do do duoc tu LM335
{
  int8 T,chuc,donvi;

 LCD_Cmd(0x80);
 LCD_Char(" NHIET DO:  "); // Hien thi dong tren

 T = Read_T;
 chuc = T/10 + 0x30; // Tach so hang chuc va hang don vi
 donvi = T%10 + 0x30;

 LCD_Cmd(0x8B);
 LCD_Char(chuc);
 LCD_Char(donvi);
 LCD_Char("*C");

 LCD_Cmd(0xC0);
 LCD_Char(" CONG SUAT: "); // Hien thi dong tren
 if(Duty>=100)
   {
    LCD_Cmd(0xCB);
    LCD_Char("100%");
   }
 else
  {
    chuc = Duty/10 + 0x30; // Tach so hang chuc va hang don vi
    donvi = Duty%10 + 0x30;
    LCD_Cmd(0xCC);
```

```
      LCD_Char(chuc);
      LCD_Char(donvi);
      LCD_Char("% ");
    }
delay_ms(50);
}
/*******************************************************/

void SetTDown()  // Ham cai dat nguong nhiet do min
{
int8 T,chuc,donvi;

LCD_Cmd(0x80);
LCD_Char(" DAT GIAM NHIET  ");

T = Tdat;        // Tach hang chuc va don vi cua tmin va hien thi ra LCD
chuc = T/10 + 48;
donvi = T%10 + 48;

LCD_Cmd(0xC0);
LCD_Char("NHIET DO: ");
LCD_Char(chuc);
LCD_Char(donvi);
LCD_Char("*C ");

delay_ms(50);
}
/*******************************************************/
```

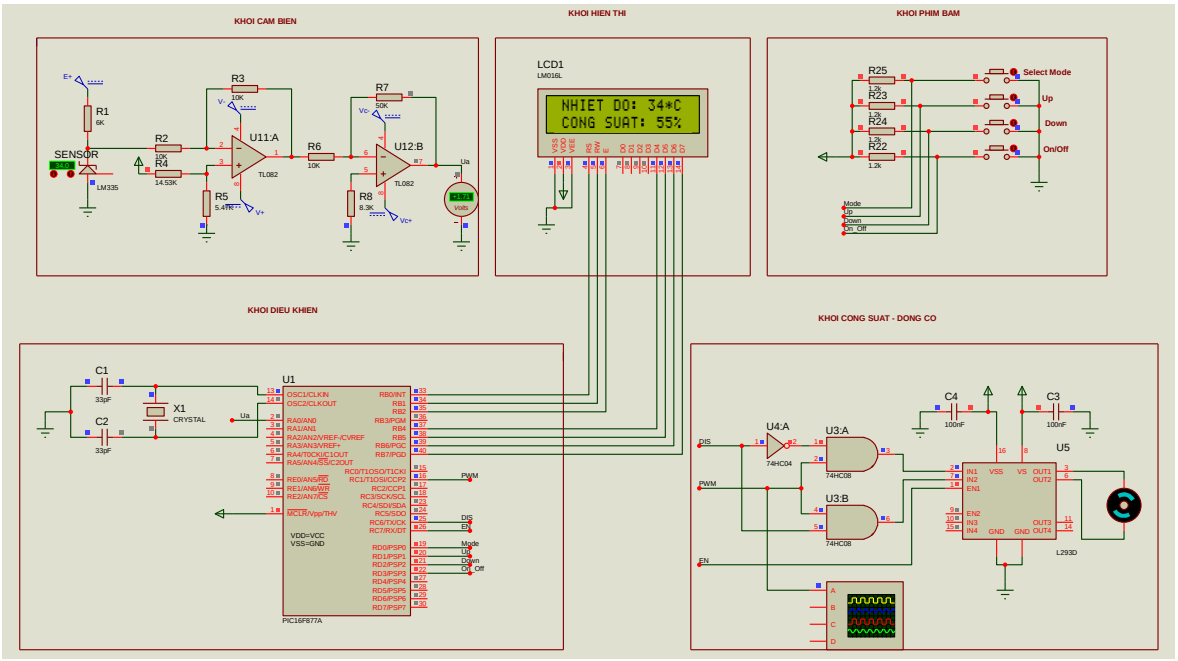
```
void SetTUp() // Ham cai dat nguong nhiet do max
{
int8 t,chuc,donvi;

LCD_Cmd(0x80);
LCD_Char(" DAT TANG NHIET  ");

t = Tdat;
chuc = t/10 + 48;
donvi = t%10 + 48;

LCD_Cmd(0xC0);
LCD_Char("NHIET DO: ");
LCD_Char(chuc);
LCD_Char(donvi);
LCD_Char("*C ");

delay_ms(50);
}
/*********************************************************/

void SelectMode()  // Ham lua chon mode hien thi
{
 switch (Mode)
   {
     case 0: HienNhietDo(); break;
     case 1: SetTUp(); break;
```

```
    case 2: SetTDown(); break;
    case 3: LCD_Cmd(0x80); LCD_Char(" CHE DO ON/OFF  ");
          if(On_Off==1) {LCD_Cmd(0xC0); LCD_Char("    ...ON...     ");}
          else {LCD_Cmd(0xC0); LCD_Char("    ...OFF...     ");}
  }
}

#INT_TIMER1  // Khai bao su dung ngat timer1
/*********************************************************/

void Timer1_Func()
{
CLEAR_INTERRUPT(INT_TIMER1); // Xoa co ngat timer1
DISABLE_INTERRUPTS(GLOBAL);  // Khong cho phep ngat khac
Count++;           // Sau moi ngat ta tang bien dem

if(Count==10)      // khi cnt=10 thi da dem duoc 10 lan tuong ung voi 1s
  {
   ReadTStatus=1;
   Read_T = READ_ADC();
   Read_T = (int16)(Read_T*100.0/255);
   set_pwm2_duty(Duty*256);
   switch (Duty)
     {
       case 0: set_pwm2_duty(0); break;
       case 40: set_pwm2_duty(102); break;
       case 55: set_pwm2_duty(140); break;
       case 70: set_pwm2_duty(178); break;
```

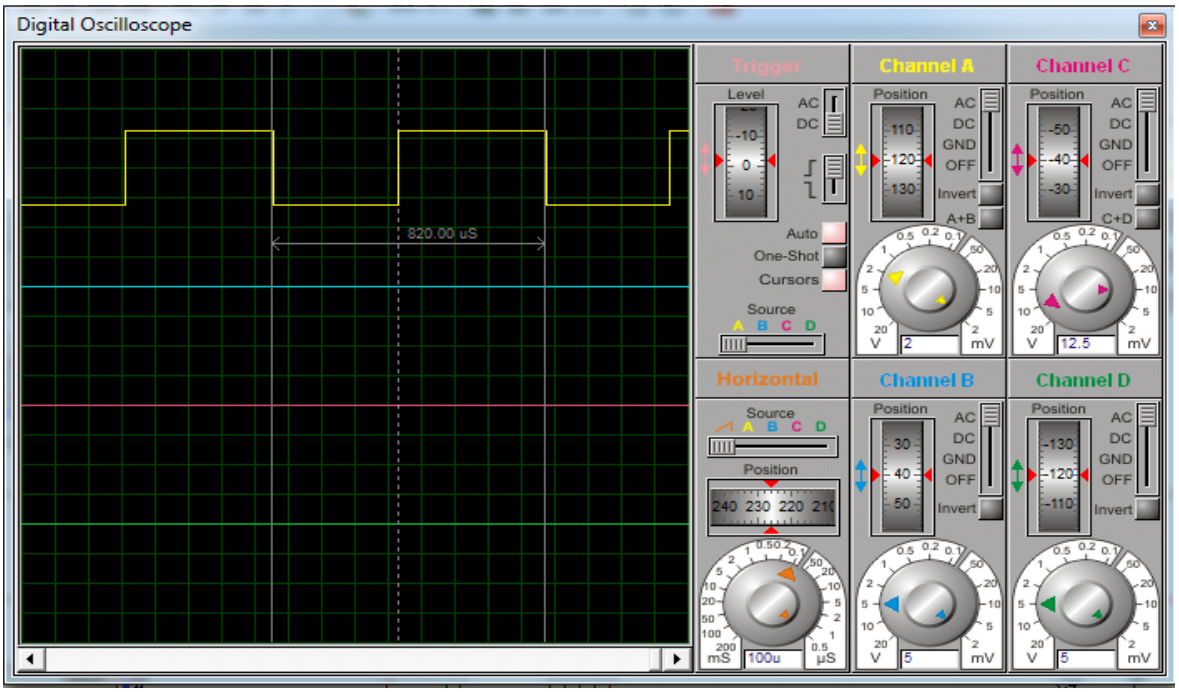
```
        case 85: set_pwm2_duty(217); break;
        case 100: set_pwm2_duty(255); break;
      }
    Count=0;
  }
else ReadTStatus=0;
SET_TIMER1(3035);
ENABLE_INTERRUPTS(GLOBAL);
}

/*************************************************************/
void main() // Ham ct chinh
{
Setup_ADC(ADC_CLOCK_INTERNAL); // Cai dat bo chuyen doi ADC
setup_adc_ports(AN0); // Ngo vao Analog la chan AN0
SET_ADC_CHANNEL(0);   // Chon kenh chuyen doi ADC la kenh 0
delay_us(20);

Mode = 0;  // Khoi tao ban dau
Tdat=30; //

lcd_init();  // Khoi tao LCD

SETUP_TIMER_1(T1_INTERNAL|T1_DIV_BY_8); // Cai dat timer1 voi bo chia tan 1:8
SETUP_TIMER_2(T2_DIV_BY_16,255,1);

SET_TIMER1(3035);   // Gia tri khoi tao cho timer1 ( dem den 65535
```

```
                // voi clock 20Mhz chia 8 => mot ngat tuong ung voi 0.1s )
ENABLE_INTERRUPTS(INT_TIMER1);  // Cho phep ngat timer1
ENABLE_INTERRUPTS(GLOBAL);  // Cho phep ngat toan cuc
setup_ccp2(CCP_PWM);

output_bit(PIN_C6,0);
output_bit(PIN_C7,1);

Read_T = READ_ADC();  // Doc gia tri nhiet do dau tien
Read_T = (int16)(Read_T*100.0/255);
while(1)  // Vong lap vo han
{
  SelectMode();
  do
    {
       if((ReadTStatus==1)&&(Mode==0)) // Neu nhiet do duoc doc thi hien
thi ra LCD neu dang
                    // o mode hien thi va tien hanh kiem tra nguong
         { HienNhietDo();
           Check();   // Kiem tra nguong
         }
//!          if(Read_T<60) {output_bit(PIN_C6,0);}
//!          else {output_bit(PIN_C6,1);}
       ReadKB();
    }
  while(KeyPress==0);  // Lien tuc lap lai viec kiem tra xem nhiet do
}                 // co duoc doc hay ko va kiem tra phim bam
}
```

- Kết quả mô phỏng thu được độ rộng xung thay đổi theo nhiệt độ bằng phương pháp PWM.



Hình 3.2: Sơ đồ nguyên lí khi điều khiển.



Hình 3.3: Độ rộng xung .

KẾT LUẬN

Sau 3 tháng tìm tòi và thực hiện đồ án tốt nghiệp với đề tài "***Thiết kế và xây dựng hệ thống điều khiển động cơ theo nhiệt độ***" đã giúp em nắm vững hơn về cách phân tích một công việc thiết kế, cách đặt vấn đề cho bài toán thiết kế. Giúp em có cách xử lí xác thực hơn và biết cách kết hợp với những kiến thức đã được học để tính toán tối ưu cho thiết kế. Không những thế, đề tài còn giúp em nắm được các kiến thức về:

- Vi điều khiển, cụ thể là PIC16F877A.

- Các kiến thức về sensor nhiệt, vi mạch thuật toán, hiển thị LCD, điều khiển motor DC theo phương pháp PWM và lập trình điều khiển trên PICC.

- Cách thức thiết kế hệ thống. Thực hiện mô phỏng hệ thống trên phần mềm thiết kế mạch Proteus

Để em có thể thực hiện được đề tài trong thời gian 3 tháng vừa qua không thể thiếu được sự hướng dẫn nhiệt tình, tỉ mỉ của các thầy cô trong khoa Điện - Điện tử, đặc biệt là Thạc sĩ Nguyễn Đoàn Phong. Em xin chân thành cám ơn các thầy cô.

Sinh viên thực hiện

Kiều Công Hòa

TÀI LIỆU THAM KHẢO

1. Nguyễn Mạnh Giang (2009), ***"Các vi điều khiển PIC"***, Nhà xuất bản Khoa học kĩ thuật.

2. Nguyễn Mạnh Giang (2007), ***"Cấu trúc, lập trình ghép nối và ứng dụng của Vi điều khiển"***, nhà xuất bản Lao Động - Xã Hội.

3. Phạm Minh Hà (2004), ***"Kỹ thuật mạch điện tử, Nhà xuất bản khoa học và kĩ thuật."***

4. Ngô Diên Tập, "***Vi xử lí trong đo lường và điều khiển***", Nhà xuất bản Khoa Học và Kĩ thuật, Hà Nội.

5. Bùi Đức Hùng (2008), ***"Máy điện"***, Nhà xuất bản giáo dục.

6. Các trang web tham khảo:

www.dientuvietnam.net

www.picvietnam.com

www.dientuvienthong.net

www.vagam.dieukhien.net

www.duyphi.phpnet.us/index.htm