

LỜI MỞ ĐẦU

Trong các nghiên cứu khoa học , trong công nghiệp và đời sống hàng ngày việc đo nhiệt độ là điều rất cần thiết. Tuy nhiên, để đo được trị số chính xác của nhiệt độ lại là vấn đề không đơn giản. Ngày nay với sự phát triển của công nghệ thông tin, công nghệ bán dẫn, sự ra đời của các cảm biến, vi điều khiển đã tạo điều kiện thuận lợi cho việc đo nhiệt độ một cách chính xác hơn , trên cơ sở đó em xin trình bày đề tài: **“Nghiên cứu thiết kế mạch đo nhiệt độ dùng LM35 và gửi tín hiệu nhiệt độ lên internet”**

Nội dung bản đồ án gồm 3 chương:

Chương 1: Tổng quan về đo nhiệt độ và cảm biến đo nhiệt độ

Chương 2: Thiết kế mạch đo nhiệt độ

Chương 3: Gửi tín hiệu đo lên internet

Em xin chân thành cảm ơn Th.s Trần Thị Phương Thảo đã hướng dẫn em trong suốt quá trình xây dựng và hoàn thành bản đồ án này!

Do thời gian thực hiện ngắn và kiến thức bản thân còn hạn chế do vậy đồ án không tránh khỏi những thiếu sót. Em rất mong nhận được sự đóng góp của các thầy cô.

Hải phòng ngày 25 tháng 11 năm 2012

Sinh viên

Nguyễn Văn Hiệp

CHƯƠNG 1.

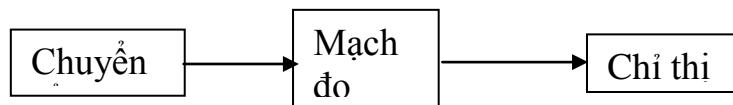
TỔNG QUAN VỀ ĐO NHIỆT ĐỘ VÀ CẢM BIẾN ĐO NHIỆT ĐỘ

1.1.TỔNG QUAN VỀ HỆ THỐNG ĐO LƯỜNG

1.1.1.Giới thiệu

Để thực hiện phép đo của một đại lượng nào đó thì tùy thuộc vào đặc tính của đại lượng cần đo,điều kiện đo,cũng như độ chính xác theo yêu cầu của một phép đo mà ta có thể thực hiện đo bằng nhiều cách khác nhau trên cơ sở của các hệ thống đo lường khác nhau.

Sơ đồ khối của một hệ thống đo lường tổng quát



Hình 1.1.Sơ đồ khối tổng quát

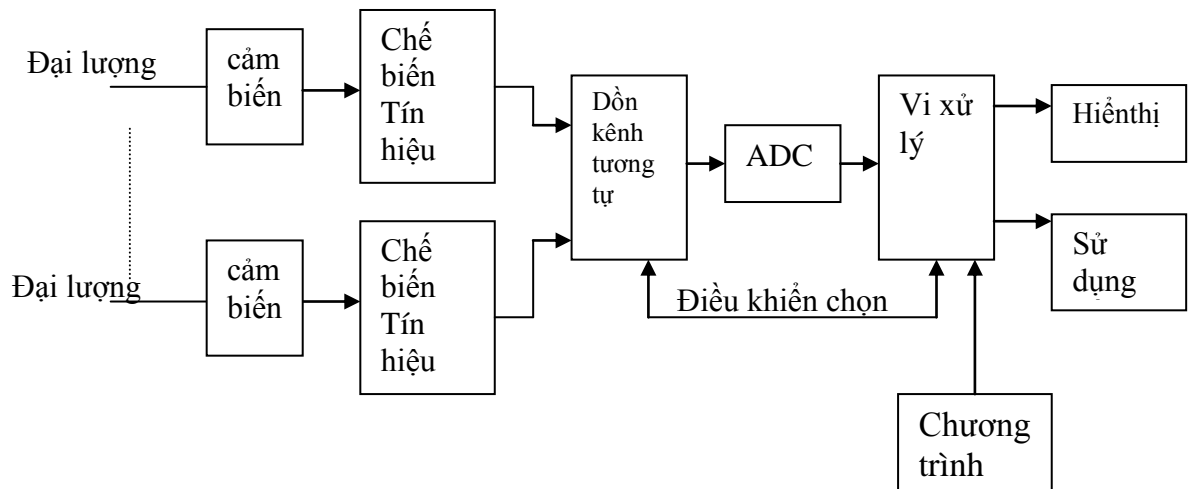
-Khối chuyển đổi: làm nhiệm vụ nhận trực tiếp các đại lượng vật lý đặc trưng cho đối tượng cần đo biến đổi các đại lượng thành các đại lượng vật lý thống nhất(dòng điện hay điện áp) để thuận lợi cho việc tính toán.

-Mạch đo: có nhiệm vụ tính toán biến đổi tín hiệu nhận được từ bộ chuyển đổi sao cho phù hợp với yêu cầu thể hiện kết quả đo của bộ chỉ thị.

-Khối chỉ thị:làm nhiệm vụ biến đổi tín hiệu điện nhận được từ mạch đo để thể hiện kết quả đo.

Hệ thống đo lường số

Hệ thống đo lường số được áp dụng để thực hiện đề tài này vì có các ưu điểm:các tín hiệu tương tự qua biến đổi thành các tín hiệu số có các xung rõ ràng ở trạng thái 0,1 sẽ giới hạn được nhiều mức tín hiệu gây sai số .Mặt khác ,hệ thống này tương thích với dữ liệu của máy tính,qua giao tiếp với máy tính ứng dụng rộng rãi trong kỹ thuật.



Hình 1.2. Sơ đồ khối của hệ thống đo lường số

1.1.2. Nguyên lý hoạt động

Đối tượng cần đo là đại lượng vật lý, dựa vào các đặc tính của đối tượng cần đo mà ta chọn một loại cảm biến phù hợp để biến đổi thông số đại lượng vật lý cần đo thành đại lượng điện, đưa vào mạch chế biến tín hiệu (gồm: bộ cảm biến, hệ thống khuếch đại, xử lý tín hiệu).

Bộ chuyển đổi tín hiệu sang số ADC (Analog Digital Converter) làm nhiệm vụ chuyển đổi tín hiệu tương tự sang tín hiệu số và kết nối với vi xử lý.

Bộ vi xử lý có nhiệm vụ thực hiện những phép tính và xuất ra những lệnh trên cơ sở trình tự những lệnh chấp hành đã thực hiện trước đó.

Bộ dồn kênh tương tự (multiplexers) và bộ chuyển ADC được dùng chung tất cả các kênh. Dữ liệu nhập vào vi xử lý sẽ có tín hiệu chọn đúng kênh cần xử lý để đưa vào bộ chuyển đổi ADC và đọc đúng giá trị đặc trưng của nó qua tính toán để có kết quả của đại lượng cần đo.

1.2. TỔNG QUAN VỀ ĐO NHIỆT ĐỘ

1.2.1. Khái niệm chung

Trong nghiên cứu khoa học, trong sản xuất cũng như trong đời sống sinh hoạt hằng ngày, luôn luôn cần xác định nhiệt độ của môi trường hay của một vật nào đó. Vì vậy việc đo nhiệt độ đã trở thành một việc làm vô cùng cần thiết. Đo nhiệt độ là một trong những phương thức đo lường không điện.

Nhiệt độ cần đo có thể rất thấp (một vài độ Kelvin), cũng có thể rất cao (vài ngàn, vài chục ngàn độ Kelvin). Độ chính xác của nhiệt độ có khi cần tới một vài phần ngàn độ, nhưng có khi vài chục độ cũng có thể chấp nhận được. Việc đo nhiệt độ được tiến hành nhờ các dụng cụ hỗ trợ chuyên biệt như cặp nhiệt điện, nhiệt điện trở, diode và transistor, IC cảm biến nhiệt độ, cảm biến thạch anh ... Tùy theo khoảng nhiệt độ cần đo và sai số cho phép mà người ta lựa chọn các loại cảm biến và phương pháp đo cho phù hợp.

1.2.2.Các phương pháp đo nhiệt độ

Tùy theo nhiệt độ đo có thể dùng các phương pháp khác nhau. Thông thường nhiệt độ đo được chia thành ba dải: Nhiệt độ thấp, nhiệt độ trung bình, nhiệt độ cao. Ở nhiệt độ trung bình và thấp phương pháp đo là phương pháp tiếp xúc nghĩa là các cảm biến được đặt trực tiếp ở ngay môi trường cần đo. Đối với nhiệt độ cao đo bằng phương pháp không tiếp xúc, dụng cụ đặt ở ngoài môi trường đo.

a)Đo nhiệt độ bằng phương pháp tiếp xúc

Phương pháp đo nhiệt độ trong công nghiệp thường là các nhiệt kế tiếp xúc. Có hai loại là: nhiệt kế nhiệt điện trở và nhiệt kế nhiệt ngẫu. Cấu tạo của nhiệt kế nhiệt điện trở và cặp nhiệt ngẫu cũng như cách lắp ghép chúng phải đảm bảo tính chất trao đổi nhiệt tốt giữa chuyển đổi với môi trường đo. Đối với môi trường khí hoặc nước, chuyển đổi được đặt theo hướng ngược lại với dòng chảy. Với vật rắn khi đặt nhiệt kế sát vào vật, nhiệt lượng sẽ truyền từ vật sang chuyển đổi và sẽ gây tổn hao nhiệt, nhất là với vật dẫn nhiệt kém. Do vậy diện tích tiếp xúc giữa vật đo và nhiệt kế càng lớn càng tốt. Khi đo nhiệt độ của các chất hạt (cát, đất...), cần phải cắm sâu nhiệt kế vào môi trường cần đo và thường dùng nhiệt kế nhiệt điện trở có cặp nối ra ngoài.

b) Đo nhiệt độ cao bằng phương pháp tiếp xúc

Ở môi trường nhiệt độ cao từ 1600°C trở lên, các cặp nhiệt ngẫu không chịu được lâu dài, vì vậy để đo nhiệt độ ở các môi trường đó người ta dựa trên hiện tượng quá trình quá độ đốt nóng cặp nhiệt :

$$\theta = f(t) = \Delta T(1 - e^{-t/\tau})$$

(1.1)

Trong đó : θ - là lượng tăng nhiệt độ của đầu nóng trong thời gian t

ΔT - hiệu nhiệt độ của môi trường đo và cặp nhiệt

τ - hằng số thời gian của cặp nhiệt ngẫu

Dựa trên quan hệ này có thể xác định được nhiệt độ của đối tượng đo mà không cần nhiệt độ đầu làm việc của cặp nhiệt ngẫu phải đạt đến nhiệt độ ấy. Nhúng nhiệt ngẫu vào môi trường cần đo trong khoảng (0,4 – 0,6)s, ta sẽ được phần đầu của đặc tính quá trình quá độ của nhiệt ngẫu và theo đó tính được nhiệt độ của môi trường. Nếu nhiệt độ đầu công tác của cặp nhiệt ngẫu trong thời gian nhúng vào môi trường cần đo đạt nhiệt độ vào khoảng một nửa nhiệt độ môi trường thì nhiệt độ tính được có sai số không quá hai lần sai số của nhiệt kế nhiệt nhẫu đo trực tiếp. Phương pháp này dùng để đo nhiệt độ của thép nấu chảy.

c) Đo nhiệt độ bằng phương pháp không tiếp xúc

Đây là phương pháp dựa trên định luật bức xạ của vật đen tuyệt đối, tức là vật hấp thụ năng lượng theo mọi hướng với khả năng lớn nhất. Bức xạ nhiệt của mọi vật thể đặc trưng bằng mật độ phổ E_λ nghĩa là số năng lượng bức xạ trong một đơn vị thời gian với một đơn vị diện tích của vật xảy ra trên một đơn vị của độ dài sóng. Quan hệ giữa mật độ phổ bức xạ của vật đen tuyệt đối với nhiệt độ và độ dài sóng được biểu diễn bằng công thức :

$$E_\lambda^0 = C_1 \lambda^{-5} (e^{c^2/(\lambda T)} - 1)^{-1} \quad (1.2)$$

Trong đó : C_1, C_2 – hằng số, λ - độ dài sóng, T – nhiệt độ tuyệt đối.
 $C_1=37,03.10^{-17} \text{ Jm}^2/\text{s}$; $C_2=1,432.10^{-2} \text{ m.độ}$.

d) Khoảng nhiệt độ đo

- Khoảng nhiệt độ đo bằng phương pháp tiếp xúc và dùng cặp nhiệt điện là từ 200°C đến 1000°C , độ chính xác có thể đạt tới $\pm 1\%$ $\rightarrow 0.1\%$.

- Khoảng nhiệt độ đo bằng phương pháp tiếp xúc và dùng cặp nhiệt điện (cặp nhiệt ngẫu) là từ -270°C đến 2500°C với độ chính xác có thể đạt tới $\pm 1\%$ $\rightarrow 0.1\%$.

- Khoảng nhiệt độ đo bằng phương pháp tiếp xúc và dùng các cảm biến tiếp giáp P-N (diode, transistor, IC) là từ -200°C đến 200°C , sai số đến $\pm 0.1\%$.

- Các phương pháp đo không tiếp xúc như bức xạ, quang phổ... có khoảng đo từ 1000°C đến vài chục ngàn độ C với sai số $\pm 1\%$ $\rightarrow 10\%$.

Thang đo nhiệt độ gồm: thang đo Celcius ($^{\circ}\text{C}$), thang đo Kelvin ($^{\circ}\text{K}$), thang đo Fahrenheit ($^{\circ}\text{F}$), thang đo Rankin ($^{\circ}\text{R}$).

$$T(^{\circ}\text{C}) = T(^{\circ}\text{K}) - 273.15$$

$$T(^{\circ}\text{F}) = T(^{\circ}\text{R}) - 459.67$$

$$T(^{\circ}\text{C}) = [T(^{\circ}\text{F}) - 32] * 5/9$$

$$T(^{\circ}\text{F}) = T(^{\circ}\text{C}) * 9/5 + 32$$

Bang 1.1. Sự liên hệ giữa các thang đo ở những nhiệt độ quan trọng:

Kelvin ($^{\circ}\text{K}$)	Celcius ($^{\circ}\text{C}$)	Rankin ($^{\circ}\text{R}$)	Fahrenheit ($^{\circ}\text{F}$)
0	-273.15	0	-459.67
273.15	0	491.67	32
273.16	0.01	491.69	32.018
373.15	100	671.67	212

1.2.3. Quá trình đo nhiệt độ

Ta có thể chia quá trình đo nhiệt độ ra làm ba khâu chính:

a) Khâu chuyển đổi

Khâu chuyển đổi nhiệt độ thường dựa vào những biến đổi mang tính đặc trưng của vật liệu khi chịu sự tác động của nhiệt độ. Có các tính chất đặc trưng sau đây:

- Sự biến đổi điện trở.
- Sức điện động sinh ra do sự chênh lệch nhiệt độ ở các mối nối của các kim loại khác nhau.
- Sự biến đổi thể tích, áp suất.
- Sự thay đổi cường độ bức xạ của vật thể khi bị đốt nóng.

Đối với chuyển đổi nhiệt điện, người ta thường dựa vào hai tính chất đầu tiên để chế tạo ra các cặp nhiệt điện (Thermocouple), nhiệt điện trở kim loại hay bán dẫn, các cảm biến nhiệt độ dưới dạng các linh kiện bán dẫn như: diode, transistor, các IC chuyên dùng.

b) Khâu xử lý

Các thông số về điện sau khi được chuyển đổi từ nhiệt độ sẽ được xử lý trước khi qua đến phần chỉ thị. Các bộ phận ở khâu xử lý gồm có: phần hiệu chỉnh, khuếch đại, biến đổi ADC (Analog-Digital-Converter)... Ngoài ra còn có thể có các mạch điện bổ sung như: mạch bù sai số, mạch phối hợp tổng trở...

c) Khâu chỉ thị

Khâu chỉ thị trước đây thường sử dụng các cơ cấu cơ điện, ở đó kết quả đo được thể hiện bằng góc quay hoặc sự di chuyển thẳng của kim chỉ thị. Ngày nay, với sự phát triển của công nghệ điện tử, đã sản xuất nhiều loại IC giải mã, IC số chuyên dùng trong biến đổi ADC, vì vậy cho phép ta sử dụng khâu chỉ thị số dễ dàng như dùng LED 7 đoạn hoặc màn hình tinh thể lỏng LCD. Ở đó, kết quả đo được thể hiện bằng các con số trong hệ thập phân.

1.3. CÁC LOẠI CẢM BIẾN ĐO NHIỆT ĐỘ

1.3.1. Nhiệt điện trở

Nhiệt điện trở thường dùng để đo nhiệt độ của hơi nước, khí than trong các đường ống, các lò phản ứng hóa học, các nồi hơi, không khí trong phòng ...

Nguyên lý làm việc của thiết bị này là dựa vào sự thay đổi điện trở theo nhiệt độ của các vật dẫn điện, tức là điện trở là một hàm theo nhiệt độ: $R = f(T)$. Cuộn dây điện trở thường nằm trong ống bảo vệ, tùy theo công dụng mà vỏ ngoài có thể làm bằng thủy tinh, kim loại hoặc gốm.

Đối với hầu hết các vật liệu dẫn điện thì giá trị điện trở R tùy thuộc vào nhiệt độ T theo một hàm tổng quát sau:

$$R(T) = R_0 \cdot F(T - T_0) \quad (1.3)$$

Với : R_0 :điện trở ở nhiệt độ T_0

F : hàm phụ thuộc vào đặc tính của vật liệu

$$F = 1 \text{ khi } T = T_0$$

-Đối với điện trở kim loại :

$$R(T) = R_0(1 + AT + BT^2 + CT^3) \quad (1.4)$$

T : tính bằng $^{\circ}\text{C}$

$$T_0 = 0^{\circ}\text{C}$$

-Đối với nhiệt điện trở bằng oxyt bán dẫn :

$$R(T) = R_0 \cdot \exp[B(1/T - 1/T_0)] \quad (1.5)$$

T : nhiệt độ tuyệt đối ($^{\circ}\text{K}$)

$$T_0 = 273.15^{\circ}\text{K}$$

Những hệ số trong công thức tính điện trở R thường được biết trước một cách chính xác nhờ sự đo những nhiệt độ đã biết.

a)Nhiệt điện trở kim loại

Đối với nhiệt điện trở kim loại thì việc chế tạo nó thích hợp hơn cả là sử dụng các kim loại nguyên chất như: platin, đồng, niken. Để tăng độ nhạy cảm nên sử dụng các kim loại có hệ số nhiệt điện trở càng lớn càng tốt. Tuy nhiên tùy thuộc vào khoảng nhiệt độ cần kiểm tra mà ta có thể sử dụng nhiệt điện trở loại này hay khác. Cụ thể: nhiệt điện trở chế tạo từ dây dẫn bằng đồng thường làm việc trong khoảng nhiệt độ từ $-50^{\circ} \div +150^{\circ}\text{C}$ với hệ số nhiệt điện trở $\alpha = 4,27 \cdot 10^{-3}$; Nhiệt điện trở từ dây dẫn platin mảnh làm việc trong

khoảng nhiệt độ $-190^{\circ}\div +650^{\circ}\text{C}$ với $\alpha=3,968.10^{-3} \left[\frac{1}{^{\circ}\text{C}} \right]$; Nhưng khi làm việc ngắn hạn, cũng như khi đặt điện trở nhiệt trong chân không hoặc khí trung tính thì nhiệt độ làm việc lớn nhất của nó có thể còn cao hơn.

Cấu trúc của nhiệt điện trở kim loại bao gồm: dây dẫn mảnh kép đôi quấn trên khung cách điện tạo thành phần tử nhạy cảm, nó được đặt trong chiếc vỏ đặc biệt có các cực đưa ra. Giá trị điện trở nhiệt được chế tạo từ $10\div 100\Omega$.

Đối với nhiệt điện trở kim loại thì quan hệ giữa điện trở với nhiệt độ có dạng sau:

$$R(\theta) = R_0(1 + \alpha.\theta + \beta.\theta^2 + \gamma.\theta^3 + \dots) \quad (1.6)$$

Trong đó : R_0 -điện trở dây dẫn ứng với nhiệt độ ban đầu 0°C .

R_0 -điện trở dây dẫn ứng với nhiệt độ θ .

θ -nhiệt độ [$^{\circ}\text{C}$]

α, β, γ -các hệ số nhiệt điện trở = const. $\left[\frac{1}{^{\circ}\text{C}} \right]$

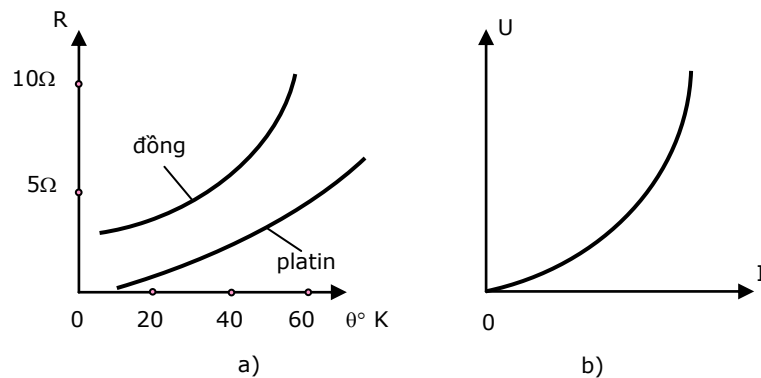
Để thấy rõ hơn nữa về bản chất của nhiệt điện trở kim loại, chúng ta có thể xem qua điện trở suất của nó được tính theo công thức :

$$\rho = \frac{l}{n \cdot e \cdot \mu} \quad (1.7)$$

Trong đó: n -số điện tử tự do trong một đơn vị thể tích.

e -điện tích của điện tử tự do.

μ -tính linh hoạt của điện tử, được đặc trưng bởi tốc độ của nó trong trường có cường độ 1 vôn/cm.



Hình 1.3. Đặc tính nhiệt (a) và đặc tính vôn_ampe của nhiệt điện trở kim loại (b).

Các kim loại dùng làm điện trở nhiệt thường có điện trở suất nhỏ $\rho \approx 10^{-5} \div 10^{-6} \Omega/\text{cm}$, và có mật độ điện tử lớn (không phụ thuộc vào nhiệt độ). Khi nhiệt độ tăng ρ phụ thuộc vào sự dao động của mạng tinh thể kim loại, tức là nó được xác định bởi tính linh hoạt của các điện tử. Như vậy khi có sự thay đổi nhiệt độ thì cũng làm cho tính linh hoạt của các điện tử thay đổi theo. Tuy nhiên tính linh hoạt của các điện tử còn phụ thuộc vào mật độ tạp chất trong kim loại. Cụ thể điện trở suất của kim loại nguyên chất có thể xác định theo dạng: $\rho = \rho_0 + \rho(\theta)$, trong đó ρ_0 không phụ thuộc vào nhiệt độ; còn $\rho(\theta)$ là một hàm phụ thuộc không cố định: ứng với nhiệt độ trong khoảng nào đó thì nó là tuyến tính $\rho(\theta) = K \cdot \theta$, nhưng ứng với nhiệt độ rất thấp ($\approx 0^\circ\text{C}$) thì quan hệ đó là hàm bậc năm của nhiệt độ. Trên hình 1.3.a biểu diễn mối quan hệ giữa điện trở và nhiệt độ, hình 1.3.b là dạng đặc tính vôn-ampe của nhiệt điện trở kim loại.

Độ nhạy cảm của nhiệt điện trở được xác định theo biểu thức:

$$S = \frac{dR}{d\theta} \approx \frac{\Delta R}{\Delta \theta} \quad (1.5)$$

Trong đó: ΔR - sự thay đổi điện trở khi có sự thay đổi nhiệt độ $\Delta \theta$. Việc sử dụng nhiệt điện trở kim loại để đo nhiệt độ cao rất tin cậy, đảm bảo độ

chính xác cao đến $0,001^{\circ}\text{C}$ và sai số đo không quá 0,5 đến 1%; Tuy nhiên khi đó dòng tải qua nó có giá trị không lớn lắm. Nếu như có dòng điện lớn luôn chạy qua nhiệt điện trở, thì sự quá nhiệt của nó sẽ lớn hơn rất nhiều so với môi trường xung quanh. Khi đó độ quá nhiệt xác lập sẽ được xác định bởi điều kiện truyền nhiệt trên bề mặt của nhiệt điện trở (tốc độ chuyển động của môi trường cần kiểm tra so với nhiệt điện trở, và tỷ trọng của môi trường đó). Hiện tượng này được sử dụng để đo tốc độ thông lượng (dòng chảy) của chất lỏng và khí, cũng như để đo tỷ trọng của khí.... Bên cạnh ưu điểm trên thì bản thân nhiệt điện trở kim loại có những nhược điểm sau:

Thứ nhất nó là khâu phi chu kỳ được mô tả bằng phương trình vi phân bậc nhất đơn giản

$$(TP+1)R(t) = K\theta(t) \quad (1.6)$$

Trong đó hằng số thời gian T của nó có giá trị từ vài giây đến vài trăm giây. K chính là độ nhạy S.

Thứ hai rất cơ bản đó là kích thước của nhiệt điện trở kim loại lớn nên hạn chế việc sử dụng nó để đo nhiệt độ ở nơi hẹp.

Một số nhiệt điện trở kim loại thông dụng:

-Nhiệt điện trở Platin: Nhiệt điện trở Platin thường được chế tạo dưới dạng dây quấn đường kính (0.05 -> 0.1)mm, đo nhiệt độ từ -200°C -> 1000°C với độ chính xác tương đối cao, ngay cả trong những điều kiện môi trường dễ oxy hóa ($\alpha = 3,9.10^{-3}/^{\circ}\text{C}$).

Tuy nhiên, ở nhiệt độ xấp xỉ 1000°C hoặc cao hơn, Platin thường kém bền và chỉ thị nhiệt mất chính xác.

-Nhiệt điện trở Nickel:

Có ưu điểm là độ nhạy nhiệt rất cao ($6,66.10^{-3}/^{\circ}\text{C}$) từ 0°C đến 100°C , điện trở suất là $1,617.10^{-8}$ (còn của Platin là $1,385.10^{-8}$). Nickel chống lại sự oxy hóa, thường được dùng ở nhiệt độ nhỏ hơn 250°C .

-Nhiệt điện trở đồng:

Được sử dụng vì đặc tuyến rất tuyến tính của sự thay đổi điện trở theo nhiệt. Tuy nhiên vì phản ứng hóa học nên không cho phép sử dụng ở nhiệt độ lớn hơn 180°C, và vì điện trở suất bé nên khi dùng, để đảm bảo có giá trị điện trở nhất định, chiều dài dây phải lớn gây nên một sự cồng kềnh bất tiện.

-Nhiệt điện trở Tungstène:

Có độ nhạy nhiệt của điện trở lớn hơn của Platin trong trường hợp nhiệt độ cao và nó thường được sử dụng ở nhiệt độ cao hơn Platin với một độ tuyến tính hơn Platin. Tungstène có thể được cấu tạo dưới dạng những sợi rất mảnh cho phép thực hiện điện trở có giá trị cao, như vậy với giá trị điện trở cho trước, chiều dài dây sẽ giảm thiểu.

b) Nhiệt điện trở bán dẫn (Thermistor)

Nhiệt điện trở được chế tạo từ vật liệu bán dẫn được gọi là termistor; Chúng được sử dụng rộng rãi trong các hệ thống tự động kiểm tra và điều khiển. Termistor được chế tạo từ hợp kim của đồng - mangan hoặc coban - mangan dưới dạng thỏi, đĩa tròn hoặc hình cầu. Loại này hoàn toàn trái ngược với nhiệt điện trở kim loại: khi nhiệt độ tăng thì điện trở của nó lại giảm theo quy luật:

$$R(\theta) = R_0 \cdot e^{-\alpha\theta} = R_0 \left(1 - \alpha\theta + \frac{\alpha^2\theta^2}{2} - \dots \right) \quad (1.7)$$

Trong đó hệ số nhiệt điện trở của termistor thường có giá trị

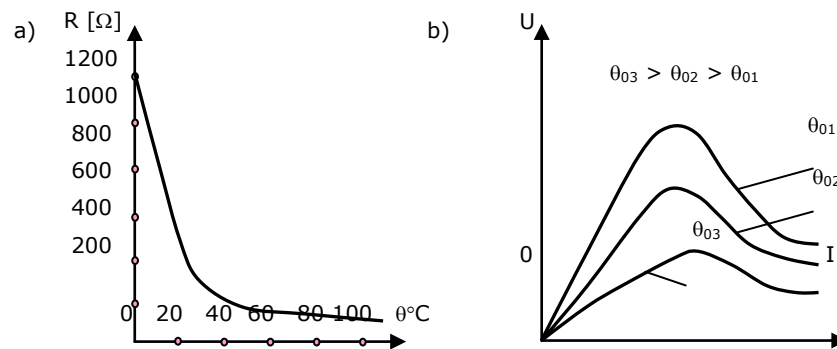
$$\alpha = (0,03 \div 0,06) \cdot \left[\frac{1}{\theta_c} \right]$$

Điện trở suất của termistor được tính theo công thức:

$$\rho = A \cdot e^{B/\theta} \quad (1.8)$$

Trong đó: A -hằng số phụ thuộc kích thước của termistor

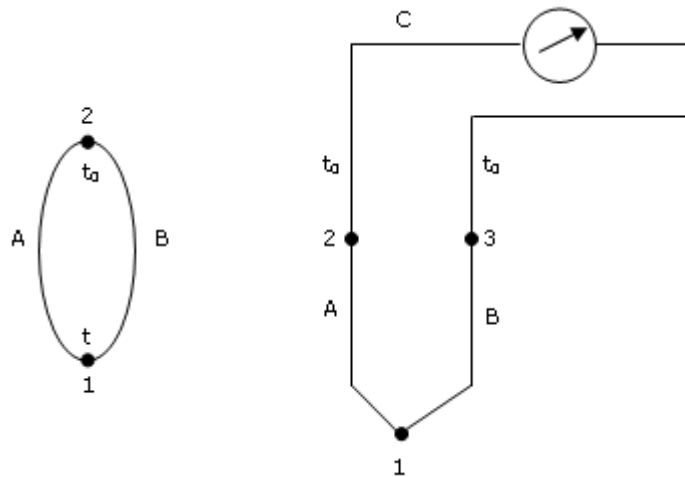
B -hằng số phụ thuộc tạp chất trong chất bán dẫn



Hình 1.4. Đặc tính nhiệt (a) và đặc tính vôn ampe(b)

Cũng như điện trở nhiệt kim loại, termistor cũng có hai đặc tính: Đặc tính nhiệt là quan hệ giữa điện trở của termistor với nhiệt độ (hình 1.2.a) và đặc tính vôn - ampe là quan hệ giữa điện áp đặt trên termistor với dòng điện chạy qua nó ứng với nhiệt độ nào đó θ_0 (hình 1.2.b). Chúng ta thấy rằng đặc tính vôn - ampe của termistor có giá trị cực đại của U ứng với I_1 nào đó, là do khi tăng dòng lớn hơn I_1 thì nó sẽ nung nóng termistor và làm cho giá trị điện trở của nó giảm xuống. Các loại termistor thường được chế tạo từ vài chục Ω đến vài chục $\text{K}\Omega$. Termistor có điện trở lớn cho phép đặt nó ở vị trí cần kiểm tra khá xa so với nơi bố trí hệ thống đo lường. Chúng có thể làm việc trong khoảng nhiệt độ từ -60°C đến $+180^{\circ}\text{C}$, và cho phép đo nhiệt độ với độ chính xác $0,0005^{\circ}\text{C}$. Để sử dụng termistor ở nhiệt độ lớn hơn, hoặc nhỏ hơn khoảng nhiệt độ làm việc bình thường thì người ta phải sử dụng đến các tổ hợp chất bán dẫn khác. So với điện trở nhiệt kim loại thì termistor có kích thước và trọng lượng nhỏ hơn, do đó cho phép chúng ta đặt nó ở những nơi chật hẹp để kiểm tra nhiệt độ của đối tượng nào đó.

1.3.2.Đo nhiệt độ bằng cặp nhiệt điện (Thermocouple)



Hình 1.5. Sơ đồ cặp nhiệt ngẫu và sơ đồ nối cặp nhiệt ngẫu

Bộ cảm biến cặp nhiệt ngẫu là một mạch có từ hai hay nhiều thanh dẫn điện gồm hai dây dẫn A và B. Chỗ nối giữa hai thanh kim loại này được hàn với nhau. Nếu nhiệt độ các mối hàn t và t_0 khác nhau thì trong mạch khép kín có một dòng điện chạy qua. Chiều của dòng nhiệt điện này phụ thuộc vào nhiệt độ tương ứng của mỗi hàn, nghĩa là $t > t_0$ thì dòng điện chạy theo hướng ngược lại. Nếu để hở một đầu thì giữa hai cực xuất hiện một sức điện động (sđđ) nhiệt. Như vậy bằng cách đo sđđ ta có thể tìm được nhiệt độ t của đối tượng đo với $t_0 = \text{const}$.

Cách đấu dụng cụ đo vào mạch bộ biến đổi nhiệt điện trên hình 1.4b.

Một số cặp nhiệt điện thông dụng:

-Thermocouple Platin_Rhodium Platin:

Nhiệt độ sử dụng : $T = -50^{\circ}\text{C} \rightarrow 1500^{\circ}\text{C}$

Đường kính dây : 0,51mm

Sức điện động Seebeck : $E = (-2,3 \rightarrow 16,7)\text{mV}$

Loại 10% Platin : $T = 0^{\circ}\text{C} \rightarrow 600^{\circ}\text{C}$, cấp chính xác là $\pm 2,5\%$

$T = 600^{\circ}\text{C} \rightarrow 1600^{\circ}\text{C}$, cấp chính xác là $\pm 0,4\%$

Loại 13% Platin : $T = 0^{\circ}\text{C} \rightarrow 538^{\circ}\text{C}$, cấp chính xác là $\pm 1,4\%$

$T = 538^{\circ}\text{C} \rightarrow 1500^{\circ}\text{C}$, cấp chính xác là $\pm 0,25\%$

Loại 30% Platin : $T = 0^{\circ}\text{C} \rightarrow 1700^{\circ}\text{C}$, cấp chính xác là $\pm 0,5\%$

-Thermocouple Wolfram-Rhenium:

Đường kính dây : 0,40mm

Sức điện động Seebeck : $E = (0 \rightarrow 38,5)\text{mV}$

Loại Wolfram_Rhenium 5% : $T = 0^{\circ}\text{C} \rightarrow 2760^{\circ}\text{C}$

Loại Wolfram_Rhenium 26% : $T = 0^{\circ}\text{C} \rightarrow 1950^{\circ}\text{C}$

Chuyên dùng để đo nhiệt độ rất cao.

-Thermocouple Chromel_Alumel:

Nhiệt độ sử dụng : $T = -270^{\circ}\text{C} \rightarrow 1250^{\circ}\text{C}$

Đường kính dây : 3,25mm

Sức điện động Seebeck : $E = (-5,35 \rightarrow 50)\text{mV}$

Cấp chính xác : $T = 0^{\circ}\text{C} \rightarrow 400^{\circ}\text{C}$ là $\pm 3\%$

$T = 400^{\circ}\text{C} \rightarrow 1250^{\circ}\text{C}$ là $\pm 0,75\%$

-Thermocouple Chromel_Constantan:

Nhiệt độ sử dụng : $T = -270^{\circ}\text{C} \rightarrow 870^{\circ}\text{C}$

Đường kính dây : 3,25mm

Sức điện động Seebeck : $E = (-9,8 \rightarrow 66)\text{mV}$

Cấp chính xác : $T = 0^{\circ}\text{C} \rightarrow 400^{\circ}\text{C}$ là $\pm 3\%$

$T = 400^{\circ}\text{C} \rightarrow 870^{\circ}\text{C}$ là $\pm 0,75\%$

-Thermocouple Fer_Constantan :

Nhiệt độ sử dụng : $T = -210^{\circ}\text{C} \rightarrow 800^{\circ}\text{C}$

Đường kính dây : 3,25mm

Sức điện động Seebeck : $E = (-8 \rightarrow 45)\text{mV}$

Cấp chính xác : $T = 0^{\circ}\text{C} \rightarrow 400^{\circ}\text{C}$ là $\pm 3\%$

$T = 400^{\circ}\text{C} \rightarrow 1250^{\circ}\text{C}$ là $\pm 0,75\%$

-Thermocouple Cu_Constantan :

Nhiệt độ sử dụng : $T = -270^{\circ}\text{C} \rightarrow 370^{\circ}\text{C}$

Đường kính dây : 1,63mm

Sức điện động Seebeck : $E = (-6,25 \rightarrow 19)\text{mV}$

Cấp chính xác : $T = -100^{\circ}\text{C} \rightarrow -40^{\circ}\text{C}$ là $\pm 2\%$

$T = -40^{\circ}\text{C} \rightarrow 100^{\circ}\text{C}$ là $\pm 8\%$

$T = 100^{\circ}\text{C} \rightarrow 350^{\circ}\text{C}$ là $\pm 0,75\%$

1.3.3.Đo nhiệt độ bằng hỏa kế quang học :

Hoả kế quang học là tên gọi chung của các dụng cụ đo nhiệt độ bằng cách ứng dụng các tính chất của hệ thống thấu kính quang học để thu lấy các bức xạ của vật thể rồi căn cứ theo độ bức xạ của vật thể để xác định nhiệt độ .

a)Nguyên lý cơ bản

Nguyên lý làm việc của hỏa kế quang học là dựa trên các hiện tượng bức xạ của các vật thể ở các nhiệt độ cao, trong đó có liên quan đến vai trò của vật đen tuyệt đối. Đó là một thực thể vật chất có khả năng hấp thụ hoàn toàn tất cả các bức xạ nhận được mà không phóng xạ.

b)Một số dạng của hỏa kế quang học thông dụng

Hiện nay, trong công nghiệp, người ta dùng rất nhiều loại hỏa kế quang học như hỏa kế bức xạ, hỏa kế vi sai, hỏa kế đo màu sắc, hỏa kế nhiệt ngẫu...

Nếu hỏa kế tiêu thụ toàn bộ năng lượng của bức xạ toàn phần của vật thể, đó là hỏa kế bức xạ toàn phần.

Hoả kế quang điện dùng sự so sánh giữa sự phát sáng của dây tóc ngọn đèn được chế tạo đặc biệt với độ sáng của vật nung nóng và xác định chính xác dây tóc và nhiệt độ.

Hỏa kế quang điện cho kết quả đo không phụ thuộc vào người quan sát và có thể nối liên mạch với các thiết bị không chế nhiệt độ tự động.

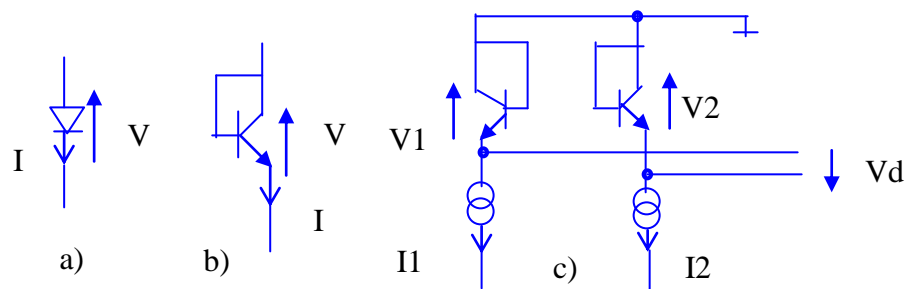
c)Phạm vi sử dụng

Phạm vi sử dụng là nhiệt độ của vật cần đo không dưới 800°C . Tất cả các loại hỏa kế quang học đều có sai số không vượt quá 1%. Tuy nhiên, bảng

chỉ nhiệt trên các hỏa kế chỉ hoàn toàn chính xác với vật đen tuyệt đối (quy ước có bức xạ bằng 1). Vì vậy, với giá trị thật của nhiệt độ các vật cần đo phụ thuộc vào mức độ đen của từng chất phát sáng. Hỏa kế quang điện là dụng cụ đo nhiệt độ gián tiếp nên có nhiều thuận lợi, có thể đo từ xa mà không cần tiếp xúc với vật cần đo.

1.3.4. Đo nhiệt độ dùng diode và transistor

Những thành phần được sử dụng, diode hay transistor Silicium được mắc như diode (cực nền và cực thu nối chung) được cung cấp theo chiều thuận dòng điện I không đổi, điện áp V ở hai đầu cực của chúng, tùy thuộc vào nhiệt độ, điều này có thể xem như tín hiệu điện đi ra từ cảm biến tùy thuộc vào nhiệt độ.



Hình 1.6. Cảm biến đo nhiệt độ dùng diode và transistor

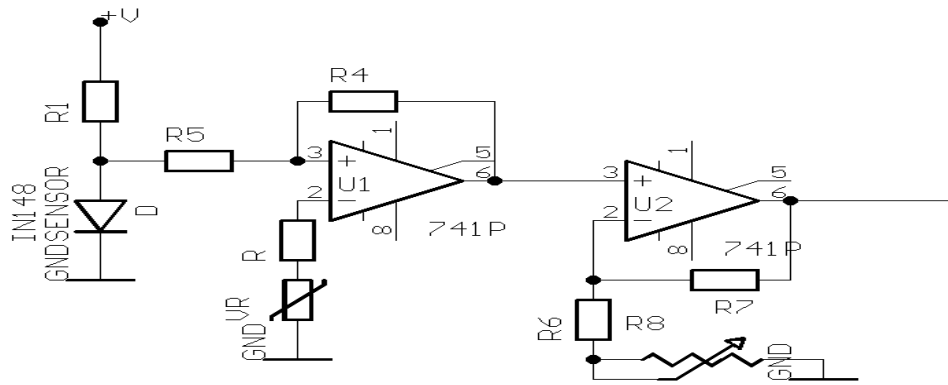
Các thành phần được sử dụng làm cảm biến đo nhiệt độ:

Hình 1.6.a) diode

Hình 1.6.b) Transistor mắc thành diode

Hình 1.6.c) Hai Transistor giống nhau được mắc như diode

Người ta lợi dụng sự thay đổi tuyến tính của mối nối p-n đối với nhiệt độ để chế tạo ra các diode và transistor chuyên dùng, làm cầu cảm biến nhiệt trong đo lường và khống chế nhiệt độ.



Hình 1.7. Sơ đồ mạch cảm biến dùng diode

Trong đó:

R1 : phân cực cho dòng chạy qua diode.

IC1 : dùng khuếch đại đảo, hệ số khuếch đại bằng 1, bù trừ điện áp DC của diode cảm biến D.

IC2 : khuếch đại không đảo, hệ số khuếch đại bằng 5.

Nguyên lý hoạt động được dùng tương tự như mạch dùng transistor cảm biến.

1.3.5. Đo nhiệt độ bằng IC

a) Giới thiệu:

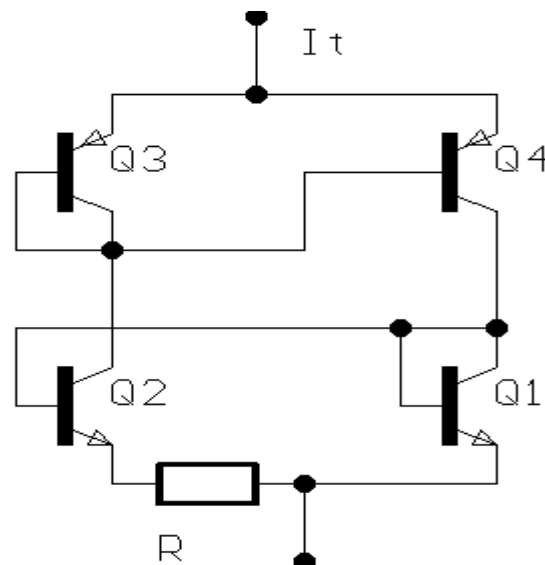
Kỹ thuật vi điện tử cho phép chế tạo được những mạch kết nối gồm những transistor giống nhau được sử dụng để làm cảm biến hoàn hảo đo nhiệt độ dựa vào việc đo sự khác biệt điện áp V_{BE} dưới tác động của nhiệt độ. Các cảm biến này tạo ra các dòng điện hoặc điện áp tỷ lệ với nhiệt độ tuyệt đối, với độ tuyến tính cao; nó có điều lợi là vận hành đơn giản, tuy nhiên phạm vi hoạt động giới hạn chỉ trong khoảng -50°C đến 150°C .

b) Nguyên lý chung của IC đo nhiệt độ:

Là mạch tích hợp nhận tín hiệu nhiệt độ chuyển thành tín hiệu dưới dạng điện áp hoặc tín hiệu dòng điện. Dựa vào đặc tính rất nhạy của các bán dẫn với nhiệt độ, tạo ra điện áp hoặc dòng điện tỷ lệ thuận với nhiệt độ tuyệt đối. Đo tín hiệu điện, ta biết được giá trị của nhiệt độ cần đo.

Sự tích cực của nhiệt độ tạo ra điện tích tự do và các lỗ trống trong chất bán dẫn bằng sự phá vỡ các phân tử, bứt các electron thành dạng tự do di chuyển qua các vùng cấu trúc mạng tinh thể, tạo sự xuất hiện các lỗ trống nhiệt làm cho tỷ lệ điện tử tự do và các lỗ trống tăng lên theo quy luật hàm mũ với nhiệt độ. Kết quả của hiện tượng này là dưới mức điện áp thuận, dòng thuận của mỗi nối p-n (trong diode hay transistor) sẽ tăng theo hàm mũ theo nhiệt độ.

Ví dụ khảo sát cảm biến IC AD590. Cảm biến này tạo ra một dòng điện thay đổi tuyến tính theo nhiệt độ tuyệt đối, nó được dùng đo nhiệt độ trong trường hợp dùng dây dẫn với khoảng cách xa.



Hình 1.8. Sơ đồ nguyên lý IC AD590

Các transistor Q3 và Q4 có cùng điện áp V_{BE} và có dòng cực phát giống nhau và bằng:

$$I_{E3} = I_{E4} = I_T / 2$$

Dòng điện này đi qua Q4 cũng chính là dòng điện cực phát của Q1 ,nó xác định điện áp nền-phát là:

$$V_{BE1} = (KT/q).log(I_T / 2I_0)$$

Với K : hằng số Boltzmann

T : $^{\circ}K$

q: điện tích

I_o : dòng điện nghịch (thông thường $I_o \ll I_T$) khi phân cực thuận.

Dòng điện $I_T/2$ đi qua Q3, qua Q2 có điện áp nền-phát là :

$$V_{BE2} = (KT/q).\log(I_T/16I_o)$$

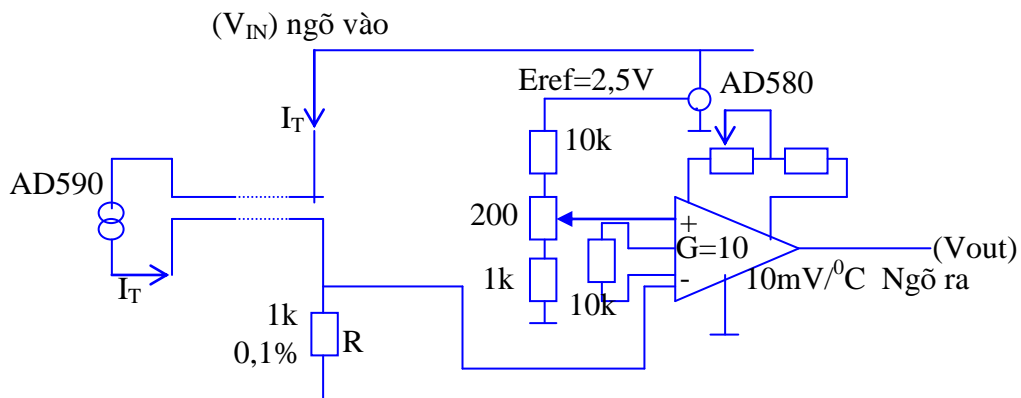
Thực tế Q2 gồm 8 transistor giống Q1, mỗi transistor có dòng điện $I_T/16$

Sự sai biệt điện áp giữa V_{BE1} và V_{BE2} , xuất hiện ở hai đầu điện trở R có dòng điện $I_T/2$ chạy qua là:

$$V_{BE1} - V_{BE2} = (KT/q).\log 8 = R.I_T/2$$

$$\Rightarrow I_T = (2/R).(KT/q).\log 8$$

Sơ đồ mạch đo nhiệt độ dùng IC AD590:



Hình 1.9. Sơ đồ nguyên lý đo nhiệt độ dùng IC AD590

Dòng điện I_T tạo nên ở hai đầu điện trở $R = 1K\Omega$ một điện áp có trị số bằng TmV (T là nhiệt độ tuyệt đối của cảm biến).

Nguồn điện áp chuẩn do IC AD580L có $E_{ref} = 2,5V$ và nhờ mạch phân áp tạo ra điện áp có giá trị khoảng $273,15mV$ với bộ khuếch đại có độ lợi $G = 10$, ở ngõ ra tín hiệu V_o tỉ lệ với nhiệt độ của cảm biến (theo $^\circ C$):

$$V_o = 10(T - 273,15)mV = 10(T^\circ C) \quad (mV)$$

Đặc tính một số IC đo nhiệt độ thông dụng:

-AD 590:

- Ngõ ra là dòng điện.
- Độ nhạy $1A/^{\circ}K$
- Độ chính xác : $+4C$
- Nguồn cung cấp : $V_{cc} = 4V \rightarrow 30V$
- Phạm vi sử dụng : $-55^{\circ}C \rightarrow 150^{\circ}C$

-LX5700:

- Ngõ ra là điện áp
- Độ nhạy : $-10mV/^{\circ}C$
- Độ chính xác : $3,8K$
- Độ tuyến tính : $\geq 1K$
- Phạm vi sử dụng : $-55^{\circ}C \rightarrow 150^{\circ}C$
- Loại này ít sử dụng vì độ chính xác thấp.

-LM135, LM235, LM335:

- Ngõ ra là điện áp.
- Độ nhạy : $10mV/^{\circ}C$
- Dòng làm việc : $400\mu A \rightarrow 500\mu A$: không thay đổi đặc

tính.

LM135 có sai số cực đại là $1,5^{\circ}C$ khi nhiệt độ lớn hơn $100^{\circ}C$.

- Phạm vi sử dụng:

LM335 : $-10^{\circ}C \rightarrow 125^{\circ}C$


LM235 : $-40^{\circ}C \rightarrow 140^{\circ}C$

LM135 : $-55^{\circ}C \rightarrow 200^{\circ}C$

LM35 : $-55^{\circ}C \rightarrow 150^{\circ}C$

1.3.6. Giới thiệu một số loại của các hãng khác nhau

Bảng 1.2. Máy đo nhiệt từ xa TI315E

	3000°C Máy đo nhiệt độ từ xa TI315E	
	Mã hàng	TI315E
	Phạm vi đo	500 ~ +3000°C
	Độ chính xác	±2% / ±2°C
	Độ hiển thị	±1C / ±1°F
	Tỉ lệ khoảng cách / tiết diện đo	120:1
Khối lượng	600g	

- Sử dụng tia hồng ngoại để xác định điểm đo, đo nhiệt độ những vật ở xa bằng phương pháp không tiếp xúc. Thiết kế nhỏ gọn, dễ sử dụng, dễ thao tác. Phạm vi đo rộng, đo chính xác. Có ống ngắm dùng cho các vật đo ở xa.

- Màn hình hiển thị LCD. Chức năng cài đặt giới hạn, tính toán và thống kê giá trị đo. Chuyển đổi đơn vị °C / °F.



Hình 1.10. Máy đo nhiệt độ siêu nhỏ

Khoảng đo: - Độ C: -40°C ~ + 200°C.- Độ F: -40°F ~ +392°F.

Độ chính xác: - ±1°C trong khoảng đo từ -20°C ~ +100°C (-4°F ~ +212°F); -
±2°C trong khoảng đo còn lại.

Độ chia: 0.1°C (0.1°F)

Pin: 1.5V

Bảng 1.3.Đồng hồ đo nhiệt độ



	Khoảng đo	Độ chia	Độ chính xác
°C	-50°C ~ 1300°C	0.1°	-50°C ~ 0°C: ±(0.5% + 1°C)
		C	0°C ~ 199.9°C: ± (0.3% + 1°C)
		1°C	-50°C ~ 300°C: ±(0.5% + 1°C) 301°C ~ 1000°C: ± (0.3% + 1°C) 1001°C ~ 1300°C: ±(0.5% + 1°C)
°F	-58°F ~ 1999°F	0.1°	-58°F ~ 199.9°F: ± (0.3% + 2°F)
		1°F	-58°F ~ 1999°F: ± (0.3% + 2°F)

Pin: 9V; Kích thước: 143 x 74 x 34mm.

Trọng lượng: 226g; Xuất xứ: Đài Loan.

CHƯƠNG 2.

THIẾT KẾ MẠCH ĐO NHIỆT ĐỘ

2.1. CÁC LINH KIỆN CHÍNH SỬ DỤNG TRONG MẠCH

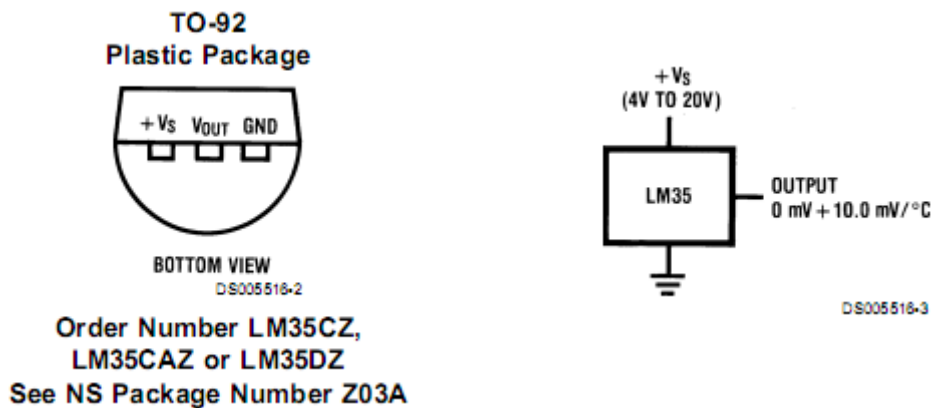
2.2.1. CẢM BIẾN ĐO NHIỆT ĐỘ LM35

a. Giới thiệu cảm biến đo nhiệt độ LM35

Trong bài này chúng ta sử dụng con LM35DZ

* Đặc điểm:

- Dải nhiệt độ biến đổi: 0 đến 100 độ
- Nhiệt độ ra thẳng thang đo Celcius nghĩa là ở 25 độ C điện áp ra là 0.25V
- Tương ứng 10mV/độ C
- Đảm bảo độ chính xác 0.5 độ C tại nhiệt độ 25 độ C.
- Làm việc với nguồn nuôi 4V đến 30V
- Trở kháng ra thấp 0.1 ohm với tải 1mA
- Khả năng tự làm nóng thấp, 0.08 độ C trong không khí.



Hình 2.1. Cảm biến nhiệt độ LM35DZ

Cách mắc: đơn giản là nối chân +Vs với nguồn và chân GND với đất chân OUTPUT nối với chân Vin+ của ADC0804

b. Dải nhiệt độ và sự thay đổi trở kháng theo nhiệt độ của LM35

Các bộ biến đổi (Transducer) chuyển đổi các đại lượng vật lý ví dụ như nhiệt độ, cường độ ánh sáng, lưu tốc và tốc độ thành các tín hiệu điện phụ thuộc vào bộ biến đổi mà đầu ra có thể là tín hiệu dạng điện áp, dòng, trở kháng hay dung kháng. Ví dụ, nhiệt độ được biến đổi thành về các tín hiệu điện sử dụng một bộ biến đổi gọi là Thermistor (bộ cảm biến nhiệt), một bộ cảm biến nhiệt đáp ứng sự thay đổi nhiệt độ bằng cách thay đổi trở kháng nhưng đáp ứng của nó không tuyến tính.

Bảng 2.1. Trở kháng của bộ cảm biến nhiệt theo nhiệt độ

Nhiệt độ ($^{\circ}\text{C}$)	Trở kháng của cảm biến ($\text{k}\Omega$)
0	29.490
25	10.000
50	3.893
75	1.700
100	0.817

Bảng 2.2. Hướng dẫn chọn loại các cảm biến nhiệt họ LM35

Mã sản phẩm	Dải nhiệt độ	Độ chính xác	Đầu ra
LM35A	-55 C to + 150 C	+ 1.0 C	10 mV/F
LM35	-55 C to + 150 C	+ 1.5 C	10 mV/F
LM35CA	-40 C to + 110 C	+ 1.0 C	10 mV/F
LM35C	-40 C to + 110 C	+ 1.5 C	10 mV/F
LM35D	0 C to + 100 C	+ 2.0 C	10 mV/F

Loại các bộ cảm biến LM35 là bộ cảm biến nhiệt mạch tích hợp chính xác cao mà điện áp đầu ra của nó tỷ lệ tuyến tính với nhiệt độ theo thang độ Celsius. Chúng cũng không yêu cầu cân chỉnh ngoài vì vốn chúng đã được cân chỉnh. Chúng đưa ra điện áp 10mV cho mỗi sự thay đổi 1°C . Bảng 2.2 hướng dẫn ta chọn các cảm biến họ LM35.

c. Nguyên lý đo và chuyển đổi tương tự/số của ADC

Khi nhiệt độ môi trường thay đổi ± 1 làm cho trở kháng của cảm biến LM35 thay đổi dẫn đến điện áp đầu vào V_{in} của ADC thay đổi. Điện áp V_{in} vào ADC sẽ được so sánh với U_d của ADC. U_d có thể thay đổi từ 0V đến $2(V_{ref}/2)$.

Ban đầu $U_d = 0$, nếu $V_{in} > U_d$ khi đó U_d sẽ được cộng thêm một giá trị là ΔU .
 $U_d = U_d + \Delta U$, trong đó :

$$\Delta U = \frac{2(V_{ref} / 2)}{256} = 10mV \quad (2.1)$$

đồng thời giá trị bộ đếm tăng thêm 1. Quá trình so sánh cứ như vậy đến khi nào $U_d = V_{in}$ thì dừng. Khi đó giá trị của bộ đếm chính là giá trị thập phân.

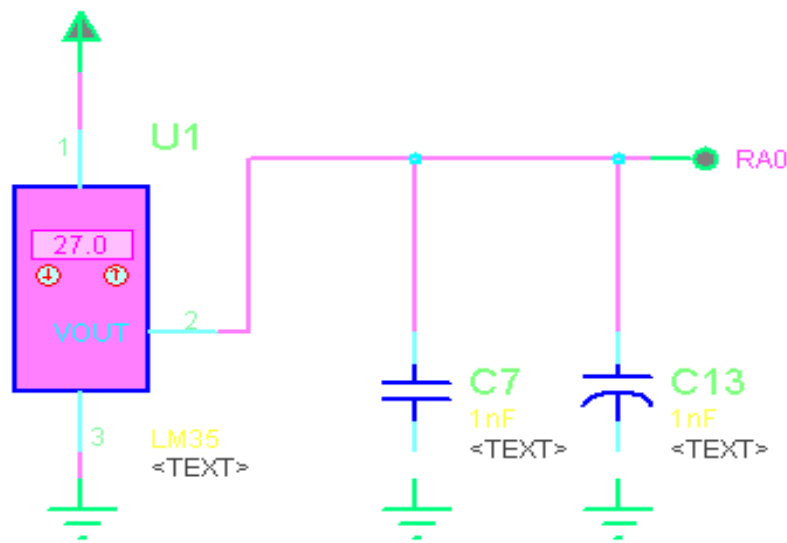
Khả năng tự làm nóng của LM35DZ trong không khí là $0,1^\circ\text{C}$.

Cảm biến LM35 có hệ số nhiệt là $10mV/^\circ\text{C}$, do đó sai số về nhiệt độ của LM35DZ sẽ là $0,5^\circ\text{C}$. Khi đó điện áp đầu vào V_{in} sẽ được làm tròn lên hoặc xuống. Khi V_{in} đi vào so sánh với điện áp U_d của ADC, ta nhận thấy rằng mức điện áp của ADC là 10mV nên sai số mà nó gây ra là 0,5mV tương ứng với giá trị nhiệt độ là $0,5^\circ\text{C}$.

Vậy sai số tổng cộng của hệ thống là $1,1^\circ\text{C}$.

Do vậy nhiệt độ thực tế của ta sẽ là : $t_{real} = t_{do} \pm 1,1^\circ\text{C}$.

Để tăng độ chính xác của phép đo lên ta có thể dùng một cảm biến khác có độ chính xác cao hơn, có thể giảm điện áp đưa vào chân $V_{ref/2}$ để giảm bước điện áp vi phân bậc thang của ADC, Tuy nhiên, khi bước điện áp của ADC và cảm biến không đồng nhất thì sẽ gây khó khăn cho quá trình xử lý dữ liệu đưa ra khâu hiển thị hoặc có thể gây ra sai số. Tùy thuộc vào phép hiển thị mà người ta có thể đặt giá trị điện áp cho chân $V_{ref/2}$ sao cho hợp lý.



Hình 2.1. Sơ đồ ghép nối LM35

2.2. GIỚI THIỆU VỀ HỌ VI ĐIỀU KHIỂN PIC

2.2.1. Tổng quan về họ vi điều khiển PIC

PIC là viết tắt của “Programmable Intelligent Computer”, có thể tạm dịch là “máy tính thông minh khả trình” do hãng General Instrument đặt tên cho vi điều khiển đầu tiên của họ:

PIC1650 được thiết kế để dùng làm các thiết bị ngoại vi cho vi điều khiển CP1600. Vi điều khiển này sau đó được nghiên cứu phát triển thêm và từ đó hình thành nên dòng vi điều khiển PIC ngày nay.

Hiện nay trên thị trường có rất nhiều họ vi điều khiển như 8051, Motorola 68HC, AVR, ARM,... Ngoài họ 8051 được hướng dẫn một cách căn bản ở môi trường đại học, bản thân người viết đã chọn họ vi điều khiển PIC để mở rộng vốn kiến thức và phát triển các ứng dụng trên công cụ này vì các nguyên nhân sau:

Họ vi điều khiển này có thể tìm mua dễ dàng tại thị trường Việt Nam.

Giá thành không quá đắt.

Có đầy đủ các tính năng của một vi điều khiển khi hoạt động độc lập.

Là một sự bổ sung rất tốt về kiến thức cũng như về ứng dụng cho họ vi điều khiển mang tính truyền thống: họ vi điều khiển 8051.

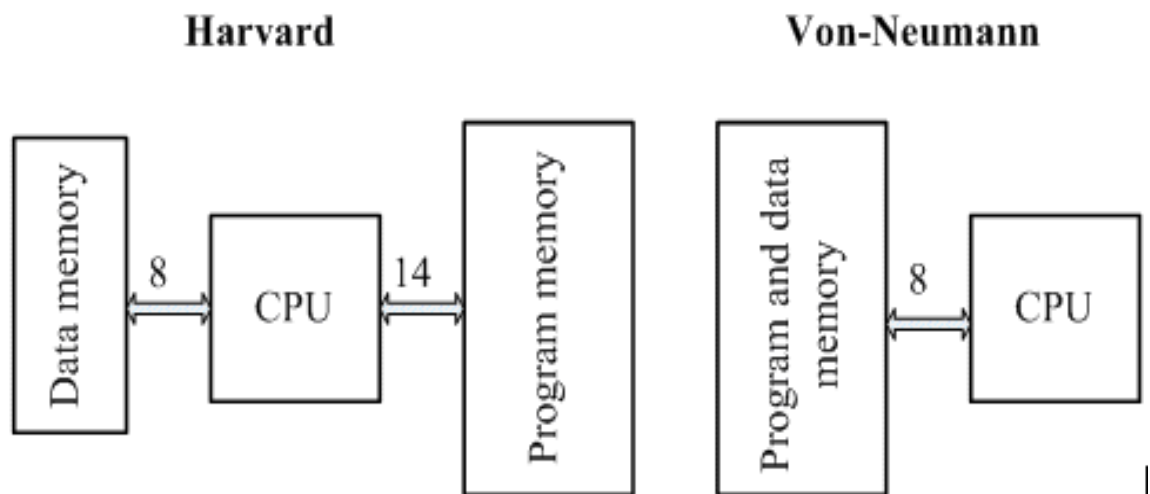
Số lượng người sử dụng họ vi điều khiển PIC. Hiện nay tại Việt Nam cũng như trên thế giới, họ vi điều khiển này được sử dụng khá rộng rãi. Điều này tạo nhiều thuận lợi trong quá trình tìm hiểu và phát triển các ứng dụng như: số lượng tài liệu, số lượng các ứng dụng mở đã được phát triển thành công, dễ dàng trao đổi, học tập, dễ dàng tìm được sự chỉ dẫn khi gặp khó khăn,...

Sự hỗ trợ của nhà sản xuất về trình biên dịch, các công cụ lập trình, nạp chương trình từ đơn giản đến phức tạp,...

Các tính năng đa dạng của vi điều khiển PIC, và các tính năng này không ngừng được phát triển.

2.2.2. Kiến trúc PIC

Cấu trúc phần cứng của một vi điều khiển được thiết kế theo hai dạng kiến trúc: kiến trúc Von Neuman và kiến trúc Harvard.



Hình 2.2. Kiến trúc Harvard và kiến trúc Von-Neuman

Tổ chức phần cứng của PIC được thiết kế theo kiến trúc Harvard. Điểm khác biệt giữa kiến trúc Harvard và kiến trúc Von-Neuman là cấu trúc bộ nhớ dữ liệu và bộ nhớ chương trình.

Đối với kiến trúc Von-Neuman, bộ nhớ dữ liệu và bộ nhớ chương trình nằm chung trong một bộ nhớ, do đó ta có thể tổ chức, cân đối một cách linh hoạt bộ nhớ chương trình và bộ nhớ dữ liệu. Tuy nhiên điều này chỉ có ý nghĩa khi tốc độ xử lý của CPU phải rất cao, vì với cấu trúc đó, trong cùng một thời điểm CPU chỉ có thể tương tác với bộ nhớ dữ liệu hoặc bộ nhớ chương trình. Như vậy có thể nói kiến trúc Von-Neuman không thích hợp với cấu trúc của một vi điều khiển.

Đối với kiến trúc Harvard, bộ nhớ dữ liệu và bộ nhớ chương trình tách ra thành hai bộ nhớ riêng biệt. Do đó trong cùng một thời điểm CPU có thể tương tác với cả hai bộ nhớ, như vậy tốc độ xử lý của vi điều khiển được cải thiện đáng kể.

Một điểm cần chú ý nữa là tập lệnh trong kiến trúc Harvard có thể được tối ưu tùy theo yêu cầu kiến trúc của vi điều khiển mà không phụ thuộc vào cấu trúc dữ liệu. Ví dụ, đối với vi điều khiển dòng 16F, độ dài lệnh luôn là 14 bit (trong khi dữ liệu được tổ chức thành từng byte, còn đối với kiến trúc Von-Neuman, độ dài lệnh luôn là bội số của 1 byte (do dữ liệu được tổ chức thành từng byte)). Đặc điểm này được minh họa cụ thể trong hình 2.2.

2.2.3.RISC và CISC

Như đã trình bày ở trên, kiến trúc Harvard là khái niệm mới hơn so với kiến trúc Von-Neuman. Khái niệm này được hình thành nhằm cải tiến tốc độ thực thi của một vi điều khiển.

Qua việc tách rời bộ nhớ chương trình và bộ nhớ dữ liệu, bus chương trình và bus dữ liệu, CPU có thể cùng một lúc truy xuất cả bộ nhớ chương trình và bộ nhớ dữ liệu, giúp tăng tốc độ xử lý của vi điều khiển lên gấp đôi. Đồng thời cấu trúc lệnh không còn phụ thuộc vào cấu trúc dữ liệu nữa mà có thể linh động điều chỉnh tùy theo khả năng và tốc độ của từng vi điều khiển. Và để tiếp tục cải tiến tốc độ thực thi lệnh, tập lệnh của họ vi điều khiển PIC được thiết kế sao cho chiều dài mã lệnh luôn cố định (ví dụ đối với họ

16Fxxxx chiều dài mã lệnh luôn là 14 bit) và cho phép thực thi lệnh trong một chu kỳ của xung clock (ngoại trừ một số trường hợp đặc biệt như lệnh nhảy, lệnh gọi chương trình con ... cần hai chu kỳ xung đồng hồ).

Điều này có nghĩa tập lệnh của vi điều khiển thuộc cấu trúc Harvard sẽ ít lệnh hơn, ngắn hơn, đơn giản hơn để đáp ứng yêu cầu mã hóa lệnh bằng một số lượng bit nhất định.

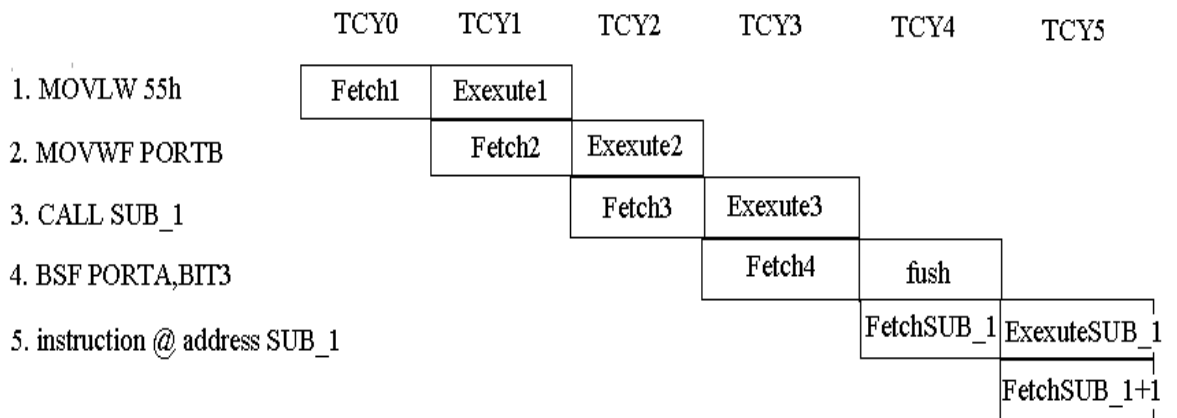
Vi điều khiển được tổ chức theo kiến trúc Harvard còn được gọi là vi điều khiển RISC (Reduced Instruction Set Computer) hay vi điều khiển có tập lệnh rút gọn. Vi điều khiển được thiết kế theo kiến trúc Von-Neuman còn được gọi là vi điều khiển CISC (Complex Instruction Set Computer) hay vi điều khiển có tập lệnh phức tạp vì mã lệnh của nó không phải là một số cố định mà luôn là bội số của 8 bit (1 byte).

2.2.4.Pipelining

Đây chính là cơ chế xử lý lệnh của các vi điều khiển PIC. Một chu kỳ lệnh của vi điều khiển sẽ bao gồm 4 xung clock. Ví dụ ta sử dụng oscillator có tần số 4 MHz, thì xung lệnh sẽ có tần số 1 MHz (chu kỳ lệnh sẽ là 1 us). Giả sử ta có một đoạn chương trình như sau:

1. MOVLW 55h
2. MOVWF PORTB
3. CALL SUB_1
4. BSF PORTA,BIT3
5. instruction @ address SUB_1

Ở đây ta chỉ bàn đến qui trình vi điều khiển xử lý đoạn chương trình trên thông qua từng chu kỳ lệnh. Quá trình trên sẽ được thực thi như sau:



Hình 2.3. Cơ chế pipelining

TCY0: đọc lệnh 1

TCY1: thực thi lệnh 1, đọc lệnh 2

TCY2: thực thi lệnh 2, đọc lệnh 3

TCY3: thực thi lệnh 3, đọc lệnh 4.

TCY4: vì lệnh 4 không phải là lệnh sẽ được thực thi theo qui trình thực thi của chương trình (lệnh tiếp theo được thực thi phải là lệnh đầu tiên tại label SUB_1) nên chu kì thực thi lệnh này chỉ được dùng để đọc lệnh đầu tiên tại label SUB_1. Như vậy có thể xem lệnh 3 cần 2 chu kì xung clock để thực thi.

TCY5: thực thi lệnh đầu tiên của SUB_1 và đọc lệnh tiếp theo của SUB_1.

Quá trình này được thực hiện tương tự cho các lệnh tiếp theo của chương trình.

Thông thường, để thực thi một lệnh, ta cần một chu kì lệnh để gọi lệnh đó, và một chu kì xung clock nữa để giải mã và thực thi lệnh. Với cơ chế pipelining được trình bày ở trên, mỗi lệnh xem như chỉ được thực thi trong một chu kì lệnh. Đối với các lệnh mà quá trình thực thi nó làm thay đổi giá trị thanh ghi PC (Program Counter) cần hai chu kì lệnh để thực thi vì phải thực hiện việc gọi lệnh ở địa chỉ thanh ghi PC chỉ tới. Sau khi đã xác định đúng vị trí lệnh trong thanh ghi PC, mỗi lệnh chỉ cần một chu kì lệnh để thực thi xong.

2.2.5. Các dòng PIC và cách lựa chọn PIC

Các kí hiệu của vi điều khiển PIC:

PIC12xxxx: độ dài lệnh 12 bit

PIC16xxxx: độ dài lệnh 14 bit

PIC18xxxx: độ dài lệnh 16 bit

C: PIC có bộ nhớ EPROM (chỉ có 16C84 là EEPROM)

F: PIC có bộ nhớ flash

LF: PIC có bộ nhớ flash hoạt động ở điện áp thấp

LV: tương tự như LF, đây là kí hiệu cũ

Bên cạnh đó một số vi điều khiển có kí hiệu xxFxxx là EEPROM, nếu có thêm chữ A ở cuối là flash (ví dụ PIC16F877 là EEPROM, còn PIC16F877A là flash).

Ngoài ra còn có thêm một dòng vi điều khiển PIC mới là dsPIC.

Ở Việt Nam phổ biến nhất là các họ vi điều khiển PIC do hãng Microchip sản xuất.

Cách lựa chọn một vi điều khiển PIC phù hợp:

Trước hết cần chú ý đến số chân của vi điều khiển cần thiết cho ứng dụng. Có nhiều vi điều khiển PIC với số lượng chân khác nhau, thậm chí có vi điều khiển chỉ có 8 chân, ngoài ra còn có các vi điều khiển 28, 40, 44, ... chân. Cần chọn vi điều khiển PIC có bộ nhớ flash để có thể nạp xóa chương trình được nhiều lần hơn.

Tiếp theo cần chú ý đến các khối chức năng được tích hợp sẵn trong vi điều khiển, các chuẩn giao tiếp bên trong.

Sau cùng cần chú ý đến bộ nhớ chương trình mà vi điều khiển cho phép. Ngoài ra mọi thông tin về cách lựa chọn vi điều khiển PIC có thể được tìm thấy trong cuốn sách “Select PIC guide” do nhà sản xuất Microchip cung cấp.

Ngôn ngữ lập trình cho PIC rất đa dạng. Ngôn ngữ lập trình cấp thấp có MPLAB (được cung cấp miễn phí bởi nhà sản xuất Microchip), các ngôn ngữ lập trình cấp cao hơn bao gồm C, Basic, Pascal, ... Ngoài ra còn có một số ngôn ngữ lập trình được phát triển dành riêng cho PIC như PICBasic, MikroBasic, ...

Đây cũng là một dòng sản phẩm rất đa dạng dành cho vi điều khiển PIC. Có thể sử dụng các mạch nạp được cung cấp bởi nhà sản xuất là hãng Microchip như: PICSTART plus, MPLAB ICD 2, MPLAB PM 3, PROMATE II. Có thể dùng các sản phẩm này để nạp cho vi điều khiển khác thông qua chương trình MPLAB. Dòng sản phẩm chính thống này có ưu thế là nạp được cho tất cả các vi điều khiển PIC, tuy nhiên giá thành rất cao và thường gặp rất nhiều khó khăn trong quá trình mua sản phẩm.

Ngoài ra do tính năng cho phép nhiều chế độ nạp khác nhau, còn có rất nhiều mạch nạp được thiết kế dành cho vi điều khiển PIC. Có thể sơ lược một số mạch nạp cho PIC như sau:

JDM programmer: mạch nạp này dùng chương trình nạp Icprog cho phép nạp các vi điều khiển PIC có hỗ trợ tính năng nạp chương trình điện áp thấp ICSP (In Circuit Serial Programming). Hầu hết các mạch nạp đều hỗ trợ tính năng nạp chương trình này. WARP-13A và MCP-USB: hai mạch nạp này giống với mạch nạp PICSTART PLUS do nhà sản xuất Microchip cung cấp, tương thích với trình biên dịch MPLAB, nghĩa là ta có thể trực tiếp dùng chương trình MPLAB để nạp cho vi điều khiển PIC mà không cần sử dụng một chương trình nạp khác, chẳng hạn như ICprog.

P16PRO40: mạch nạp này do Nigel thiết kế và cũng khá nổi tiếng. Ông còn thiết kế cả chương trình nạp, tuy nhiên ta cũng có thể sử dụng chương trình nạp Icprog.

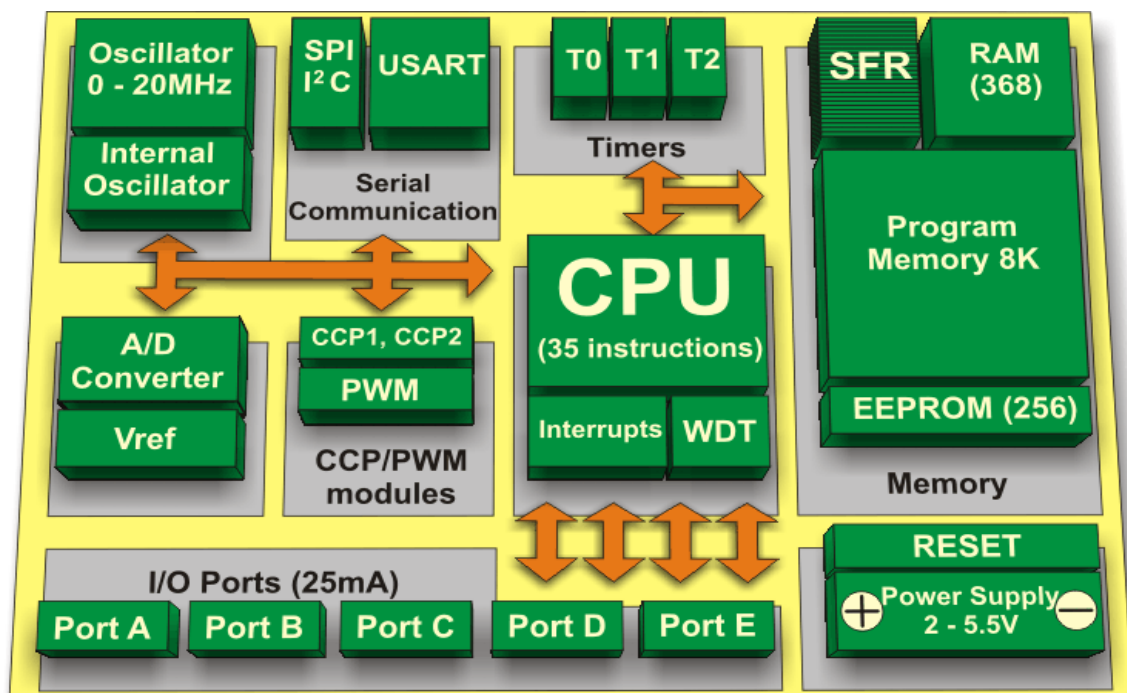
Mạch nạp Universal của Williem: đây không phải là mạch nạp chuyên dụng dành cho PIC như P16PRO40.

Các mạch nạp kể trên có ưu điểm rất lớn là đơn giản, rẻ tiền, hoàn toàn có thể tự lắp ráp một cách dễ dàng, và mọi thông tin về sơ đồ mạch nạp, cách thiết kế, thi công, kiểm tra và chương trình nạp đều dễ dàng tìm được và download miễn phí thông qua mạng Internet. Tuy nhiên các mạch nạp trên có nhược điểm là hạn chế về số vi điều khiển được hỗ trợ, bên cạnh đó mỗi mạch nạp cần được sử dụng với một chương trình nạp thích hợp.

2.3.VI ĐIỀU KHIỂN PIC 18F4680

2.3.1.Giới thiệu về PIC18F4680:

Đây là vi điều khiển thuộc họ PIC18Fxxx với tập lệnh gồm 35 lệnh có độ dài 16 bit. Mỗi lệnh đều được thực thi trong một chu kì xung clock. Tốc độ hoạt động tối đa cho phép là 40 MHz với một chu kì lệnh là 200ns. Bộ nhớ chương trình 48K byte , bộ nhớ dữ liệu SRAM 3328 byte và bộ nhớ dữ liệu EEPROM với dung lượng 1024 byte. Số PORT I/O là 5 với 44 pin I/O.



Hình 2.4. Sơ đồ cấu trúc chung của Pic18f4680

Sơ lược về Vi điều khiển PIC 18F4680:

48 K Flash ROM.

3328 Bytes RAM.

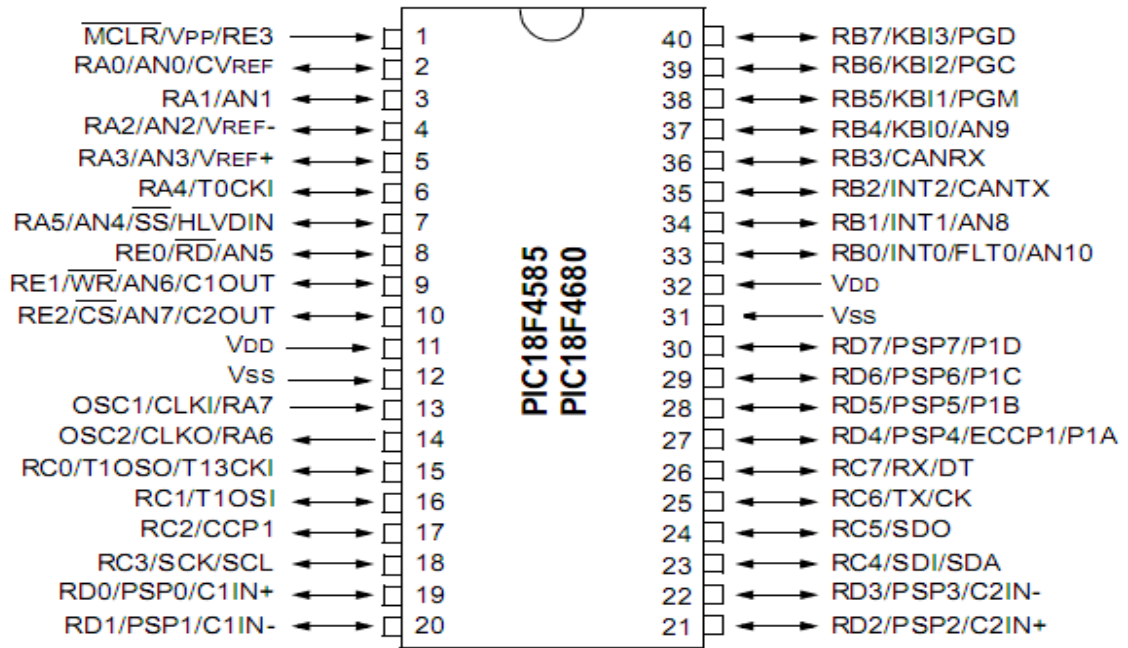
1024 Bytes EEPROM.

5 ports (A, B, C, D, E) vào ra với tín hiệu điều khiển độc lập.

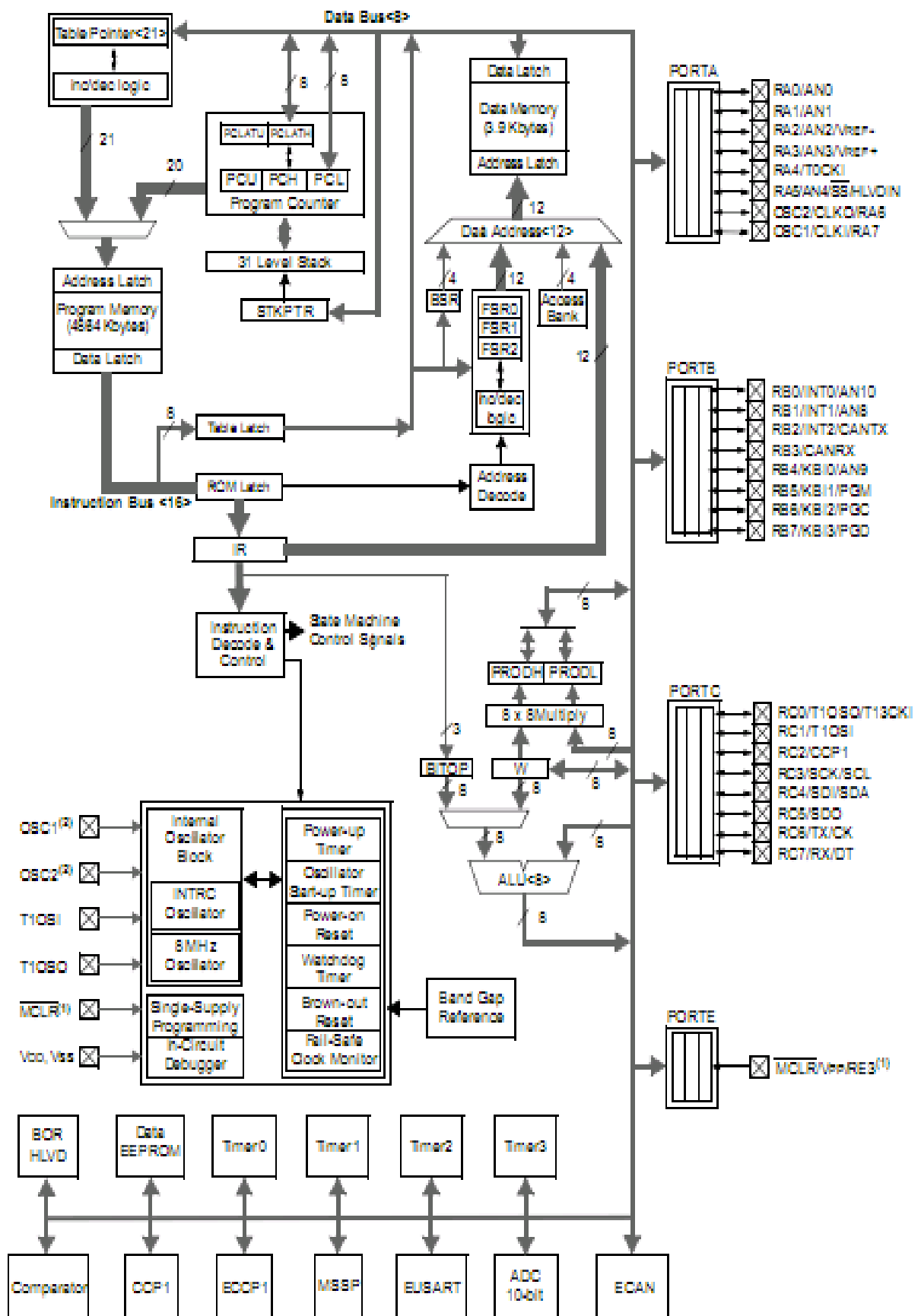
2 bộ định thời 8 bits (Timer 0 và Timer 2).

2 bộ định thời 16 bits (Timer 1 và Timer 3) có thể hoạt động trong chế độ tiết kiệm năng lượng (SLEEP MODE) với nguồn xung Clock ngoài.

- 2 bộ CCP(Capture / Compare/ PWM).
- 1 bộ biến đổi AD 10 bits, 10 ngõ vào.
- 2 bộ so sánh tương tự (Comparator).
- 1 bộ định thời giám sát (WatchDog Timer).
- Một cổng song song 16 bits với các tín hiệu điều khiển.
- Một cổng nối tiếp.
- độ tiết kiệm năng lượng.
- Nạp chương 15 nguồn ngắt.
- Có chế trình bằng cổng nối tiếp ICSP.
- 35 tập lệnh có độ dài 16 bits.
- Tần số hoạt động tối đa 40MHz



Hình 2.5. Hình dạng và sơ đồ chân Pic18f4680



Hình 2.6. Sơ đồ khối của Pic18f4680

Một số điểm đặc biệt của CPU:

a) Dao động:

Pic18F4680 có khả năng sử dụng một trong 4 loại oscillator, đó là:

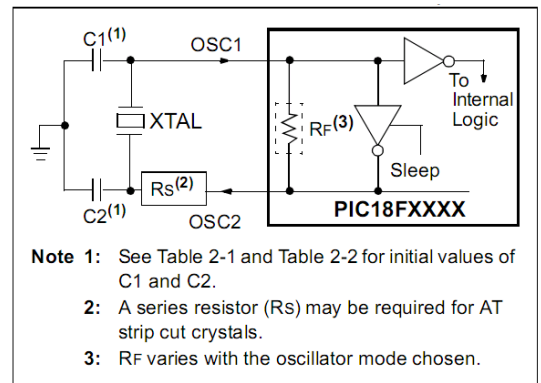
- LP: (low power crystal).
- XT: thạch anh bình thường
- HS: (high-speed crystal).
- RC: (resistor/capacitor)

Trong các chế độ LP, XT và HS chúng ta sử dụng thạch anh dao động nối vào các chân OSC1 và OSC2 để tạo dao động.

Bảng 2.3. lựa chọn tụ trong dao động thạch anh

TABLE 14-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq.	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF

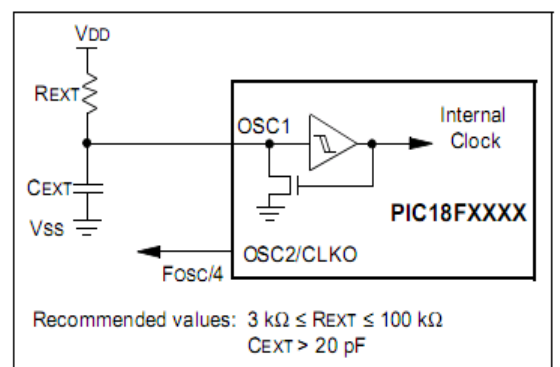


Hình 2.7. Sơ đồ tạo dao động

Việc lựa chọn tụ trong dao động thạch anh dựa vào bảng sau:

Lưu ý: Tụ có giá trị lớn sẽ tăng tính ổn định của dao động nhưng cũng làm tăng thời gian khởi động.

Chế độ dao động RC được sử dụng như một giải pháp tiết kiệm trong các ứng dụng không cần sự chính xác về thời gian.



Hình 2.8. Chế độ dao động RC

b) Cách tính chu kỳ máy:

Ví dụ ta sử dụng thạch anh 10Mhz. Khi đó:

Tần số dao động của thạch anh là $F_{osc} = 10\text{Mhz}$

=> Chu kỳ dao động của thạch anh là $T_{osc} = 1/F_{osc} = 1/10 \cdot 10^6 \text{ (s)}$

Chu kỳ máy:

$$T_{instruction} = 4 \cdot T_{osc} = 4/10 \cdot 10^6 \text{ (s)} = 0.4 \mu\text{s} = 400 \text{ ns}$$

2.3.2. Các chế độ Reset

Có nhiều chế độ reset vi điều khiển, bao gồm:

- Power-on Reset POR (Reset khi cấp nguồn hoạt động cho vi điều khiển).
- Reset trong quá trình hoạt động.
- Từ chế độ sleep.
- WDT reset (reset do khối WDT tạo ra trong quá trình hoạt động).
- WDT wake up từ chế độ sleep.
- Brown-out reset (BOR).

- Power-on reset (POR): Đây là xung reset do vi điều khiển tạo ra khi phát hiện nguồn cung cấp VDD. Khi hoạt động ở chế độ bình thường, vi điều khiển cần được đảm bảo các thông số về dòng điện, điện áp để hoạt động bình thường. Nhưng nếu các tham số này không được đảm bảo, xung reset do POR tạo ra sẽ đưa vi điều khiển về trạng thái reset và chỉ tiếp tục hoạt động khi nào các tham số trên được đảm bảo.

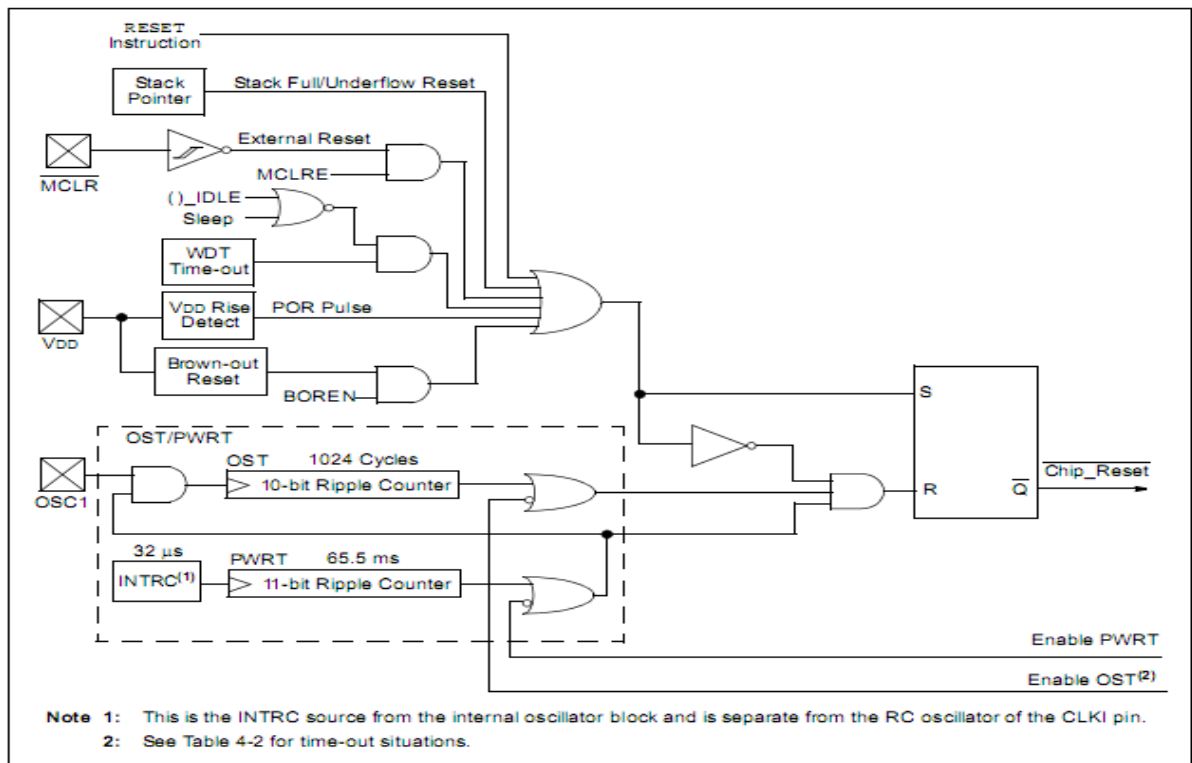
Power-up Timer (PWRT): đây là bộ định thời hoạt động dựa vào mạch RC bên trong vi điều khiển. Khi PWRT được kích hoạt, vi điều khiển sẽ được đưa về trạng thái reset.

PWRT sẽ tạo ra một khoảng thời gian delay (khoảng 72 ms) để VDD tăng đến giá trị thích hợp.

Oscillator Start-up Timer (OST): OST cung cấp một khoảng thời gian delay bằng 1024 chu kỳ xung của oscillator sau khi PWRT ngưng tác động (vi điều khiển đã đủ điều kiện hoạt động) để đảm bảo sự ổn định của xung do

oscillator phát ra. Tác động của OST còn xảy ra đối với POR reset và khi vi điều khiển được đánh thức từ chế độ sleep. OST chỉ tác động đối với các loại oscillator là XT, HS và LP.

Brown-out reset (BOR): Nếu VDD hạ xuống thấp hơn giá trị VBOR (khoảng 4V) và kéo dài trong khoảng thời gian lớn hơn TBOR (khoảng 100 us), BOR được kích hoạt và vi điều khiển được đưa về trạng thái BOR reset. Nếu điện áp cung cấp cho vi điều khiển hạ xuống thấp hơn VBOR trong khoảng thời gian ngắn hơn TBOR, vi điều khiển sẽ không được reset. Khi điện áp cung cấp đủ cho vi điều khiển hoạt động, PWRT được kích hoạt để tạo ra một khoảng thời gian delay (khoảng 72ms). Nếu trong khoảng thời gian này điện áp cung cấp cho vi điều khiển lại tiếp tục hạ xuống dưới mức điện áp VBOR, BOR reset sẽ lại được kích hoạt khi vi điều khiển đủ điện áp hoạt động. Một điểm cần chú ý là khi BOR reset được cho phép, PWRT cũng sẽ hoạt động bất chấp trạng thái của bit PWRT.



Hình 2.9. Sơ đồ các chế độ reset của PIC 18F4680

Tóm lại để vi điều khiển hoạt động được từ khi cấp nguồn cần trải qua các bước sau:

- POR tác động.
- PWRT (nếu được cho phép hoạt động) tạo ra khoảng thời gian delay

TPWRT để ổn định nguồn cung cấp.

- OST (nếu được cho phép) tạo ra khoảng thời gian delay bằng 1024 chu kỳ xung của oscillator để ổn định tần số của oscillator.

Đến thời điểm này vi điều khiển mới bắt đầu hoạt động bình thường. Thanh ghi điều khiển và chỉ thị trạng thái nguồn cung cấp cho vi điều khiển là thanh ghi PCON

a) MCLR

PIC18F4680 có một bộ lọc nhiễu ở phần MCLR. Bộ lọc nhiễu này sẽ phát hiện và bỏ qua các tín hiệu nhiễu.

Ngõ vào MCLR trên chân 4 của PIC18F4680. Khi đưa chân này xuống thấp thì các thanh ghi bên trong VDK sẽ được tải những giá trị thích hợp để khởi động lại hệ thống.

(Lưu ý: reset do WDT không làm chân MCLR cuống mức thấp).

b) Interrupts

PIC18F4680 có nhiều nguồn ngắt khác nhau. Đây là một số ngắt tiêu biểu:

- Ngắt ngoài xảy ra trên chân INT.
- Ngắt do Timer0.
- Ngắt do Timer1.
- Ngắt do Timer2.
- Ngắt do Timer3.
- Ngắt do thay đổi trạng thái trên các chân PortB.
- Ngắt so sánh điện thế.
- Ngắt do Port song song.
- Ngắt USART.
- Ngắt nhận dữ liệu.

- Ngắt truyền dữ liệu .
- Ngắt chuyển đổi ADC.
- Ngắt màn hình LCD.
- Ngắt hoàn tất ghi EEPROM.
- Ngắt module CCP.
- Ngắt Module SSP.

Các thanh ghi chức năng ngắt: INTCON,PIE1,PIR1, PIE2,PIR2 (các thanh ghi này sẽ được nghiên cứu ở các phần sau).

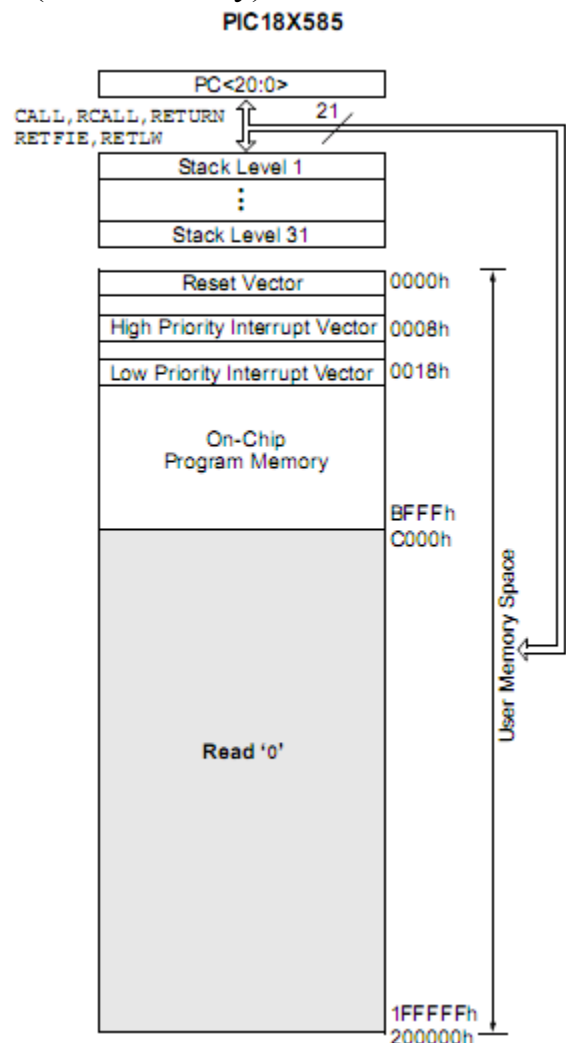
2.3.3. Tổ chức bộ nhớ

Cấu trúc bộ nhớ của vi điều khiển PIC18F4680 bao gồm bộ nhớ chương trình (Program Memory) và bộ nhớ dữ liệu (Data Memory).

a) Bộ nhớ chương trình

Bộ nhớ chương trình của vi điều khiển PIC18F4680 là bộ nhớ flash, dung lượng bộ nhớ 48K byte và được phân thành nhiều trang (từ page0 đến page 3) . Như vậy bộ nhớ chương trình có khả năng chứa được $48 \times 1024 = 49154$ lệnh (vì một lệnh sau khi mã hóa sẽ có dung lượng 1 word (14 bit).

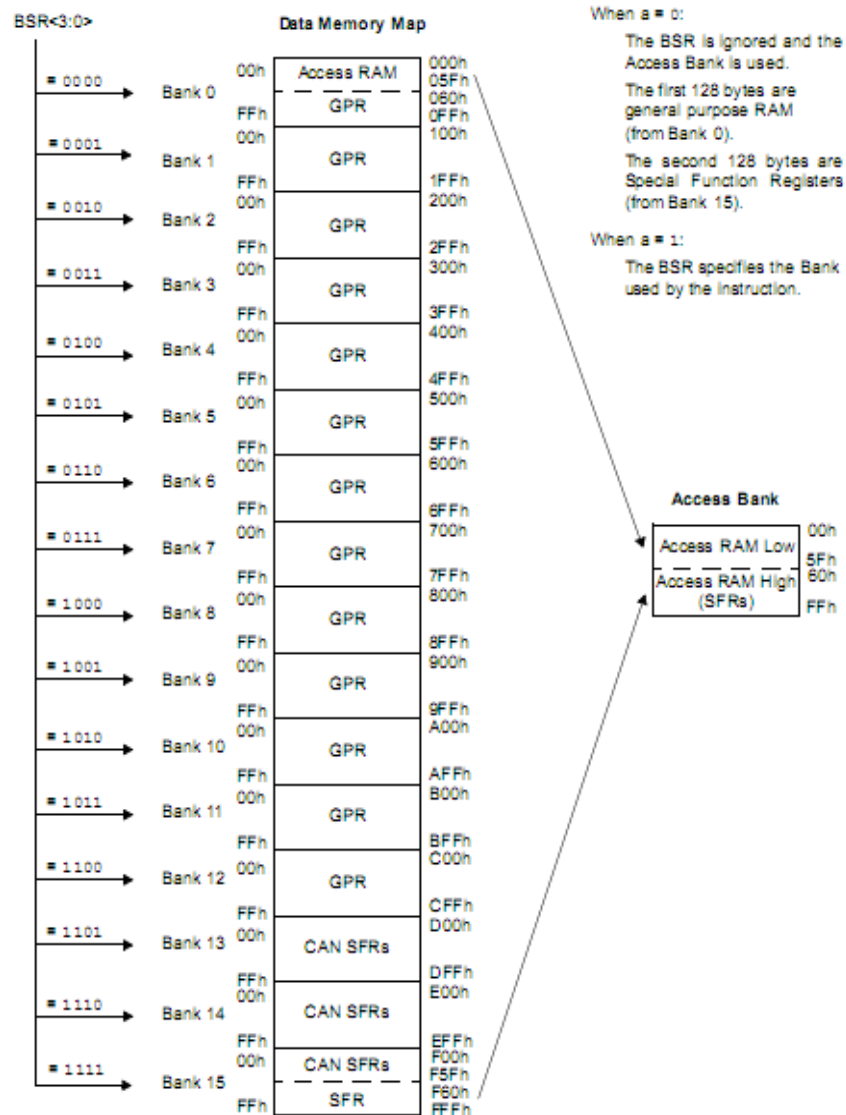
Khi vi điều khiển được reset, bộ đếm chương trình sẽ chỉ đến địa chỉ 0000h (Reset vector). Khi có ngắt xảy ra, bộ đếm chương trình sẽ chỉ đến địa chỉ 0004h (Interrupt vector). Bộ nhớ chương trình không bao gồm bộ nhớ stack và không được địa chỉ hóa bởi bộ đếm chương trình.



Hình 2.10. Bộ nhớ chương trình Pic18f4680

b) Bộ nhớ dữ liệu

Bộ nhớ dữ liệu của PIC là bộ nhớ EEPROM được chia ra làm nhiều bank. Đối với PIC18F4680 bộ nhớ dữ liệu được chia ra làm 16 bank. Mỗi bank có dung lượng 256 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 PIC18F4680 như sau:



Hình 2.11. Bộ nhớ dữ liệu Pic18f4680

c) 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 làm hai loại: thanh ghi SFR liên quan đến các chức năng bên trong (CPU) và thanh ghi SRF dùng để thiết lập và điều khiển các khối chức năng bên ngoài (ví dụ như 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.

Bảng 2.4. Thanh ghi STATUS (03h, 83h, 103h, 183h):

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.

Bảng 2.5. Thanh ghi OPTION_REG (81h, 181h)

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 interrupton- change tại các chân của PORTB.

Bảng 2.6. Thanh ghi INTCON (0Bh, 8Bh, 10Bh, 18Bh)

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.

Bảng 2.7.Thanh ghi PIE1 (8Ch)

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.

Bảng 2.8.Thanh ghi PIR1 (0Ch)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

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

Bảng 2.9.Thanh ghi PIE2 (8Dh)

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	CMIE	—	EEIE	BCLIE	—	—	CCP2IE
bit 7							bit 0

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.

Bảng 2.10.Thanh ghi PIR2 (0Dh)

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	CMIF	—	EEIF	BCLIF	—	—	CCP2IF
bit 7							bit 0

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

Bảng 2.11.Thanh ghi PCON (8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-1
—	—	—	—	—	—	POR	BOR
bit 7							bit 0

d) Thanh ghi mục đích GPR

Các thanh ghi này có thể được truy xuất trực tiếp hoặc gián tiếp thông qua thanh ghi FSG (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.

e) Stack

Stack không nằm trong bộ nhớ chương trình hay bộ nhớ 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 qui trình định trước.

Bộ nhớ Stack trong vi điều khiển PIC họ 18F4680 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 bộ nhớ 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.

2.3.4. Các cổng xuất nhập của Pic18f4680

Cổng xuất nhập (I/O port) chính là phương tiện mà vi điều khiển dùng để tương tác với thế giới bên ngoài. Bên cạnh đó, do vi điều khiển được tích hợp sẵn bên trong các đặc tính giao tiếp ngoại vi nên bên cạnh chức năng là cổng xuất nhập thông thường, một số chân xuất nhập còn có thêm các chức năng khác để thể hiện sự tác động của các đặc tính ngoại vi nêu trên đối với thế giới bên ngoài.

Vi điều khiển PIC18F4585 có 5 cổng xuất nhập, bao gồm PORTA, PORTB, PORTC, PORTD và PORTE.

a) port a

PORTA (RPA) bao gồm 8 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 tự đối với các PORT và các thanh ghi điều khiển tương ứng TRIS (đối với PORTA là TRISA, đối với PORTB là TRISB, đối với PORTC là TRISC, đối với PORTD là TRISD và đối với PORTE là TRISE). 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 bao 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.

Bảng 2.12. Tóm tắt các thanh ghi liên quan đến Port A

TABLE 10-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
PORTA	RA7 ⁽¹⁾	RA6 ⁽¹⁾	RA5	RA4	RA3	RA2	RA1	RA0	52
LATA	LATA7 ⁽¹⁾	LATA6 ⁽¹⁾	LATA Data Output Register						52
TRISA	TRISA7 ⁽¹⁾	TRISA6 ⁽¹⁾	PORTA Data Direction Register						52
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	50
CMCON ⁽²⁾	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	51
CVRCON ⁽²⁾	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	51

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTA.

Note 1: RA7:RA6 and their associated latch and data direction bits are enabled as I/O pins based on oscillator configuration; otherwise, they are read as '0'.

2: These registers are unimplemented on PIC18F2X8X devices.

b) port b

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 bao 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.

Bảng 2.13.Tóm tắt các thanh ghi liên quan đến Port B**TABLE 10-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	52
LATB	LATB Data Output Register (Read and Write to Data Latch)								52
TRISB	PORTB Data Direction Control Register								52
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	49
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	49
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	50

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTB.

c)port c

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.

Bảng 2.14.Tóm tắt các thanh ghi liên quan đến Port C**TABLE 10-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	52
LATC	PORTC Data Output Register								52
TRISC	PORTC Data Direction Register								52

d)port d:

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 bao gồm:

- Thanh ghi PORTD : chứa giá trị các pin trong PORTD.
- Thanh ghi TRISD : điều khiển xuất nhập.

Bảng 2.15.Các thanh ghi liên quan đến Port D

TABLE 10-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
PORTD ⁽¹⁾	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	52
LATD ⁽¹⁾	LATD Data Output Register								52
TRISD ⁽¹⁾	PORTD Data Direction Register								52
TRISE ⁽¹⁾	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction bits			52
ECCP1CON ⁽¹⁾	EPWM1M1	EPWM1M0	EDC1B1	EDC1B0	ECCP1M3	ECCP1M2	ECCP1M1	ECCP1M0	51

e)port e:

PORTE (RPE) gồm 4 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 là các chân điều khiển của chuẩn giao tiếp PSP.

Các thanh ghi liên quan đến PORTE bao 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.
- ADCON1 : thanh ghi điều khiển khối ADC.

Bảng 2.16.Các thanh ghi liên quan đến Port E

TABLE 10-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

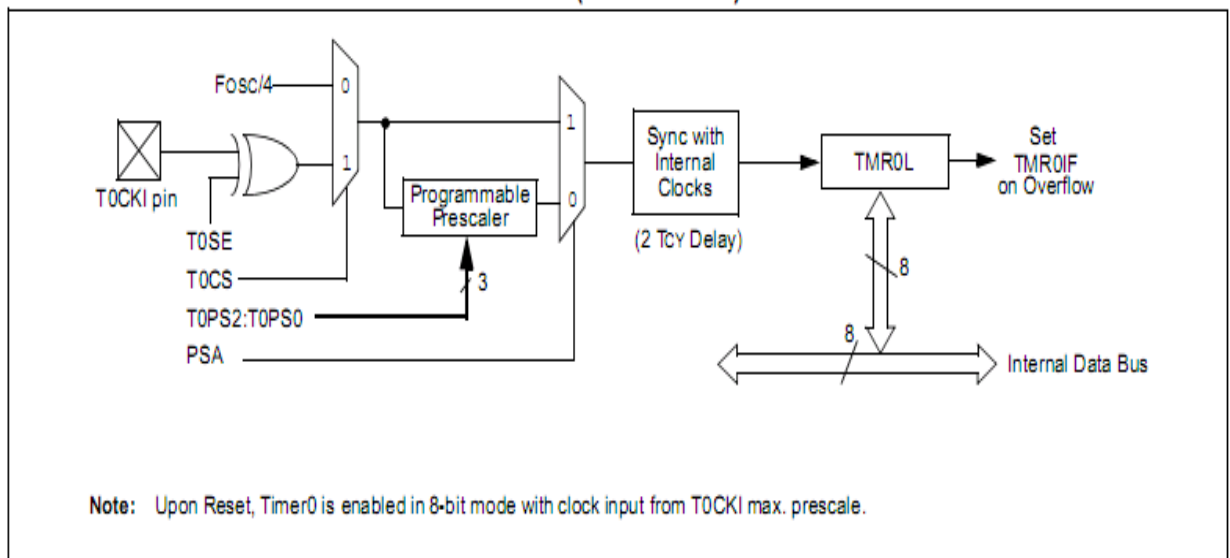
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
PORTE ⁽³⁾	—	—	—	—	RE3 ^(1,2)	RE2	RE1	RE0	52
LATE ⁽²⁾	—	—	—	—	—	LATE Data Output Register			52
TRISE ⁽³⁾	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	52
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	50
CMCON ⁽³⁾	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	51

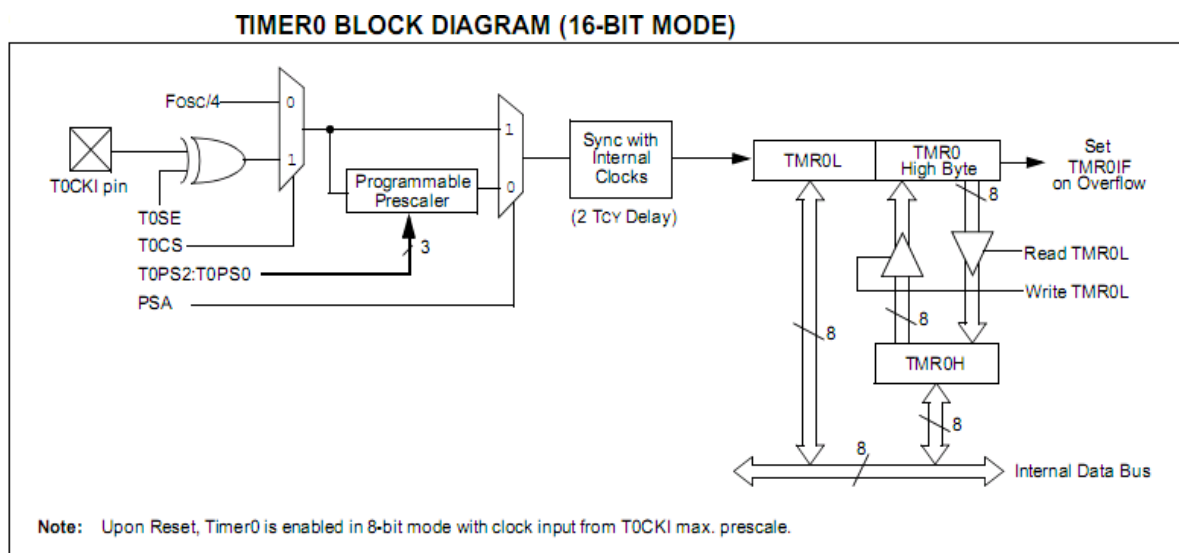
2.3.5.Các timer

a) Timer 0

Đây là một trong ba bộ đếm hoặc bộ định thời của vi điều khiển PIC18F4585. Timer0 là bộ đếm 16 bit hoặc 8 bit được kết nối với bộ chia tần số (prescaler) 16 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.

TIMER0 BLOCK DIAGRAM (8-BIT MODE)





Hình 2.12: Sơ đồ khối của timer 0

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 kỳ xung đồng hồ (tần số vào Timer0 bằng $\frac{1}{4}$ 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.

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

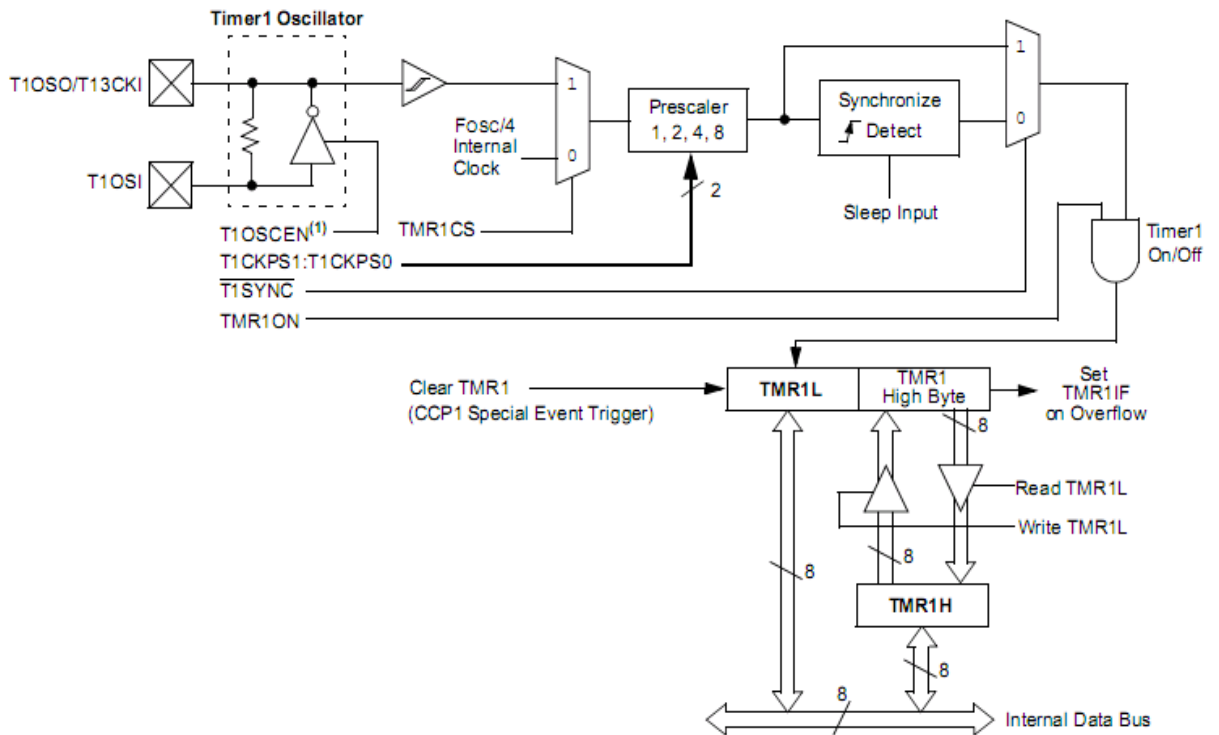
nhưng không làm thay đổi đối tượng tác động của prescaler. Khi đối tượng tác động là WDT, lệnh CLRWDT sẽ xóa prescaler, đồng thời prescaler sẽ ngưng tác vụ hỗ trợ cho WDT.

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.

b)Timer 1

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 $\frac{1}{4}$ 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 (T1CON<1>). Sau đây là sơ đồ khối của Timer1:



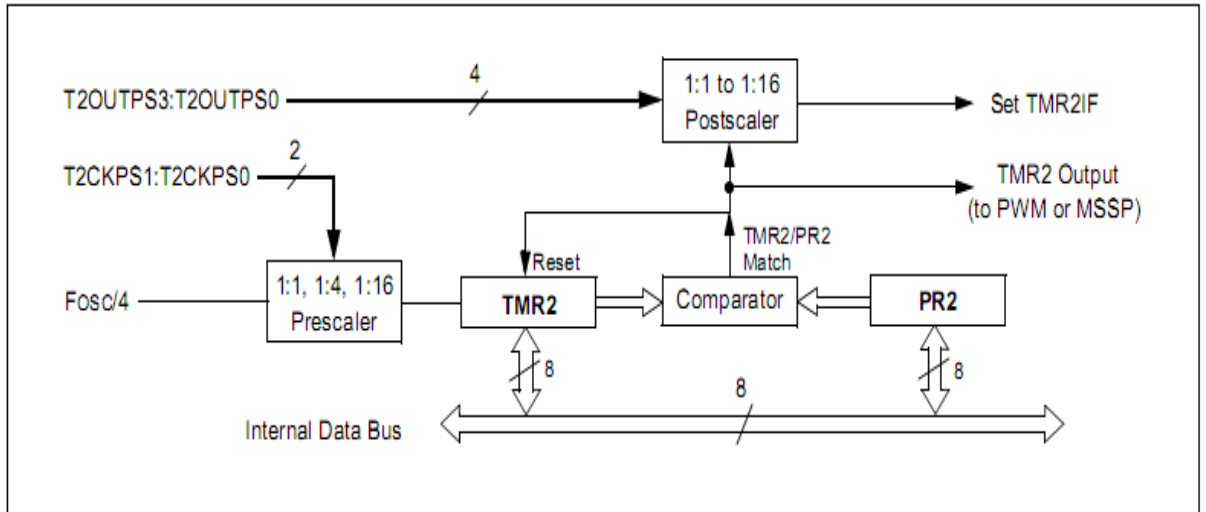
Hình 2.13. Sơ đồ khối của Timer1

Các thanh ghi liên quan đến Timer1 bao 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 0Fh): chứa giá trị 8 bit cao của bộ đếm Timer1.
- T1CON (địa chỉ 10h): xác lập các thông số cho Timer1.

d)Timer 2

Timer2 là bộ định thời 8 bit và được hỗ trợ bởi hai bộ chia tần số prescaler và postscaler. 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 1/4 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 1:16 và được điều khiển bởi các bit T2CKPS1:T2CKPS0 (T2CON<1:0>)).



Hình 2.14. Sơ đồ khối của Timer 2

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 giao tiếp SSP.

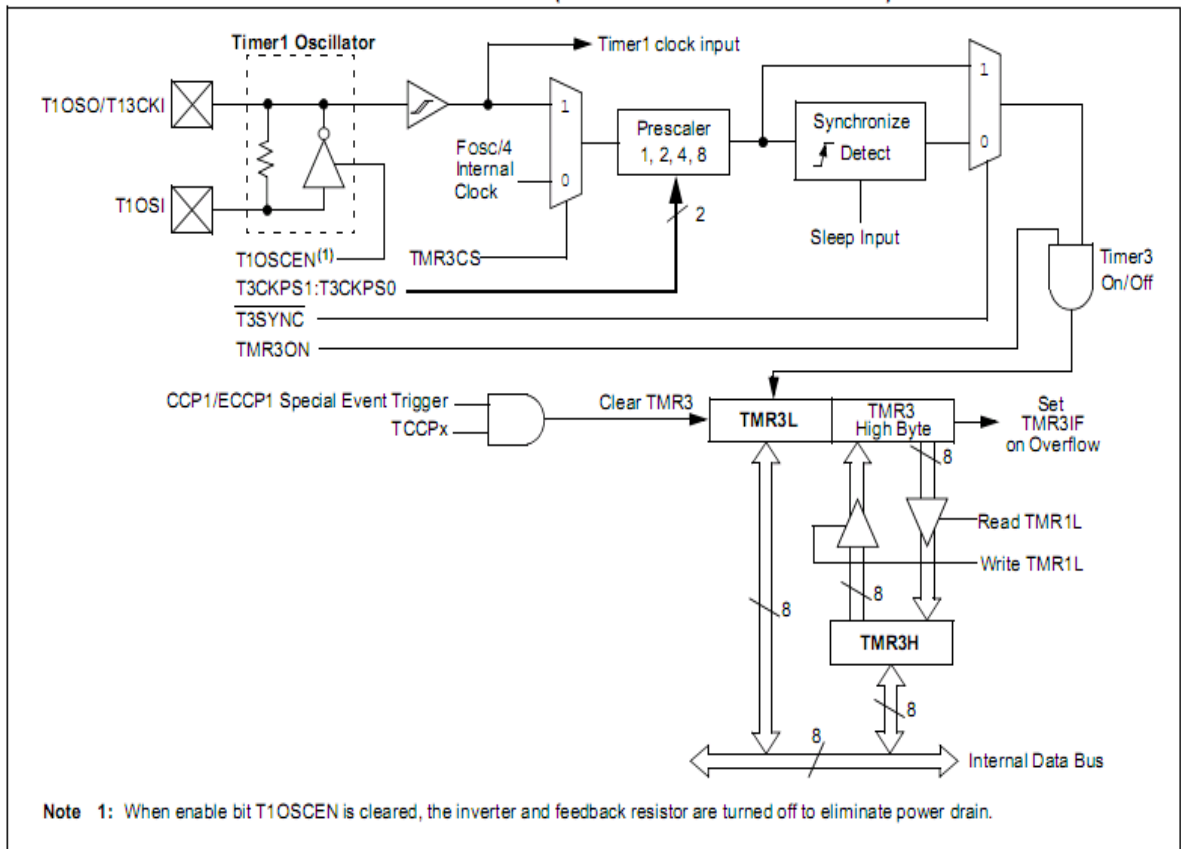
Các thanh ghi liên quan đến Timer2 bao gồm:

- 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.

Timer0 và Timer2 là bộ đếm 8 bit (giá trị đếm tối đa là FFh), trong khi Timer1 là bộ đếm 16 bit (giá trị đếm tối đa là FFFFh). Timer0, Timer1 và Timer2 đều có hai chế độ hoạt động là timer và counter. Xung clock có tần số bằng $\frac{1}{4}$ tần số của oscillator. Xung tác động lên Timer0 được hỗ trợ bởi prescaler và có thể được thiết lập ở nhiều chế độ khác nhau (tần số tác động, cạnh tác động) trong khi các thông số của xung tác động lên Timer1 là cố định. Timer2 được hỗ trợ bởi hai bộ chia tần số prescaler và postcaler độc lập, tuy nhiên cạnh tác động vẫn được cố định là cạnh lên. Timer1 có quan hệ với khối CCP, trong khi Timer2 được kết nối với khối SSP.

e) Timer 3

Timer1 là bộ định thời 16 bit, giá trị của Timer1 sẽ được lưu trong hai thanh ghi (TMR3H:TMR3L). Cờ ngắt của Timer1 là bit TMR3IF (PIR1<0>). Bit điều khiển của Timer3 sẽ là TMR3IE (PIE<0>). Tương tự như Timer0, Timer1, Timer2, Timer3 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 $\frac{1}{4}$ 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 (T3CON<1>).

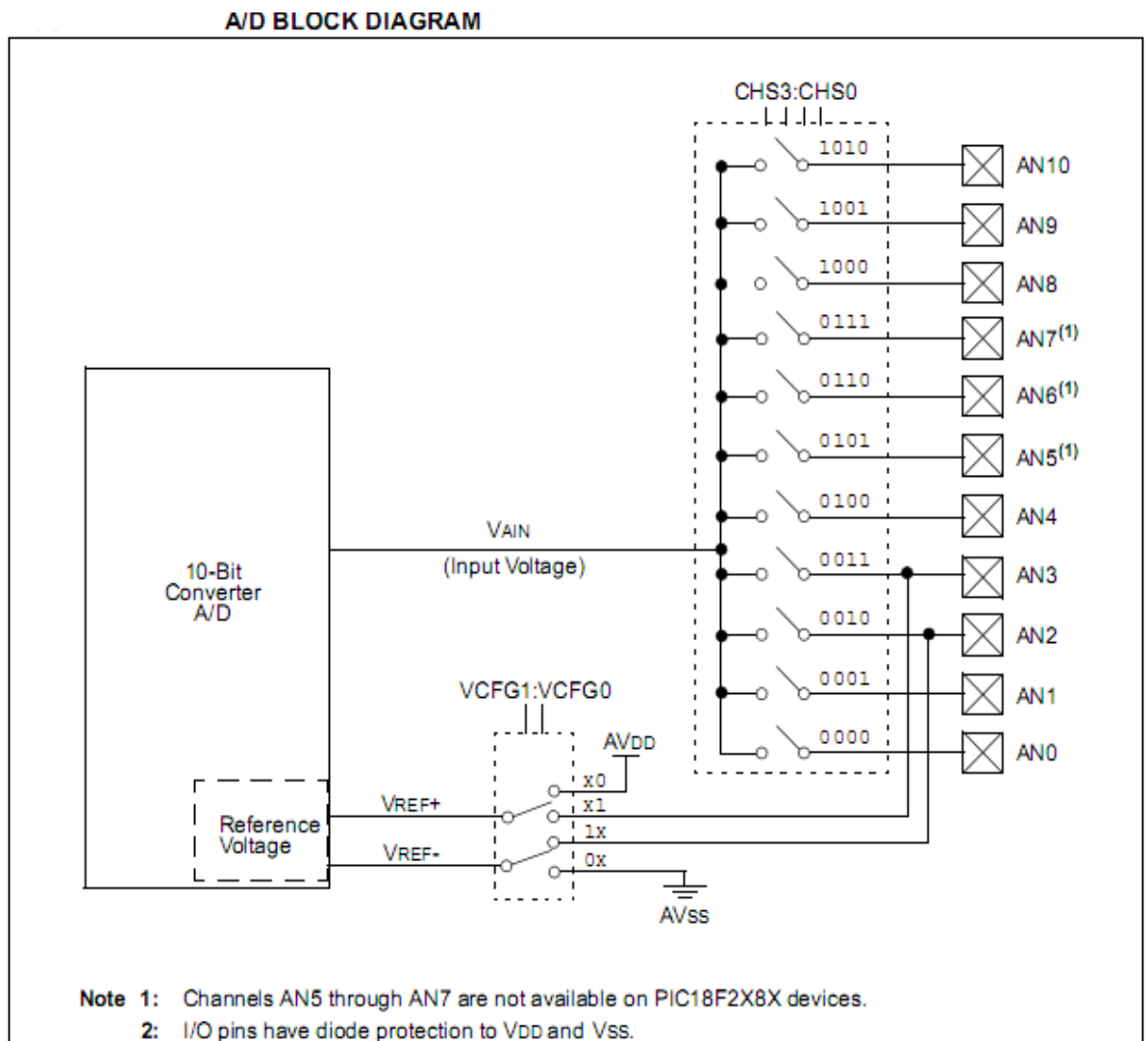


Hình 2.15. Sơ đồ khối của Timer 2

f) ADC

ADC (Analog to Digital Converter) là bộ chuyển đổi tín hiệu giữa hai dạng tương tự và số. PIC18F4680 có 11 ngõ vào analog (RA0,RA1,RA2,RA3,RA5;

RE2:RE0 và RB0, RB1, RB7). Hiệu điện thế chuẩn VREF có thể được lựa chọn là VDD, VSS hay hiệu điện thế chuẩn được xác lập trên hai chân RA2 và RA3



Hình 2.16. Sơ đồ khối ADC

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.

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

- INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): cho phép các 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.

g) Giao tiếp nối tiếp

USART (Universal Synchronous Asynchronous Receiver Transmitter) là một trong hai chuẩn giao tiếp nối tiếp. USART còn được gọi là giao diện giao tiếp nối tiếp SCI

(Serial Communication Interface). Có thể sử dụng giao diện này cho các giao tiếp với các

thiết bị ngoại vi, với các vi điều khiển khác hay với máy tính. Các dạng của giao diện USART ngoại vi bao gồm:

- Bất đồng bộ (Asynchronous).
- Đồng bộ_ Master mode.
- Đồng bộ_ Slave mode.

PIC16F4680 được tích hợp sẵn bộ tạo tốc độ baud BRG (Baud Rate Genetator) 8 bit hoặc 16bit dùng cho giao diện USART. BRG thực chất là một bộ đếm có thể được sử dụng cho cả hai dạng đồng bộ và bất đồng bộ và được điều khiển bởi thanh ghi PSBRG. Ở dạng bất đồng bộ, BRG còn được điều khiển bởi bit BRGH (TXSTA<2>). Ở dạng đồng bộ tác động của bit BRGH được bỏ qua. Tốc độ baud do BRG tạo ra được tính theo công thức sau:

Bảng 2.15. Tốc độ baud do BRG

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64 (X + 1))$	Baud Rate = $F_{osc}/(16 (X + 1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4 (X + 1))$	N/A

Trong đó X là giá trị của thanh ghi RSBRG (X là số nguyên và $0 < X < 255$).

Các thanh ghi liên quan đến BRG bao gồm:

- TXSTA (địa chỉ 98h): chọn chế độ đồng bộ hay bất đồng bộ (bit SYNC) và chọn mức tốc độ baud (bit BRGH).

- RCSTA (địa chỉ 18h): cho phép hoạt động cổng nối tiếp (bit SPEN).

- RSBRG (địa chỉ 99h): quyết định tốc độ baud.

* USART bất đồng bộ: Ở chế độ truyền này USART hoạt động theo chuẩn NRZ (None-Return-to-Zero), nghĩa là các bit truyền đi sẽ bao gồm 1 bit Start, 8 hay 9 bit dữ liệu (thông thường là 8 bit) và 1 bit Stop. Bit LSB sẽ được truyền đi trước. Các khối truyền và nhận data độc lập với nhau sẽ dùng chung tần số tương ứng với tốc độ baud cho quá trình dịch dữ liệu (tốc độ baud gấp 16 hay 64 lần tốc độ dịch dữ liệu tùy theo giá trị của bit BRGH), và để đảm bảo tính hiệu quả của dữ liệu thì hai khối truyền và nhận phải dùng chung một định dạng dữ liệu.

Các thanh ghi liên quan đến quá trình truyền dữ liệu bằng giao diện USART bất đồng bộ:

- Thanh ghi INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): cho phép tắt cả các ngắt.

- Thanh ghi PIR1 (địa chỉ 0Ch): chứa cờ hiệu TXIF.

- Thanh ghi PIE1 (địa chỉ 8Ch): chứa bit cho phép ngắt truyền TXIE.

- Thanh ghi RCSTA (địa chỉ 18h): chứa bit cho phép cổng truyền dữ liệu (hai pin

RC6/TX/CK và RC7/RX/DT).

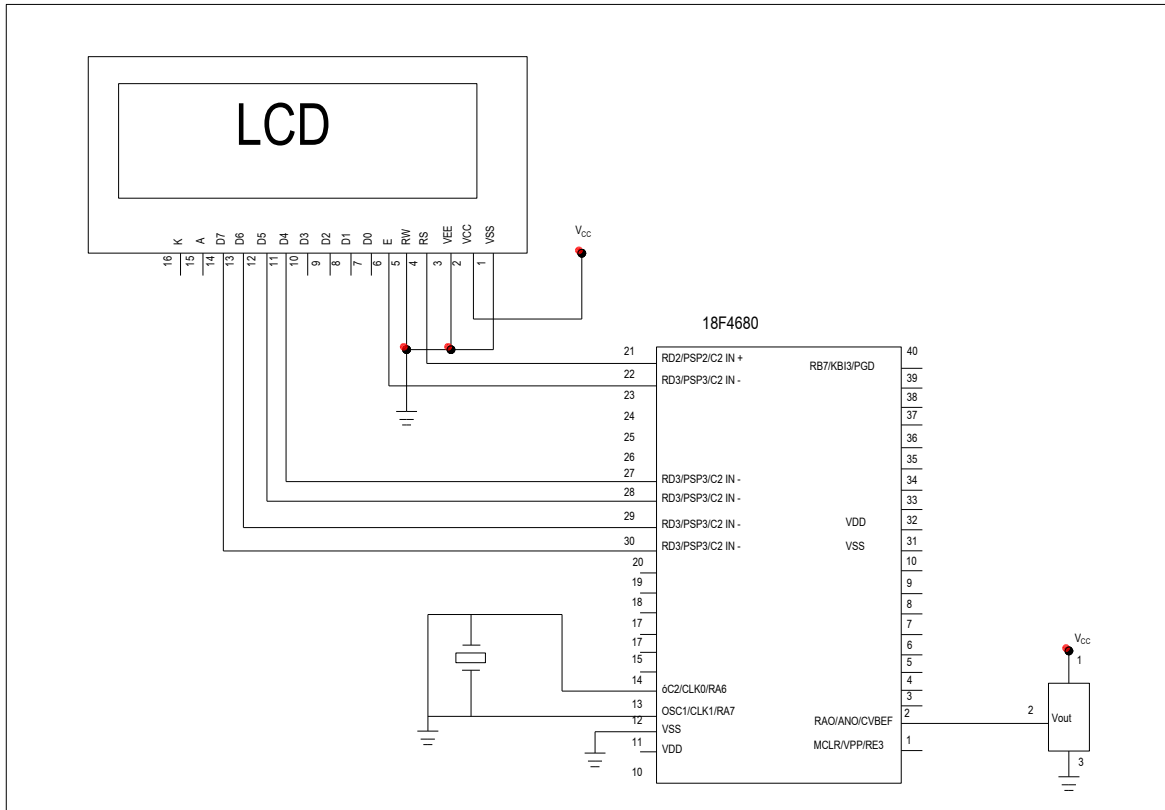
- Thanh ghi TXREG (địa chỉ 19h): thanh ghi chứa dữ liệu cần truyền.
- Thanh ghi TXSTA (địa chỉ 98h): xác lập các thông số cho giao diện.
- Thanh ghi SPBRG (địa chỉ 99h): quyết định tốc độ baud.

* USART đồng bộ: Giao diện USART đồng bộ được kích hoạt bằng cách set bit SYNC. Cổng giao tiếp nối tiếp vẫn là hai chân RC7/RX/DT, RC6/TX/CK và được cho phép bằng cách set bit SPEN. USART cho phép hai chế độ truyền nhận dữ liệu là Master mode và Slave mode. Master mode được kích hoạt bằng cách set bit CSRC (TXSTA<7>), Slave mode được kích hoạt bằng cách clear bit CSRC. Điểm khác biệt duy nhất giữa hai chế độ này là Master mode sẽ lấy xung clock đồng bộ từ bộ tạo xung baud BRG còn Slave mode lấy xung clock đồng bộ từ bên ngoài qua chân RC6/TX/CK. Điều này cho phép Slave mode hoạt động ngay cả khi vi điều khiển đang ở chế độ sleep.

Các thanh ghi liên quan đến quá trình truyền dữ liệu bằng giao diện USART đồng bộ Master mode:

- Thanh ghi INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): cho phép tắt cả các ngắt.
- Thanh ghi PIR1 (địa chỉ 0Ch): chứa cờ hiệu TXIF.
- Thanh ghi PIE1 (địa chỉ 8Ch): chứa bit cho phép ngắt truyền TXIE.
- Thanh ghi RCSTA (địa chỉ 18h): chứa bit cho phép cổng truyền dữ liệu (hai pin RC6/TX/CK và RC7/RX/DT).
- Thanh ghi TXREG (địa chỉ 19h): thanh ghi chứa dữ liệu cần truyền.
- Thanh ghi TXSTA (địa chỉ 98h): xác lập các thông số cho giao diện.
- Thanh ghi SPBRG (địa chỉ 99h): quyết định tốc độ baud.

2.4.6. Sơ đồ ghép nối mạch đo nhiệt độ



Mạch đo nhiệt độ LM35 hiển thị trên LCD 16x2

2.3.7 code mạch đo nhiệt độ, hiển thị trên LCD.

```
//=====
// Ten chương trình : Mạch đo nhiệt độ
// Mô tả phần cứng : Dùng PIC16F4680 - thạch anh 20MHz
// : LCD giao tiếp với PORTD
// : Đầu ra LM35 đưa vào chân AN0
//-----
// Chú thích : hoặc các chú thích khác
// : dùng chế độ Power On Reset
// : chương trình viết cho PIC Tutorial
#include <16F4680.h>
#include <def_4680.h>
#define * =16 adc=10
#FUSES NOWDT, HS, NOPUT, NOPROTECT, NODEBUG,
NOBROWNOUT,
NOLVP, NOCPD, NOWRT
```

```

#include <math>
#include <math>
#include <math>
int8 low,high,key,mode,min,max,model,i;
int1 do_F;
void convert_bcd(int8 x);
void bao_dong();
void test();
//-----
void main()
{ float value;
  on_off =1;
  min  =15; //nhiet do min default
  max  =35; //nhiet do max default
  do_F =0 ;
  i = 50 ;
  mode =0 ;
  model = 0 ;
  trisa = 0xFF;
  trisb = 0x01;
  trisd = 0x00;
  LCD_init();
  Printf(LCD_putchar,"dc 1201");
  LCD_putcmd(0xC0);
  Printf(LCD_putchar,"Khoi tao...");
// Khoi tao cho ngat ngoai
  enable_interrupts (INT_EXT);
  ext_int_edge(H_TO_L);
  enable_interrupts (GLOBAL);
// Khoi tao che do cho bo ADC
  setup_adc_ports(AN0);
  setup_adc(ADC_CLOCK_INTERNAL);
  delay_us(10);
// Lay mau nhiet do lan dau tien
  value=(float)read_adc();
  value = (value - 558.5)/2.048; // For 5V supply
  // value = (value - 754.8)/2.048; // For 3.7V Supply
  // value = (value - 698.2)/2.048; // For 4V supply
  convert_bcd((int8)value); // Tach so tram, chuc, donvi de hien thi len
LED 7
  delay_ms(1000);
  LCD_putcmd(0xC0);

```

```

Printf(LCD_putchar, "Khoi tao xong");
while(1)
{
if (i==50)
{
value = read_adc();

value=(value-558.5)/2.048;
if (do_F==1) value=1.8*value+32;
convert_bcd((int8)value);
printf("\n\rNhiet do phong: %u",value);// Gui gia tri len may tinh
LCD_putcmd(0xC0);
printf(LCD_putchar, " Temp = ");
LCD_putchar(high); LCD_putchar(low);
if (do_F==0) printf(LCD_putchar, " C");
else printf(LCD_putchar, " F");
i=0;
}
i++;
if(((int8)value > 40) || ((int8)value < 15)) on_off=1;
else
{
on_off = 0;
LCD_Putcmd(0xCF);
LCD_putchar(" ");
blink=0;
}
if (on_off==1)
{
if (blink==0) {
LCD_Putcmd(0xCF);LCD_putchar("!");blink=1;delay_ms(250);}
else {LCD_Putcmd(0xCF);LCD_putchar("
");blink=0;delay_ms(250);}
}
}
} //end main-----
#INT_EXT

void test()
{
if (do_F == 1) do_F=0;
else do_F=1;
}

```

```

}
void convert_bcd(int8 x)
{
    low=x%10; //chia lay phan du, so hang don vi

    high=x/10; //tach hang tram va hang chuc
    low = low + 0x30;
    high = high + 0x30;
}
void bao_dong(){
int8 i;
if (blink == 0) blink = 1;
else      blink=0;
    for(i=0;i<50;i++)
    {
        LCD_Putcmd(0xCF);
        if (blink==0) LCD_putchar(□!□);
        else      LCD_putchar(□ □);
    }
}

```


CHƯƠNG 3.

GỬI TÍN HIỆU ĐO LÊN INTERNET

3.1.TỔNG QUAN VỀ MẠNG INTERNET

3.1.1.Giới thiệu

Internet là một hệ thống thông tin toàn cầu có thể được truy nhập công cộng gồm các mạng máy tính được liên kết với nhau. Hệ thống này truyền thông tin theo kiểu nối chuyển gói dữ liệu (packet switching) dựa trên một giao thức liên mạng đã được chuẩn hóa (giao thức IP). Hệ thống này bao gồm hàng ngàn mạng máy tính nhỏ hơn của các doanh nghiệp, của các viện nghiên cứu và các trường đại học, của người dùng cá nhân, và các chính phủ trên toàn cầu. Chúng cung cấp một khối lượng thông tin và dịch vụ khổng lồ trên Internet. Mạng Internet mang lại rất nhiều tiện ích hữu dụng cho người sử dụng, một trong các tiện ích phổ thông của Internet là hệ thống thư điện tử (email), trò chuyện trực tuyến (chat), máy truy tìm dữ liệu (search engine), các dịch vụ thương mại và chuyển ngân, và các dịch vụ về y tế giáo dục như là chữa bệnh từ xa hoặc tổ chức các lớp học ảo.

Nguồn thông tin khổng lồ kèm theo các dịch vụ tương ứng chính là hệ thống các trang Web liên kết với nhau và các tài liệu khác trong WWW (World Wide Web). Trái với một số cách sử dụng thường ngày, Internet và WWW không đồng nghĩa. Internet là một tập hợp các mạng máy tính kết nối với nhau bằng dây đồng, cáp quang, v.v..; còn WWW, hay Web, là một tập hợp các tài liệu liên kết với nhau bằng các siêu liên kết (hyperlink) và các địa chỉ URL, và nó có thể được truy nhập bằng cách sử dụng Internet. Trong tiếng Anh, sự nhầm lẫn của đa số dân chúng về hai từ này thường được châm biếm bằng những từ như "the intarweb".

3.1.2. Hoạt động của mạng Internet

Internet dựa trên mô hình khách – chủ (client – server), trong đó dữ liệu được trao đổi thông qua các trang web. Trong mô hình client – server, mỗi máy tính được xác định bởi một địa chỉ Internet protocol (IP) và cả máy tính client, server cùng chấp nhận một giao thức chung để giao tiếp với nhau.

a) Các giao thức (Internet protocols)

- Các giao thức là tập các luật mà các máy tính phải tuân theo khi giao tiếp trên Internet.

- Transmission Control Protocol (TCP): thiết lập kết nối giữa hai máy tính để truyền tải dữ liệu, chia dữ liệu thành những gói nhỏ và đảm bảo việc truyền nhận dữ liệu. TCP là giao thức hướng kết nối (connection-oriented protocol).

- User Datagram Protocol (UDP): thiết lập kết nối nhanh nhưng không chắc chắn giữa các máy tính để truyền tải dữ liệu, cung cấp ít dịch vụ để khắc phục lỗi.

- Internet Protocol (IP): điều chỉnh đường đi của những gói dữ liệu đường truyền nhận trên Internet. TCP là giao thức phi kết nối (connectionless protocol).

- HTTP: cho phép trao đổi thông tin trên Internet.

- FTP: cho phép truyền nhận file trên Internet.

- SMTP: cho phép gửi thư điện tử trên Internet.

- POP3: cho phép nhận thư điện tử trên Internet.

b) Địa Chỉ IP (IP address)

Internet là một mạng kết nối rộng lớn giữa các máy tính. Để xác định một máy tính trên mạng này, người ta dùng một con số gọi là địa chỉ IP. Địa chỉ IP gồm một tập 4 số nhỏ hơn 255 và được ngăn cách bởi các dấu ‘.’. Ví dụ: 41.212.196.197.

c) Mô hình khách – chủ (Client – server model)

Trong mô hình khách - chủ, máy khách (client computer) yêu cầu thông tin từ một máy chủ (server). Máy chủ chấp nhận yêu cầu và gửi thông tin về cho máy khách. Việc trao đổi thông tin này được diễn ra thông qua những trang web.

d) Hệ thống tên miền (Domain name system)

Mỗi máy tính trên mạng Internet được xác định bằng địa chỉ IP, nhưng con số này rất khó nhớ. Để khắc phục nhược điểm này, người ta dùng hệ thống tên miền để đặt tên cho máy tính. Ví dụ: tên miền www.yahoo.com ứng với địa chỉ IP 216.109.127.28. request web page Miền (domain) ứng với một tập các máy tính trên Internet. Phần mở rộng của tên miền (domain name extension) được dùng để xác định quốc gia hay tổ chức.

3.1.3. Kết nối Internet

Để kết nối với Internet cần có một số yêu cầu về phần cứng và phần mềm sau:

-Phần cứng: máy tính, kết nối thông qua đường dây điện thoại hoặc kết nối cáp, modem.

-Phần mềm: kết nối internet, hệ điều hành, giao thức TCP/IP, trình duyệt web.

Các yêu cầu thiết yếu khi kết nối với Internet: kết nối với Internet thông qua một nhà cung cấp Internet (Internet Service Provider), modem, trình duyệt và địa chỉ URL

-TCP/IP được dùng làm giao thức chuẩn khi giao tiếp trên Internet vì nó độc lập với nền của hệ thống (platform independent) và không có tổ chức nào có quyền sở hữu giao thức này..

3.2. GIỚI THIỆU IC ENC28J60

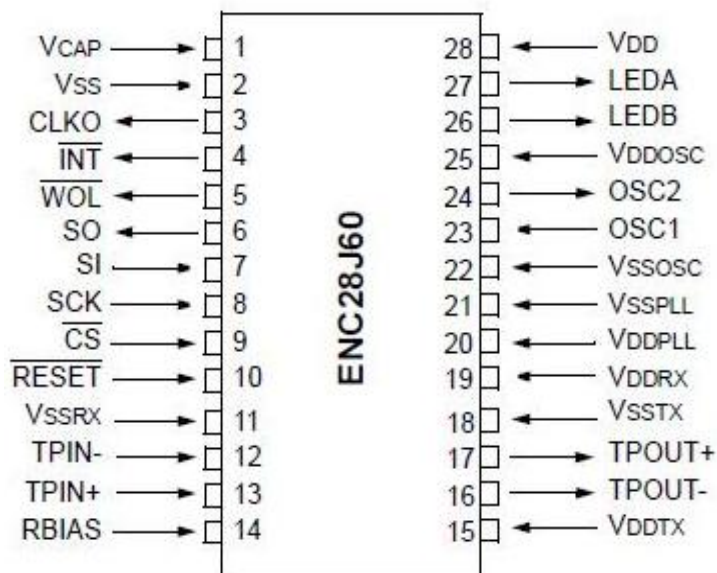
3.2.1. Giới thiệu chung về IC ENC28J60

Trên thế giới có rất nhiều hãng cung cấp vi mạch hỗ trợ sẵn sàng cho các giao tiếp theo giao thức TCP/IP, như các hãng lớn: Realtek, Conexant,

MicroChip,... Tuy nhiên, xét trên quan điểm của người thiết kế thì các chip được đánh giá cao là chip đáp ứng được nhu cầu thực tế (không thiếu và cũng không quá dư thừa tính năng), được sự hỗ trợ tối đa từ hãng về các thư viện lập trình, dễ dàng giao tiếp với các vi xử lý, mạch thiết kế đơn giản, ít đòi hỏi các linh kiện ngoài. Từ những tiêu chí đó, cùng với sự ra đời dòng chip hỗ trợ hoàn toàn giao thức ethernet, việc lựa chọn ENC28J60 của MicroChip là lựa chọn rất phù hợp, cả về tính năng kỹ thuật và giá cả. Ngoài ra, hãng hỗ trợ đầy đủ tài liệu và các thư viện lập trình, rất thân thiện khi ghép nối với các họ vi xử lý có trên thị trường.

ENC28J60 là một bộ điều khiển Ethernet độc lập với SPI. Vi mạch ENC28J60 được sử dụng như một giao diện Ethernet cho bất kỳ thiết bị ngoại vi nào.

ENC28J60 đáp ứng tất cả các kỹ thuật theo chuẩn IEEE802.3. Nó cung cấp một Model DMA nội bộ cho phép truyền dữ liệu nhanh và được sử dụng trong các giao thức mạng.



Hình 3.1. Sơ đồ chân ENC28J60

3.2.2. Cấu trúc và chức năng

-Giao tiếp với các thiết bị ngoại vi theo chuẩn SPI với tốc độ lên tới 10Mbps.

-Giao thức chuẩn IEEE802.3.

-Bộ lọc đặc biệt tự động đánh giá, chấp nhận hoặc từ chối MagicPacket, unicast, multicast.

-Có bộ nhớ đệm Sram lên tới 8kb, có thể quản lý dữ liệu đáng tinh cậy và linh hoạt.

3.2.3. Chức năng các chân của ENC28J60:

- V_{DD} , V_{DDOSC} , V_{DDPLL} , V_{DDRFX} , V_{DDTX} : các chân này dùng để cấp nguồn nuôi cho IC, nối với +3,3V, .

- V_{SS} , V_{SSOSC} , V_{SSPLL} , V_{SSRX} , V_{SSTX} : các chân này được nối Mass.

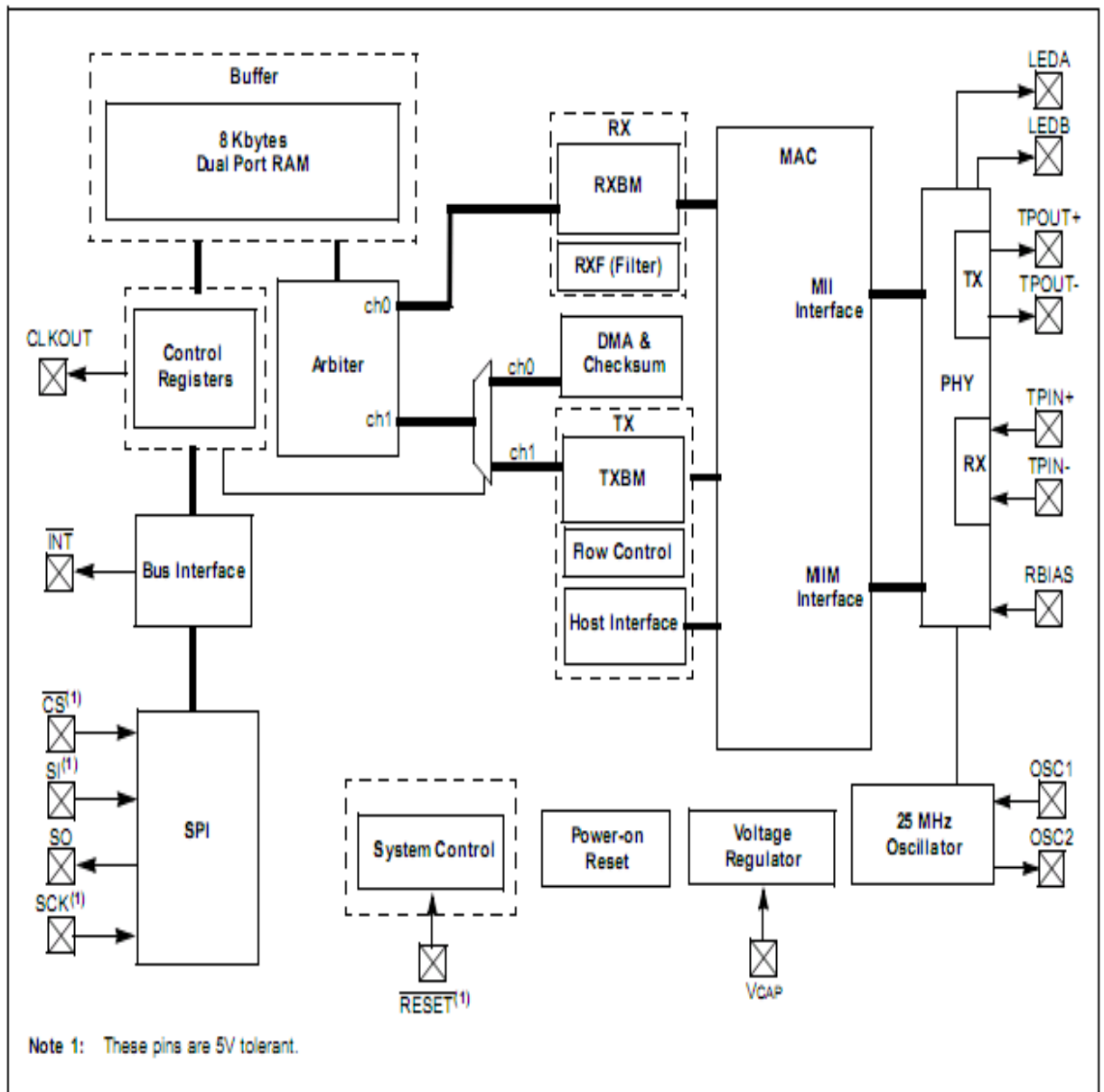
-/RESET: chân Reset về mức thấp để thiết lập lại chế độ cho chip.

-OSC1,OSC2: 2 chân nhân bộ giao đồng bên ngoài, ENC28J60 hoạt động với thạch anh 25Mhz.

-CLOUT: tạo ra xung clock khoảng 6,25MHz.

-LedA, LedB: có chức năng hỗ trợ tự động phát hiện và Reset.

3.2.4. Các khối cơ bản bên trong ENC28J60:



Hình 3.2. Các khối cơ bản bên trong ENC28J60

Bao gồm 7 khối cơ bản sau:

1. Một giao diện SPI dùng để giao tiếp với vi điều khiển.
2. Khối giám sát và điều khiển ENC28J60.
3. Bộ nhớ đệm Sram cho việc truyề và nhận gói tin.
4. Khối giám sát việc truy cập vào bộ nhớ đệm.
5. Khối Bus dữ liệu giả mã lệnh thông qua SPI.

6. Module Mac theo chuẩn IEEE802.3.

7. Module PHY mã hóa và giải mã tín hiệu điện.

3.2.5. Bộ thư viện TCP/IP STACK

a) Sơ lược về TCP/IP Stack của Microchip

Bộ thư viện TCP / IP Stack của Microchip cung cấp một nền tảng cho các ứng dụng Ethernet do xử lý hầu hết các yêu cầu tương tác giữa các tầng vật lý và tầng ứng dụng. Nó bao gồm các ứng dụng phổ biến của tầng ứng dụng, bao gồm HTTP2 phục vụ các trang web, SMTP để gửi e-mail, SNMP cung cấp tình trạng, kiểm soát, Telnet, TFTP, -to Serial Ethernet, và nhiều hơn nữa. Ngoài ra, bộ thư viện TCP/IP stack này có dung lượng tương đối nhẹ và hiệu suất triển khai cao của TCP và UDP tầng giao vận, cũng như các module hỗ trợ khác như IP, ICMP, DHCP, ARP, và DNS.

Bộ thư viện TCP / IP Stack của Microchip. Tối ưu hoá cho tất cả các PIC18, PIC24, dsPIC và gia đình vi điều khiển PIC32. Các ngăn xếp được chia thành nhiều lớp, mỗi lớp truy cập dịch vụ từ một hoặc nhiều lớp trực tiếp dưới nó. Bộ thư viện TCP / IP Stack bao gồm các tính năng chính:

-Tối ưu hoá cho tất cả các PIC18, PIC24, dsPIC và PIC32 gia đình. Hỗ trợ giao thức: ARP, IP, ICMP, UDP, TCP, DHCP, SNMP, HTTP, FTP, TFTP

-Socket hỗ trợ cho TCP và UDP

-Secure Sockets Layer (SSL)

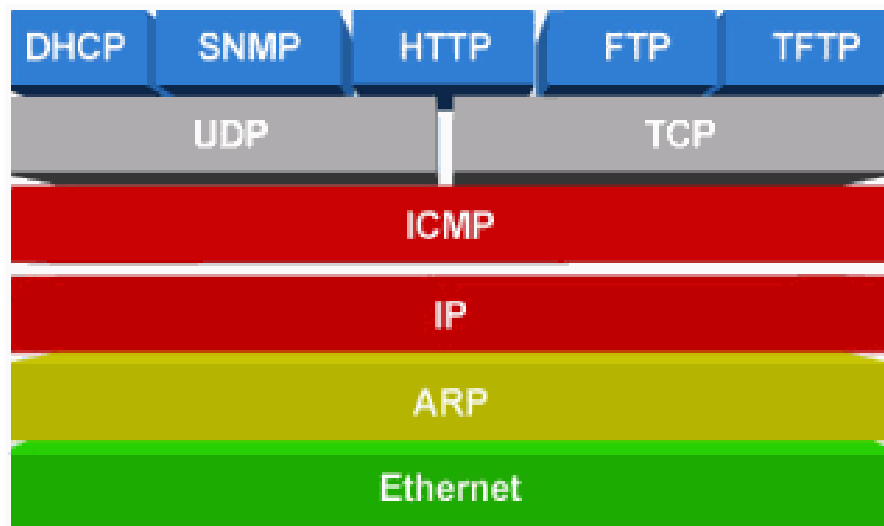
-NetBIOS vụ dịch vụ tên

-DNS - Domain Name System

-Ethernet Device

Discovery

-Hỗ trợ cho MPLAB C18, C30, C32 và trình biên dịch

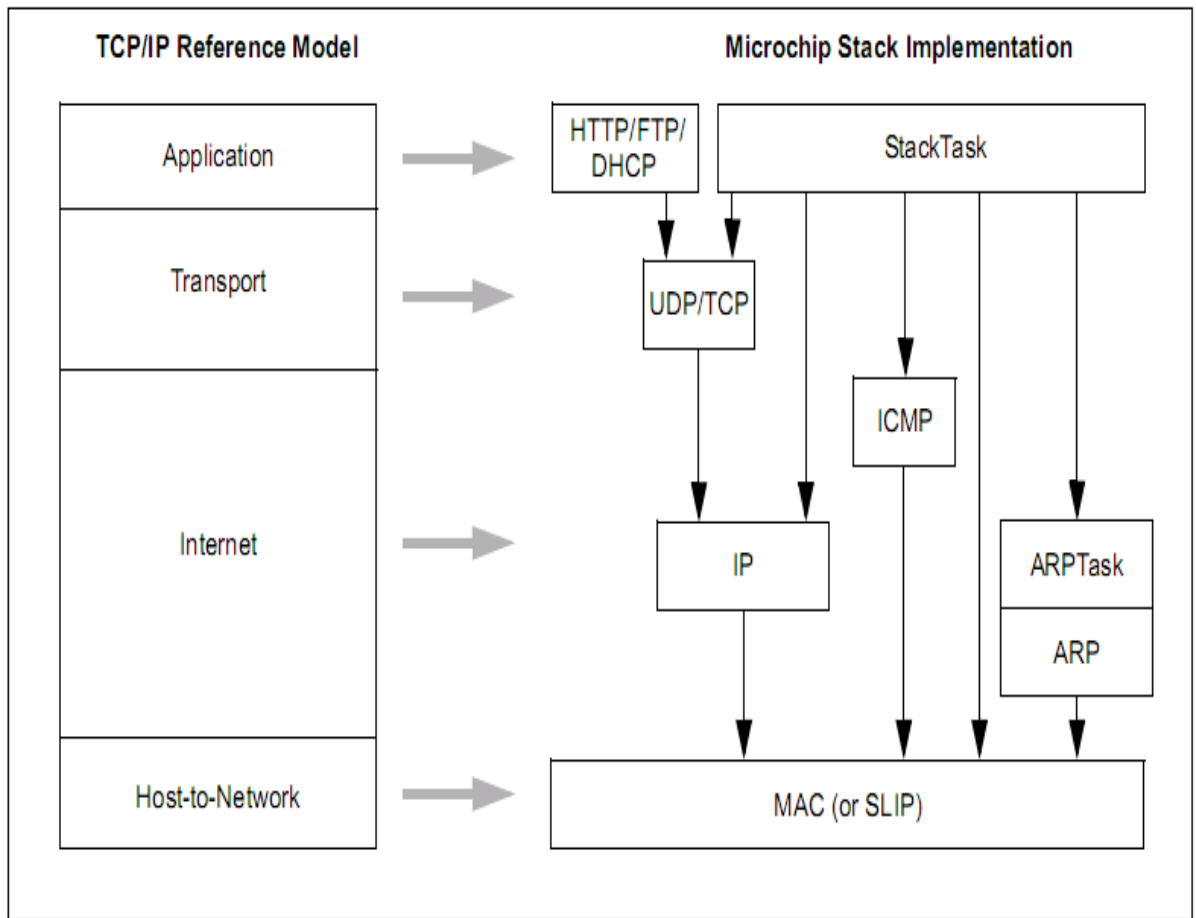


Hình 3.3.Sơ đồ bộ thư viện TCP/IP stack

Bộ thư viện TCP/IP stack được viết bằng của ngôn ngữ lập trình 'C. Toàn bộ code có dung lượng khoảng 28-34 KB, tùy thuộc vào mục đích được sử dụng, Bộ thư viện để lại nhiều không gian trên để người dùng có thể điều chỉnh cho phù hợp với ứng dụng của mình.

b) Mô hình tham chiếu TCP/IP stack

Giống như mô hình tham chiếu TCP / IP, TCP / IP Stack cũng được chia thành nhiều lớp (Hình 2). Mỗi lớp có 1 chức năng riêng, trong khi các dịch vụ và các API (Giao diện lập trình ứng dụng) được định nghĩa thông qua các tiêu đề, bao gồm các tập tin. Không giống mô hình TCP / IP, các lớp trong TCP / IP Stack trực tiếp truy cập vào một hoặc nhiều lớp mà không phụ thuộc vào lớp dưới nó. Một bổ sung lớn của TCP / IP Stack là thực hiện việc bổ sung hai modules mới: "StackTask" và "ARPTask". StackTask quản lý hoạt động của Stack và tất cả các module của nó, trong khi ARPTask quản lý các dịch vụ của các Địa chỉ Nghị quyết Protocol (ARP) lớp. Như đã đề cập trước đó, TCP / IP Stack là một bộ chương trình; một số lớp của nó phải có khả năng thực hiện một số hoạt động không đồng bộ. Để có thể đáp ứng được điều này đòi hỏi phải có tính tương đối độc lập khi sử dụng dịch vụ.



Hình 3.4. So sánh mô hình TCP/IP và mô hình TCP/IP Stack

c) Cấu hình

TCP/IP Stack là bộ chương trình được viết bằng ngôn ngữ “C”. Để bật, tắt hoặc thiết lập một tham số nào đó người sử dụng chỉ cần thay đổi một hoặc nhiều tham số trong các định nghĩa. Hầu hết các tham số này được định nghĩa trong tập tin header, "StackTsk.h". Một số định nghĩa được quy định tại các tập tin khác được hiển thị với tên tập tin tương ứng. Một khi tập tin này được sửa đổi, người sử dụng phải xây dựng lại các ứng dụng cho phù hợp với dự án của mình. Các định nghĩa được liệt kê trong Bảng sau

Bảng 3.1. Các định nghĩa

Định nghĩa	Giá trị	Sử dụng	Chức năng
CLOCK_FREQ (compiler.h)	Oscillator Frequency (Hz)	Tick.c	Xác định hệ thống dao động tần số
TICKS_PER_SECONDS	10-255	Tick.c	Để tính toán một lần thứ hai
TICK_PRESCALE_VALUE	2, 4, 8, 16, 32, 64, 128, 256	Tick.c	Để xác định giá trị đánh dấu vào truy cập
MPFS_USE_PGRM	N/A	MP File System (MPFS.c)	Sử dụng bộ nhớ chương trình để lưu trữ MPFS
MPFS_USE_EEPROM	N/A	MPFS.c	Sử dụng eeprom để lưu trữ MPFS
MPFS_RESERVE_BLOCK	0-255	MPFS.c	Số byte MPFS duo9c5 lưu trữ
EEPROM_CONTROL	External Data EEPROM Control Code	MPFS.c	Địa chỉ lưu vào eeprom
STACK_USE_ICMP	N/A	StackTsk. c	Yêu cầu sử dụng ICMP
STACK_USE_SLIP	N/A	SLIP.c	Yêu cầu sử dụng SLIP
STACK_USE_IP_GLEANING	N/A	StackTsk. c	Yêu cầu sử dụng IP
STACK_USE_DHCP	N/A	DHCP.c, StackTsk. c	Yêu cầu sử dụng dhcp
STACK_USE_FTP_SERVER	N/A	FTP.c	Yêu cầu sử dụng FTP

STACK_USE_TCP	N/A	TCP.c, StackTsk. c	Yêu cầu sử dụng TCP
STACK_USE_UDP	N/A	UDP.c, StackTsk. c	Yêu cầu sử dụng TCP
STACK_CLIENT_MODE	N/A	ARP.c, TCP.c	Kích hoạt client
TCP_NO_WAIT_FOR_ACK	N/A	TCP.c	TCP sẽ đợi ACK trước khi truyền
MY_DEFAULT_IP_ADDR_BYTE MY_DEFAULT_MASK_BYTE? MY_DEFAULT_GATE_BYTE? MY_DEFAULT_MAC_BYTE?	0-255	User Application	Xác định mặc định IP, MAC, gateway và subnet mask giá trị.
MY_IP_BYTE? MY_MASK_BYTE? MY_GATE_BYTE? MY_MAC_BYTE?	0-255	MAC.c, ARP.c, DHCP.c, User Application	IP, MAC, gateway và subnet mặt nạ giá trị như lưu / định nghĩa bởi ứng dụng. Nếu DHCP được kích hoạt, những hạn ảnh các giá trị hiện tại máy chủ DHCP giao cấu hình.
MAX_SOCKETS	1-253	TCP.c	Để xác định tổng số sockets hỗ trợ

			(giới hạn bởi bộ nhớ RAM có sẵn). kiểm tra thời gian thực hiện để đảm bảo rằng sockets là đủ để chọn TCP ứng dụng.
MAX_UDP_SOCKETS	201-1500	TCP.c, MAC.c	Xác định kích thước bộ đệm.
MAX_TX_BUFFER_COUNT	1-255	MAC.c	Xác định tổng số truyền qua bộ đệm. con số này được hạn chế bởi kích thước bộ đệm của Mac
MAX_HTTP_CONNECTIONS	1-255	HTTP.c	Xác định số lượng tối đa HTTP được kết nối
MPFS_WRITE_PAGE_SIZE (MPFS.h)	1-255	MPFS.c	Xác định kích thước có thể ghi trên MPFS
FTP_USER_NAME_LEN (FTP.h)	1-31	FTP.c	Xác định độ dài tối đa của FTP
MAX_HTTP_ARGS (HTTP.c)	1-31	HTTP.c	Xác định số tối đa của HTML
MAX_HTML_CMD_LEN (HTTP.c)	1-128	HTTP.c	Xác định độ dài tối đa của HTML

d) Sử dụng TCP/IP Stack

Bộ thư viện TCP/IP Stack chứa đầy đủ các Modem hỗ trợ như :HTTP, IP, FTP, DHCP. Tùy vào mục đích sử dụng mà người lập trình có thể thay đổi cho phù hợp với dự án của mình, nhưng một số tập tin nhất định phải có trong dự án để phù hợp với biên dịch. Các Module và tập tin đó được trình bày trong bảng sau:

Bảng 3.2. Các module và tập tin

Module	Tập tin bắt buộc	Mục đích
MAC	MAC.c Delay.c	Media Access Layer
SLIP	SLIP.c	Media Access Layer for SLIP
ARP	ARP.c ARPTsk.c MAC.c or SLIP.c Helpers.c	Address Resolution Protocol
IP	IP.c MAC.c or SLIP.c Helpers.c	Internet Protocol
ICMP	ICMP.c StackTsk.c IP.c MAC.c or SLIP.c Helpers.c	Internet Control Message Protocol
TCP	StackTsk.c UDP.c IP.c MAC.c or SLIP.c Helpers.c	Transmission Control Protocol

UDP	StackTsk.c UDP.c IP.c MAC.c or SLIP.c Helpers.c	User Datagram Protocol
Stack Manager	StackTsk.c TCP.c IP.c ICMP.c ARPTsk.c ARP.c MAC.c or SLIP.c Tick.c Helpers.c	Stack Manager (“StackTask”), which coordinates the other Microchip TCP/IP Stack modules
HTTP Server	HTTP.c TCP.c IP.c MAC.c or SLIP Helpers.c Tick.c MPFS.c XEEPROM.c(1)	HyperText Transfer Protocol Server
DHCP Client	DHCP.c UDP.c IP.c MAC.c Helpers.c Tick.c	Dynamic Host Configuration Protocol

IP Gleaning	StackTsk.c ARP.c ARPTsk.c ICMP.c MAC.c or SLIP.c	To configure node IP address only.
FTP Server	FTP.c TCP.c IP.c MAC.c or SLIP.c	File Transfer Protocol Server.

***Các lớp chức năng trong TCP/IP Stack**

TCP/IP Stack có nhiều lớp để sử dụng 1 lớp thì ta cần phải tìm hiểu ý nghĩa và chức năng của chúng.

***Media Access Control Layer (MAC)**

TCP/IP Stack bao gồm nhiều ứng dụng đã được cụ thể bằng văn bản để sử dụng giao diện mạng lưu ý ứng dụng này đã được cụ thể bằng văn bản để sử dụng giao diện mạng Realtek RTL8019AS Controller (NIC). RTL8019AS là một NE2000 tương thích NIC, mà được thực hiện cả trên vật lý Ethernet (PHY) và lớp MAC. Nếu một NIC khác được sử dụng, người sử dụng sẽ cần phải sửa đổi hoặc tạo một file MAC.c mới để chứa truy cập. Miễn là dịch vụ được cung cấp bởi MAC.c không thay đổi thì tất cả các module khác sẽ vẫn không thay đổi.

Ngoài việc quản lý bộ đệm FIFO cho NIC. Mac còn quản lý bộ đệm truyền nhận, nó có thể yêu cầu một tập tin nào đó được truyền lại. người sử dụng có thể xác định được kích thước của bộ đệm nhờ vào file Mac.c.

***Serial Line Internet Protocol (SLIP)**

Lớp SLIP sử dụng cáp nối tiếp như các giao tiếp phương tiện truyền thông, thay vì cáp ethernet. SLIP không đòi hỏi phải có NIC, do đó cung cấp rất đơn giản và kết nối không tốn kém IP. SLIP thường là một kết nối một-một, nơi

một máy chủ kết nối với một máy khách. Các SLIP module được thiết kế để hoạt động với một Windows ® dựa trên máy tính, mặc dù nó có thể được sửa đổi để làm việc với hệ điều hành khác với rất ít thay đổi. Các APIs được cung cấp bởi các module SLIP cũng được sử dụng giống như MAC

***Address Resolution Protocol**

Lớp ARP của TCP / IP Stack được thực hiện bởi hai mô lớp :ARP và ARPTask .ARP thực hiện bởi các tập tin "ARP.c", ARPTask thực hiện bởi các tập tin"ARPTsk.c" cung cấp hoàn thành dịch vụ ARP. ARPTask được thực hiện như một bộ máy tập hợp, đáp ứng yêu cầu ARP từ xa. Nó cũng duy trì một bộ nhớ cache một cấp để lưu trữ ARP trả lời và đưa một mức độ cao hơn cho phù hợp với các cuộc gọi. ARPTask là một cơ chế không thực hiện thử lại. Do đó, các lớp trên hoặc các ứng dụng này phải phát hiện ra điều kiện thời gian và trả lời phù hợp.

ARPTask hoạt động ở hai chế độ: chế độ Server và Server / Client. Trong chế độ Server Client , một phần của mã được kích hoạt và biên soạn để tạo ARP yêu cầu từ các máy chủ địa phương riêng của mình. Trong chế độ Server, cácARP mã yêu cầu không được biên dịch. Thông thường, nếu TCP/IP Stack được sử dụng với các ứng dụng máy chủ (như HTTP Serverhoặc phục vụ FTP) r, ARPTask nên được biên soạn trong Server hình thức để giảm kích thước mã.Trình biên dịch xác định bao gồm STACK_CLIENT_MODE phần mã khách hàng. Trong chế độ Server/ Client, ARPTask duy trì một bộ nhớ cache một cấp để lưu trữ các ARP trả lời của máy chủ từ xa. Khi Server / Client không được kích hoạt, bộ nhớ cache là không được định nghĩa và tương ứng với bộ nhớ RAM và bộ nhớ chương trình không được sử dụng..

***Internet Protocol (IP)**

Lớp IP của TCP / IP Stack được thực hiện bởi các tập tin "IP.c". Các phần đầu tập tin "IP.h"xác định các dịch vụ được cung cấp bởi lớp này .Trong

kiến trúc này, lớp IP là thụ động, nó không đáp ứng với các gói dữ liệu IP. Thay vào đó, lớp cao cấp sử dụng IP nguyên thủy và lấy các gói tin IP, giải mã nó và xử lý một cách thích hợp. Đặc điểm kỹ thuật IP này là yêu cầu các máy chủ địa phương tạo ra một gói tin nhận dạng duy nhất cho mỗi gói tin truyền qua nó, nhận diện cho phép máy chủ từ xa xác định các gói tin trùng lặp và loại bỏ chúng. Lớp IP của TCP / IP Stack duy trì một tư nhân 16-bit biến để theo dõi các gói tin định danh.

***Internet Control Message Protocol (ICMP)**

Lớp ICMP được thực hiện bởi các tập tin "ICMP.c". Các tập tin tiêu đề "ICMP.h" định nghĩa các dịch vụ được cung cấp bởi lớp này. Trong kiến trúc này, lớp ICMP là một lớp thụ động, nó không đáp ứng với các gói dữ liệu ICMP. Thay vào đó, lớp cao cấp sử dụng ICMP nguyên thủy và tìm nạp ICMP gói tin, giải thích nó và xử lý cho thích hợp.

Thông thường, ICMP được sử dụng để gửi và nhận lỗi hoặc thăm dò tin nhắn. Trong TCP / IP Stack, ICMP thực hiện nguyên thủy có thể được sử dụng để tạo ra bất kỳ thông điệp ICMP. Trong những ứng dụng, ICMP là hữu ích cho mục đích thăm dò. Khi được kích hoạt, ICMP có thể đáp ứng với "ping" các gói tin, do đó cho phép một máy chủ từ xa có thể xác định máy chủ địa phương.

Các vi mạch lớp ICMP chỉ đáp ứng dữ liệu ping các dữ liệu lên đến 32 byte; gói tin lớn hơn sẽ được bỏ qua. Nếu ta muốn xử lý các gói tin lớn hơn thì chỉ cần sửa đổi các trình biên dịch xác định MAX_ICMP_DATA_LEN (trong các tập tin tiêu đề "StackTsk.h").

***Transmission Control Protocol (TCP)**

Lớp TCP của TCP / IP Stack được thực hiện bởi các tập tin "TCP.c". Các phần đầu tập tin "TCP.h" xác định các dịch vụ được cung cấp bởi lớp này. Trong kiến trúc của TCP/IP Stack, TCP là một lớp đa hoạt

động. Nó có thể tìm TCP các gói dữ liệu và đáp ứng đến các máy chủ từ xa theo TCP của máy chủ quản lý.

"TCP.h" cung cấp dịch vụ socket TCP và giấu tất cả TCP gói xử lý từ người gọi. Lớp này cho phép 2-253 TCP socket, số lượng giới hạn bởi bộ nhớ và trình biên dịch được sử dụng. Với hơn một socket, các ứng dụng cấp cao hơn có thể duy trì đồng thời nhiều kết nối TCP và có thể có nhiều hơn một ứng dụng được sử dụng lớp này. Cơ sở này rất hữu ích khi HTTP Server được sử dụng. Mỗi socket tiêu thụ khoảng 36 bytes (kiểm tra tập tin nguồn cho thực tế tiêu thụ) và làm tăng thời gian. tổng thể chế biến TCP

Không giống như các giao thức TCP / IP , tất cả socket trong TCP / IP Stack chia sẻ một hoặc nhiều hơn một bộ đệm truyền. Cách tiếp cận này làm giảm tổng thể được yêu cầu RAM, nhưng nó có thể tạo ra một vấn đề khác, như tại một vài socket không nhận đủ bộ đệm truyền và không báo thối gia sử dụng các socket. Trong hoàn cảnh này các máy chủ và máy địa phương không thể liên lạc với nhau . Để tránh điều này, người dùng phải đảm bảo rằng có đủ bộ đệm truyền cho tất cả các socket.

Về việc nhận dữ liệu , chỉ có một bộ đệm được nhận . Nếu socket nhận được dữ liệu của nó, chủ sở hữu của socket đó phải tìm nạp và loại bỏ bộ đệm nhận trong một khoảng thời gian ngắn để các socket khác có thể nhận dữ liệu của họ. Khi nó phát hiện gói tin nó là quan tâm, nó phải xử lý các gói hoàn chỉnh trong một thời gian ngắn. Nếu trong khoảng thời gian đó mà vẫn không lấy được gói tin thì nó sẽ đợi và lấy gói tin sau đó.

Theo yêu cầu của TCP, mỗi phân đoạn TCP chứa một bộ kiểm tra bao gồm toàn bộ các giao thức TCP và cả khu vực dữ liệu. Để giảm bộ nhớ RAM, lớp TCP sử dụng trong bộ đệm MACNIC là lưu trữ và thực hiện các tính toán checksum MAC trong bộ đệm riêng của mình. Nếu NIC được sử dụng như là một MAC thì các SRAM NIC được sử dụng như một bộ đệm không gian. Nhưng nếu là SLIP được sử dụng như là một MAC thì dữ liệu được lưu vào Ram nội.

Các lớp TCP của TCP/IP Stack thực hiện hầu hết các phần TCP máy nước chủ đề xuất bởi RFC793. Nó cũng thực hiện tự động và thử lại hẹn giờ hoạt động, trong đó người dùng có thể kích hoạt hoặc vô hiệu hóa bởi các biên dịch `TCP_NO_WAIT_FOR_ACK` định nghĩa thời gian. Khi việc gửi lại tự động được kích hoạt, mỗi socket truyền đệm được dành riêng cho đến khi nhận được một sự trả lời từ máy chủ là nhận được. Thiết kế này tạo ra hiệu quả một truyền cửa sổ của một trong những phân đoạn TCP. Vì vậy, dữ liệu thông qua sẽ thấp hơn đáng kể hơn trong chế độ "Không gửi lại". Nếu chỉ có các HTTP Server được sử dụng, người sử dụng có thể vô hiệu hoá việc tự động gửi lại và tăng hiệu quả thông qua. Nếu việc áp dụng chính của logic yêu cầu mỗi gói sẽ được thừa nhận trước khi một cái mới có thể được truyền, người sử dụng nên cho phép chế độ "tự động gửi lại". Với tự động gửi lại được kích hoạt, một số các kết nối mở có thể không nhận phục vụ trên thời gian, và các máy chủ từ xa có thể được hưởng thời gian-out. Thiết lập lại hoặc sai sót.

***User Datagram Protocol (UDP)**

Lớp UDP của TCP / IP Stack được thực hiện bởi các tập tin "`UDP.c`". Các phần đầu tập tin "`UDP.h`" xác định các dịch vụ được cung cấp bởi lớp này. Trong kiến trúc của TCP / IP Stack, UDP là một lớp đa hoạt động. Nó có thể tìm UDP các gói dữ liệu và thông báo cho UDP socket tương ứng của dữ liệu đến hoặc truyền tải đi.

"`UDP.h`" cung cấp dịch vụ socket UDP và giấu tất cả Xử lý gói tin UDP từ người gọi. Lớp này cho phép lên đến 254 UDP socket (số lượng giới hạn bởi sẵn bộ nhớ và trình biên dịch được sử dụng). Với hơn một socket, các ứng dụng cấp cao hơn có thể duy trì Mul-tiple đồng thời kết nối UDP; có nhiều ứng dụng có thể được sử dụng lớp này. Mỗi socket tiêu thụ khoảng 19 byte (kiểm tra "`UDP.h`" tập tin để tiêu thụ thực tế) và tăng thời gian xử lý tổng thể UDP.

Không giống như các giao thức TCP / IP , tất cả socket trong TCP / IP Stack chia sẻ một hoặc nhiều hơn một bộ đệm truyền. Cách tiếp cận này làm giảm tổng thể được yêu cầu RAM, nhưng nó có thể tạo ra một vấn đề khác, như tại một vài socket không nhận đủ bộ đệm truyền và không báo thối gia sử dụng các socket. Trong hoàn cảnh này các máy chủ và máy địa phương không thể liên lạc với nhau . Để tránh điều này, người dùng phải đảm bảo rằng có đủ bộ đệm truyền cho tất cả các socket

***Dynamic Host Configuration Protocol (DHCP)**

Lớp DHCP của TCP / IP Stack được thực hiện bởi các tập tin "dhcp.c". Các phần đầu tập tin "dhcp.h" định nghĩa các dịch vụ được cung cấp bởi lớp này .DHCP là một lớp đang hoạt động mà chương trình phát sóng DHCP yêu cầu, tự động nhận được và giải mã DHCP phản ứng. Các tính năng chính của nó bao gồm:

- Cấu hình địa chỉ IP, địa chỉ gateway và subnet mask.
 - DHCP tự động cho thuê thời gian, đổi mới và quản lý việc cho thuê.
- Hoạt động hoàn toàn tự động mà người sử dụng không cần can thiệp.

Các module DHCP được thực hiện như một khối thống nhất có nhiệm vụ thực hiện các hoạt động tự động mà không có kiến thức về các ứng dụng chính. Các DHCP thực tế hội nhập và kiểm soát được thực hiện bởi các quản lý Stack; nó xử lý tất cả các hoạt động cần thiết của nó như một phần của tiêu chuẩn nhiệm vụ, bằng cách sử dụng API DHCP để kiểm soát các module của hành vi. Người dùng không cần biết về DHCP để sử dụng nó.

Một ứng dụng người dùng cũng có thể chọn để gọi một số các API để trực tiếp kiểm soát DHCP hoạt động, chẳng hạn như liệu DHCP được cấu hình hay không, và liệu DHCP có vĩnh viễn ngừng hoạt động. Thông thường, ứng dụng của người dùng không cần trực tiếp tương tác với DHCP .Để sử dụng module DHCP, người sử dụng phải sử dụng các tập tin sau:

1. Bỏ ghi chú "STACK_USE_DHCP" trong tiêu đề tập tin "StackTsk.h".

2. Phải có hai tập tin "dhcp.c" và "udp.c" trong dự án.

3. Tăng MAX_UDP_SOCKETS lên một (ít nhất một UDP socket phải có sẵn cho DHCP điều chỉnh số lượng socket dựa trên UDP và DHCP).

Khi DHCP được thực hiện, ứng dụng người dùng phải cố gắng truyền thông mạng cho đến khi DHCP được cấu hình đúng. Thông thường, nếu một ứng dụng người dùng có chứa các ứng dụng một hoặc nhiều khách hàng có yêu cầu thông tin liên lạc trên điện-up hoặc thiết lập lại, việc áp dụng phải kiểm tra xem DHCP được cấu hình trước khi truyền tải dữ liệu bằng cách sử dụng các mô-đun lớp thấp hơn. Điều này có thể được thực hiện với chức năng DHCPsBound.

Đặc điểm kỹ thuật chính thức DHCP (RFC1541) là đòi hỏi các khách hàng DHCP phải làm mới cấu hình IP trước khi cho thuê thời gian. Để theo dõi thời gian thuê, người sử dụng phải chắc chắn rằng TickUpdate () được gọi theo yêu cầu, và đó chính xác là thời gian hợp lý duy trì (xem mã nguồn tập tin "webservr.c" cho một ví dụ làm việc). Độ phân giải thời gian cần thiết là 15 phút, cộng hoặc trừ, có nghĩa là TickUpdate () có thể được gọi là một ưu tiên rất thấp. Đối với các module DHCP điều tự động cập nhật cấu hình địa chỉ và subnet mask, phải có ít nhất một DHCP server trên mạng. Đó là trách nhiệm của người sử dụng để thực hiện một số phương pháp cho "xuất bản" các cấu hình lại cho người sử dụng tiềm năng. Gồm có nhiều tùy chọn tự việc đọc thông tin đến việc hiển thị trên mỗi nút, để lưu trữ thông tin trong một máy chủ trung tâm DHCP ta chỉ cần cập nhật kết quả các giá trị của MY_IP_BYTE?, MY_GATE_BYTE? và MY_MASK_BYTE?

*** IP Gleaning for IP Address Configuration**

Như là một thay thế DHCP, TCP / IP Stack cũng thực hiện một phương pháp đơn giản, được gọi là IP Gleaning, từ xa thiết lập địa chỉ IP của TCP / IP Stack. Phương pháp này không phải là một giao thức Internet chuẩn, và

không có tương ứng với RFC. Gleaning IP chỉ cho phép địa chỉ IP được thiết lập. Đối với cấu hình IP hoàn tất, DHCP phải được sử dụng.

Gleaning IP không cần bất kỳ phần mềm trên module. Thay vào đó, nó sử dụng module ICMP ARP. Để sử dụng nó, tập tin "icmp.c" phải được khai báo, và trình biên dịch xác định `S TACK_USE_IP_GLEANING` ghi chú bên trong tập tin `StackTsh.h`

e) HTTP SERVER

HTTP Server kèm theo ứng dụng được thực hiện như một khối thống nhất tồn tại với các vi mạch TCP / IP Stack và ứng dụng chính của người dùng. Các HTTP Server chính nó được thực hiện tại tập tin "HTTP.c", với một ứng dụng người dùng thực hiện hai chức năng. Các ứng dụng mã nguồn này nằm trong tập tin "Websrvr.c" nên được sử dụng như là một mẫu ứng dụng để tạo ra các giao diện cần thiết.

HTTP Server được cung cấp ở đây không thực hiện tất cả chức năng HTTP, nó là một máy chủ tối thiểu cho các mục tiêu của hệ thống nhúng. Người sử dụng có thể dễ dàng thêm mới chức năng theo yêu cầu và mục đích sử dụng.

HTTP server thích hợp những chức năng sau:

- Hỗ trợ nhiều kết nối HTTP.
- Hỗ trợ các trang web đặt tại một trong hai nội bộ chương trình hoặc bộ nhớ EEPROM nối tiếp bên ngoài.
- Bao gồm một chương trình dựa trên PC để tạo ra MPFS hình ảnh từ một thư mục nhất định.
- Hỗ trợ các phương thức HTTP "GET" (các phương pháp khác có thể dễ dàng thêm vào).
- Hỗ trợ một đối lần Common Gateway Interface (CGI) để gọi chức năng xác định trước từ bên trong trình duyệt từ xa.
- Hỗ trợ nội dung trang web năng động thể hệ

Các máy chủ bao gồm các phần chính sau:

- MPFS Builder Image.
- Thư viện MPFS truy cập.
- MPFS Download thường (thực hiện bởi chính của ứng dụng)
- HTTP Server Task.

Để sử dụng module DHCP, người sử dụng phải sử dụng các tập tin sau:

1. Bỏ ghi chú "STACK_USE_HTTP_SERVER" trong tiêu đề tập tin "StackTsk.h".
2. Đặt ghi chú này MAX_HTTP_CONNECTIONS trong tập tin "StackTsk.h".
3. Phải có hai tập tin "http.c" và "mpfs.c" trong dự án.
4. Tùy theo nơi lưu trữ trang web mà có thể bỏ ghi chú MPFS_USE_PGRM hoặc MPFS_USE_EEPROM. Nếu dữ liệu vào EEPROM ngoài thì sử dụng thêm tập tin "XEEPROM.c".

Các HTTP Server sử dụng file "index.htm" như Trang web mặc định. Nếu trình duyệt (một khách hàng từ xa) truy cập các HTTP Server bằng địa chỉ IP hoặc tên miền thì "index.htm" là trang mặc định phục vụ. Điều này đòi hỏi rằng tất cả các ứng dụng bao gồm một tập tin có tên "Index.htm" phải được lưu trong MPFS. Nếu cần thiết, tên của tập tin này có thể thay đổi mặc định bằng cách sửa đổi định nghĩa của trình biên dịch HTTP_DEFAULT_FILE_STRING trong tập tin "http.c". Chú ý trong các trang web có thể được chứa các ký tự sau:

- Một hoặc hai dấu ngoặc kép ("và")
- Trái hoặc dấu ngoặc vuông góc (<và>)
- Dấu thăng (#)
- Những dấu phần trăm (%)
- Trái hoặc bên phải khung hoặc niềng răng ([, {, } và))
- Ổng "" biểu tượng (|)
- Các dấu gạch chéo ngược (\)

-Dấu sót (^)

-Các dấu ngã (~)

HTTP cũng duy trì một danh sách các loại tập tin mà nó hỗ trợ. Nó sử dụng thông tin này để tư vấn cho một từ xa trình duyệt trên làm thế nào để giải thích một tập tin cụ thể, dựa trên của tập tin mở rộng ba ký tự. Theo mặc định,Vi mạch HTTP Server hỗ trợ ". Txt", ". Htm","Gif". "Cgi"., ". Jpg", "CLA". Và ". Wav" files. Nếu một ứng dụng sử dụng các loại tập tin mà không có trong này danh sách, người sử dụng có thể sửa đổi các bảng "httpFiles", cùng với tương ứng"httpContents" trong tập tin "http.c".

f)FTP SERVER

FTP Server có nhiệm vụ hợp tác với TCP / IP Stack để tạo ra các ứng dụng chính của người dùng. FTP Server được thực hiện tại tập tin "FTP.c".

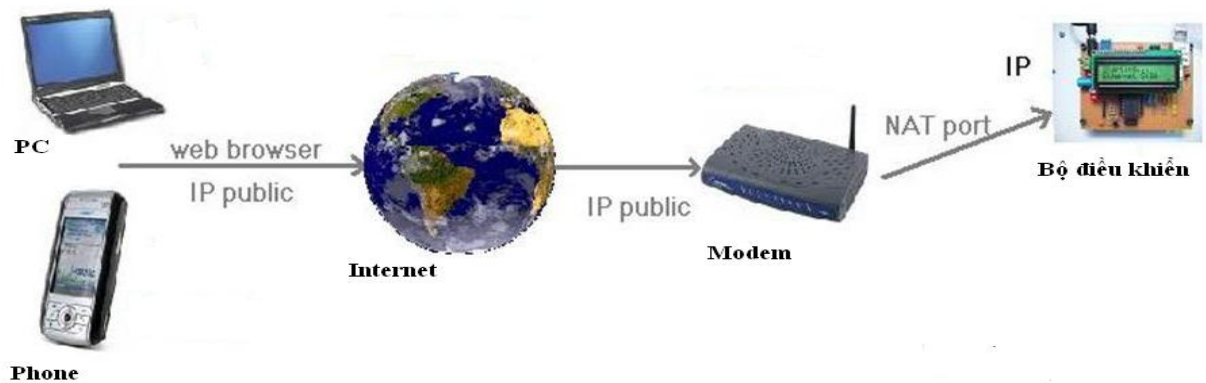
FTP server có những tính năng sau:

- Một kết nối FTP cho chúng ta biết có một người dùng đang sử dụng ứng dụng đó.
- Tự động tương tác với tập tin hệ thống (MPFS).
- Upload tap tin cá nhân hoặc hỗ trợ.

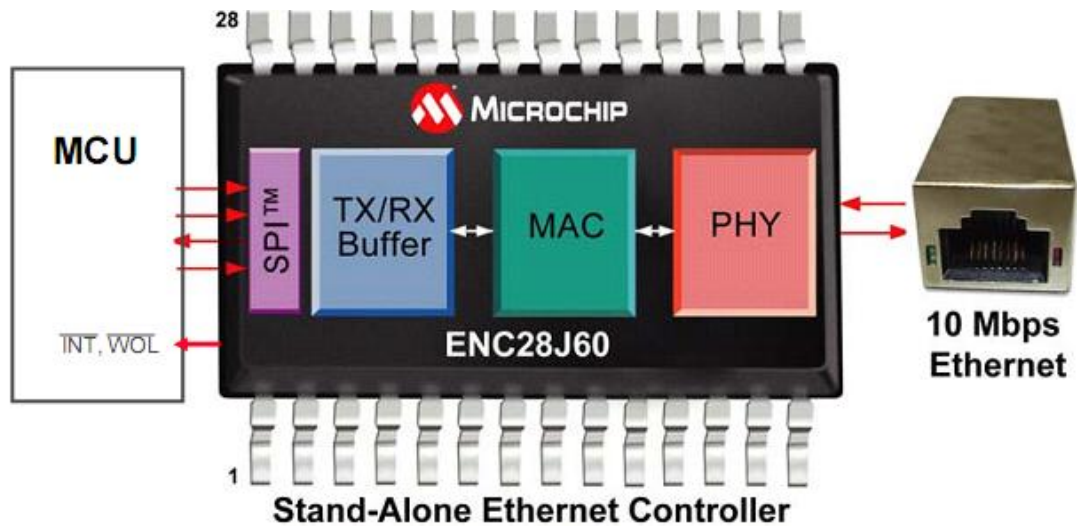
Để sử dụng module DHCP, người sử dụng phải sử dụng các tap tin sau:

1. Bỏ ghi chú "STACK_USE_FTP_SERVER" trong tiêu đề tập tin "StackTsk.h".
2. Phải có hai tập tin"FTP.c " và "mpfs.c" trong dự án.
3. Tùy theo nơi lưu trữ trang web mà có thể bỏ ghi chú MPFS_USE_PGRM hoặc MPFS_USE_EEPROM. Nếu dữ liệu vào EEPROM ngoài thì sử dụng thêm tập tin"XEEPROM.c".

3.2.6. Sơ đồ kết nối với các thiết bị ngoại vi

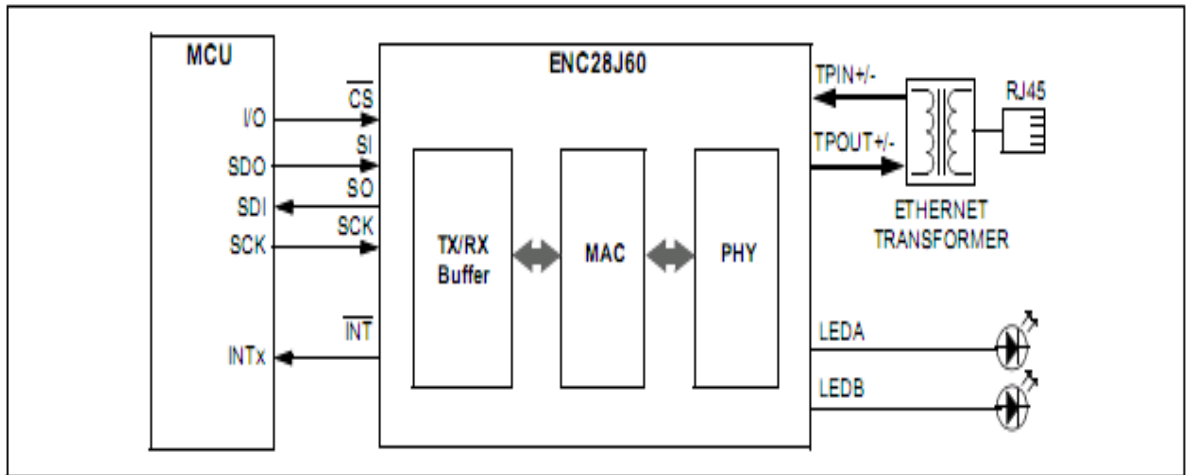


Hình 3.4. Mô hình hệ thống điều khiển thông qua mạng Internet.



Hình 3.5. Sơ đồ vi mạch giao tiếp ethernet ENC28J60

Vi mạch ENC28J60 giao tiếp với các vi xử lý khác thông qua chuẩn truyền dữ liệu nối tiếp SPI (Serial Peripheral Interface), đây là chuẩn giao tiếp rất phổ biến, được dùng để nối các vi mạch trong cùng một hệ thống với ưu điểm là chạy nhanh và tốn rất ít dây nối tín hiệu, chỉ cần 3 dây cho cả đường ghi và đọc, đó là:



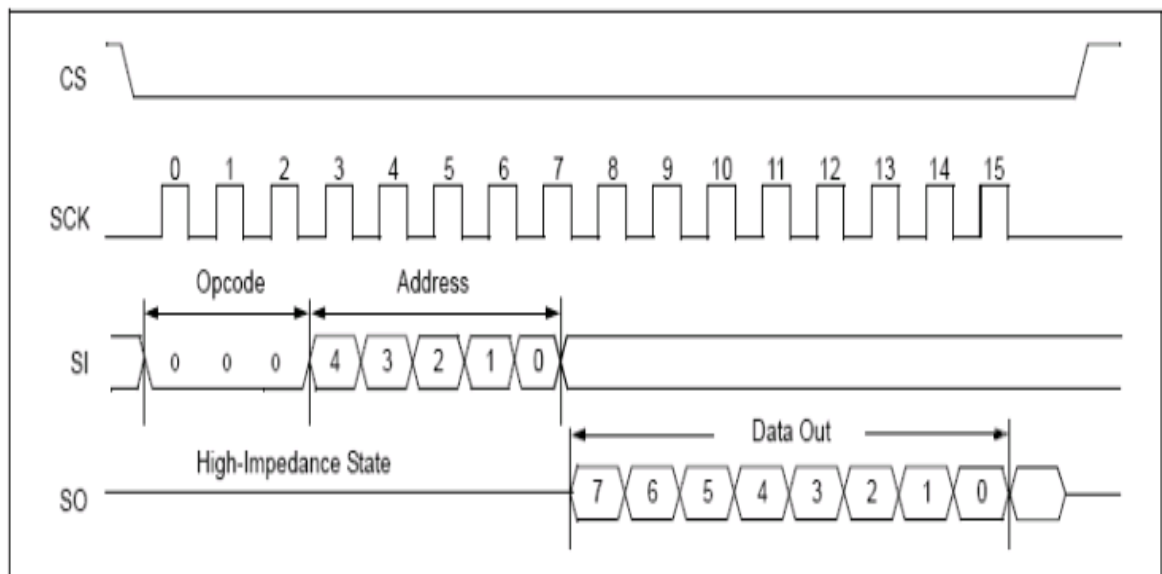
Hình 3.6. Sơ đồ ghép nối ENC28J60 với vi xử lý

SCK Serial Clock – Xung đồng bộ cho đường nối tiếp.

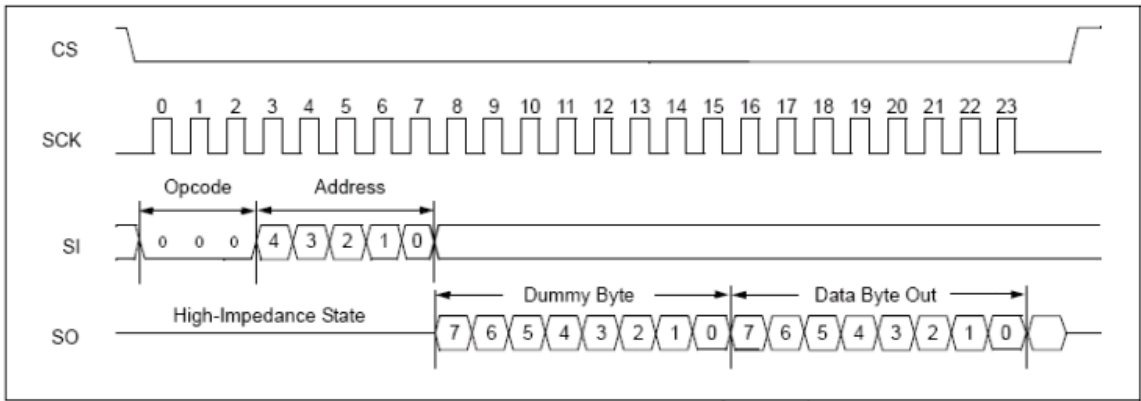
SI Serial Input – Tín hiệu nối tiếp vào (ghi).

SO Serial Output – Tín hiệu nối tiếp ra (đọc).

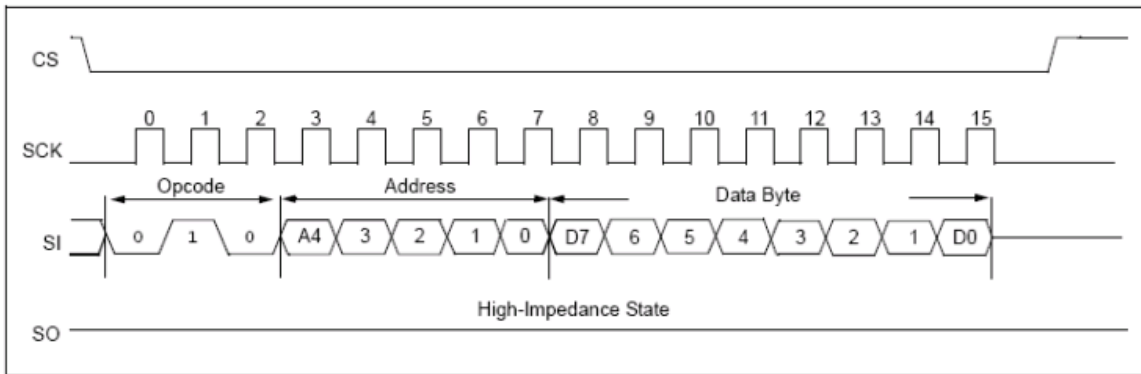
Sau đây là bộ lệnh SPI dùng cho vi mạch ENC28J60 và các giản đồ xung thể hiện các quá trình ghi, đọc dữ liệu, lệnh giữa vi xử lý và ENC28J60.



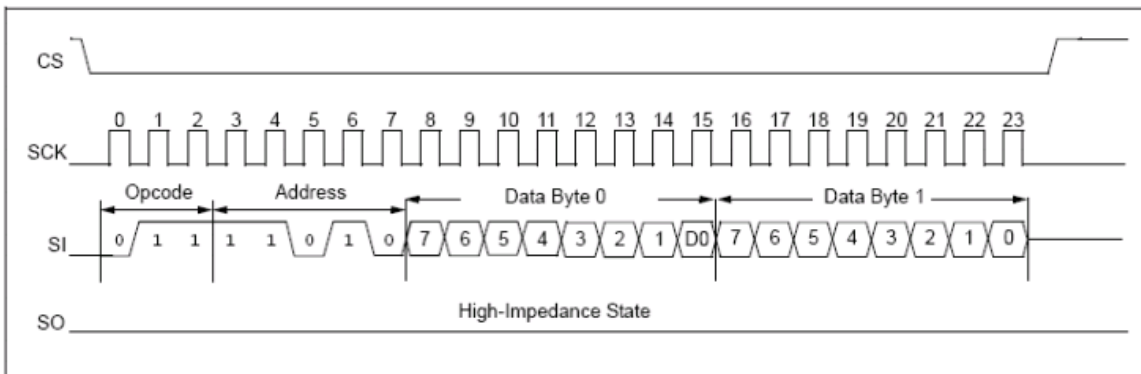
Hình 3.7. Quá trình đọc thanh ghi điều khiển Ethernet



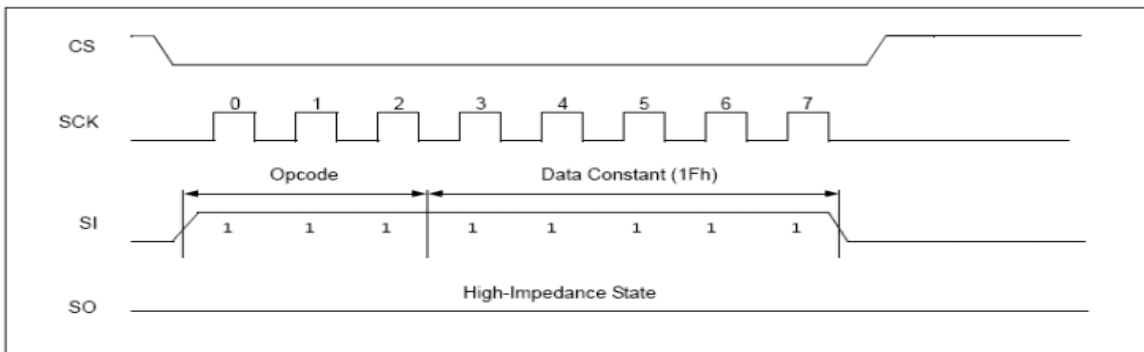
Hình 3.8. Quá trình đọc thanh ghi điều khiển MAC



Hình 3.9. Quá trình ghi vào thanh ghi lệnh



Hình 3.10. Quá trình ghi vào bộ đếm lệnh



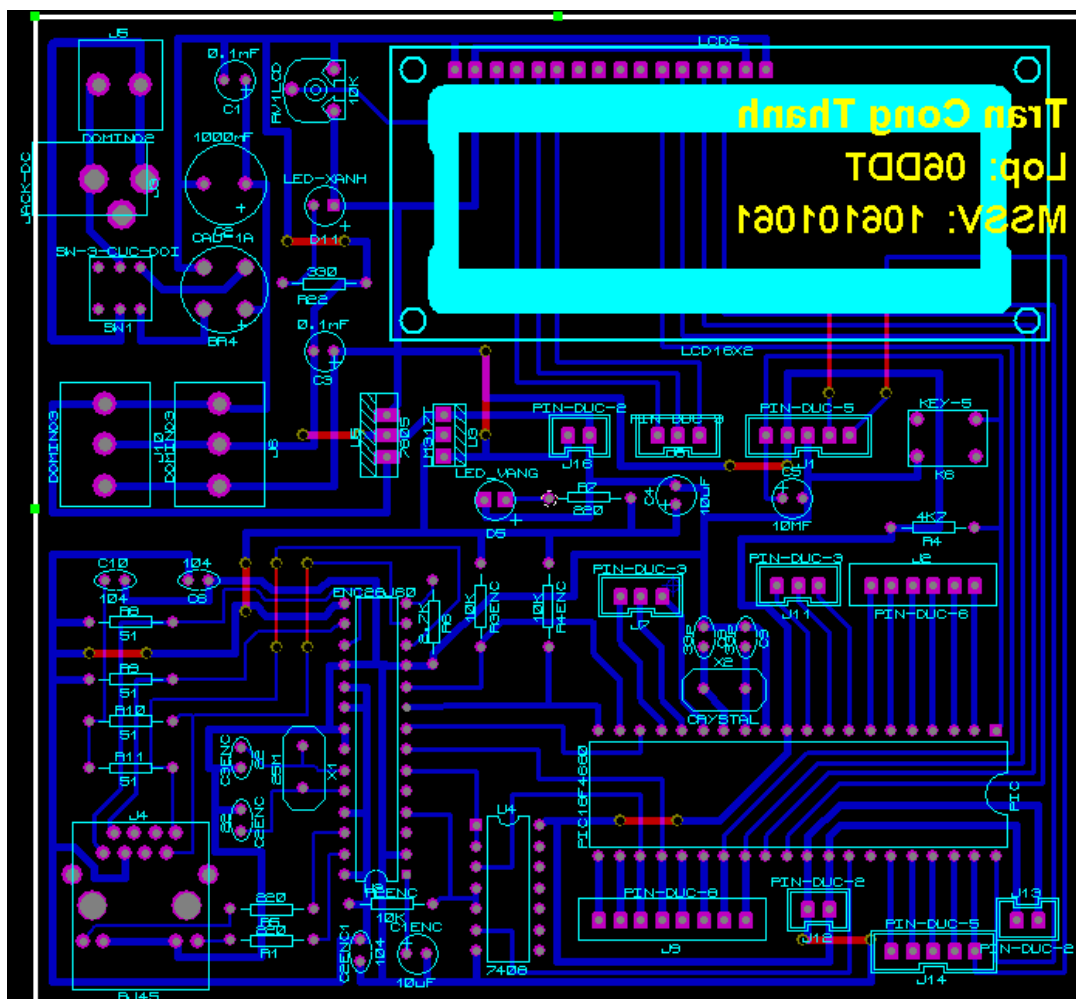
Hình 3.11. Quá trình ghi vào lệnh của hệ thống

Mặc dù đã được thiết kế theo cấu trúc “stand alone” (một chip làm hết các việc) nhưng việc điều khiển đề vi mạch này hoạt động được là rất phức tạp và đòi hỏi có hiểu biết khá toàn diện về vi điều khiển, mạch điện tử và mạng Internet. Tuy nhiên, rất may mắn cho người thiết kế và lập trình vì hãng MicroChip (hãng sản xuất vi mạch ENC28J60) đã hỗ trợ tối đa, bằng cách đưa ra đầy đủ các thư viện phục vụ cho việc lập trình, làm cho việc lập trình giao tiếp với vi mạch này trở nên khá dễ dàng.

3.3.SƠ ĐỒ NGUYÊN LÝ

3.3.1.Sơ đồ nguyên lý chung

Sau khi nghiên cứu thiết kế mạch đo nhiệt độ dùng LM35, đồng thời gửi tín hiệu nhiệt độ lên internet nhờ IC ENC28J60 với vi điều khiển trung tâm là PIC18F4680 ta có sơ đồ nguyên lý chung như hình 3.12.



Hình 3.12. Sơ đồ mạch in

3.3.2. Nguyên lý hoạt động

- Toàn bộ mạch sẽ được reset khi ta đưa chân MCLR của vi điều khiển xuống mức thấp.

- Vi điều khiển PIC 18F4680 hoạt động với dao động ở tần số 40MHz sử dụng thạch anh dao động nối vào các chân OSC1 và OSC2 để tạo dao động.

*Đo nhiệt độ :

- Đầu ra Vout của cảm biến nhiệt độ LM35 được nối với ngõ vào analog RA0 của PIC18F4680, vđk này có 11 ngõ vào analog (RA0,RA1 ,RA2,RA3 ,RA4,RA5 ;RE2,RE0 và RB0,RB1,RB7),tất cả các ngõ vào analog này đều được nối với đầu vào của bộ chuyển đổi ADC 10 bit tích hợp sẵn trong vđk.

- Khi nhiệt độ môi trường thay đổi ± 1 làm cho trở kháng của cảm biến LM35 thay đổi dẫn đến điện áp đầu vào V_{in} của ADC thay đổi. Điện áp V_{in} vào ADC sẽ được so sánh với U_d của ADC. U_d có thể thay đổi từ 0V đến $2(V_{ref}/2)$.Hiệu điện thế chuẩn V_{ref} có thể được lựa chọn là VDD,VSS hay hiệu điện thế chuẩn được xác lập trên 2 chân RA2 và RA3 tùy thuộc vào chương trình.Giả sử ban đầu $U_d = 0$, nếu $V_{in} > U_d$ khi đó U_d sẽ được cộng thêm một giá trị là $\Delta U = 10\text{mv}$; $U_d = U_d + \Delta U$ đồng thời giá trị bộ đếm tăng thêm 1. Quá trình so sánh cứ như vậy đến khi nào $U_d = V_{in}$ thì dừng. Khi đó giá trị của bộ đếm chính là giá trị thập phân. Giá trị thập phân này sẽ được đưa qua một bộ giải mã, giải mã ra nhị phân rồi đưa ra đầu ra 10 bit của bộ ADC.

- Kết quả đo nhiệt độ trên, có thể được hiển thị trên máy tính thông qua chuẩn giao tiếp nối tiếp USART giữa vi điều khiển và cổng COM của máy tính thông qua IC MAX 232.

*Giao tiếp mạng :

- Điện áp từ khối nguồn qua mạch sử dụng ic LM317T cho điện áp đầu ra 3V cấp nguồn cho ic ENC28J60 và toàn bộ module giao tiếp mạng.

- IC ENC28J60 hoạt động với dao động ở tần số 25MHz sử dụng thạch anh dao động nối vào các chân OSC1 và OSC2 để tạo dao động.

- ENC28J60 hoạt động khi

- Vi điều khiển PIC18F4680 giao tiếp với ic ENC28J60 thông qua giao diện SPI (giao tiếp nối tiếp đồng bộ) kiểu chủ - tớ ,với vi điều khiển đóng vai trò là chủ.Các chân SCK,SI,SO,CS của ic ENC28J60 được nối lần lượt với RC3/SCK ,RC5/SD0,RC4/SDA,CS (chân CS được đưa xuống mức thấp).

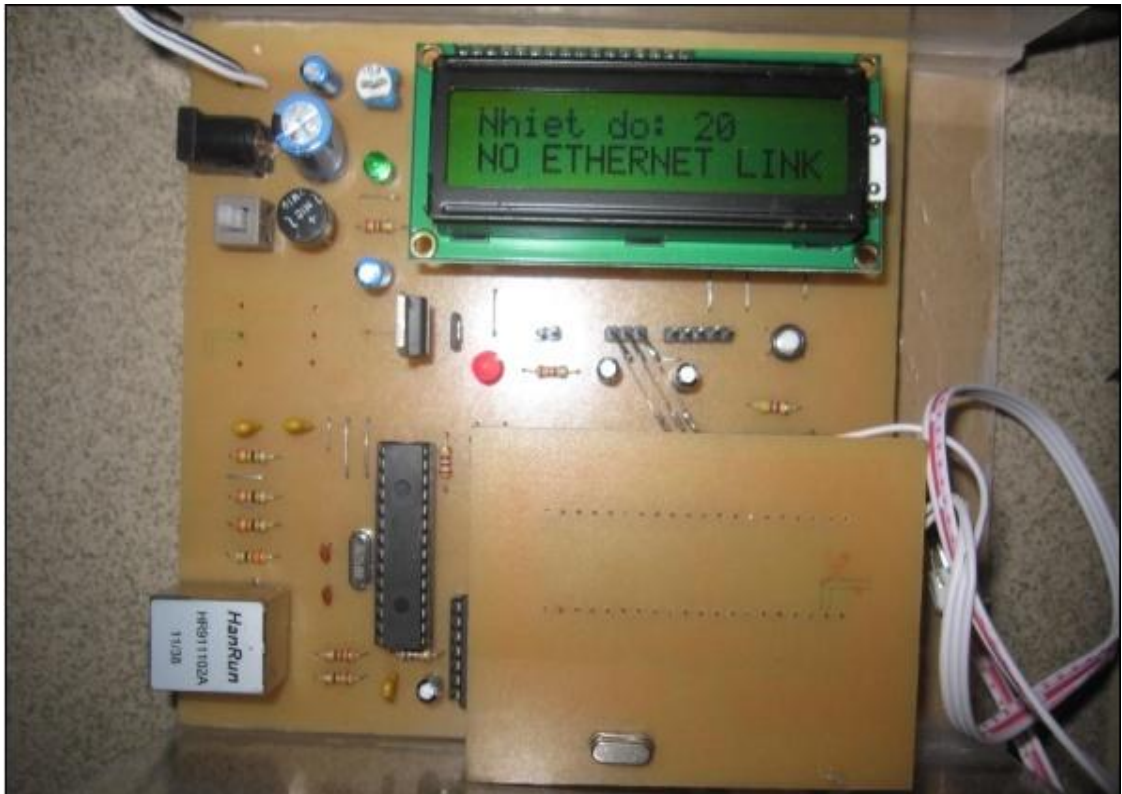
- SCK : là xung giữ nhịp cho giao tiếp SPI,mỗi nhịp trên chân SCK sẽ báo 1 bit được nhận hay gửi.Hai chân còn lại đóng vai trò nhận và gửi dữ liệu.

- Module giao tiếp mạng giao tiếp với các thiết bị ngoại vi (ở đây là modem) thông qua chân TPIN+ , TPIN- ,TPOUT+, TPOUT-.

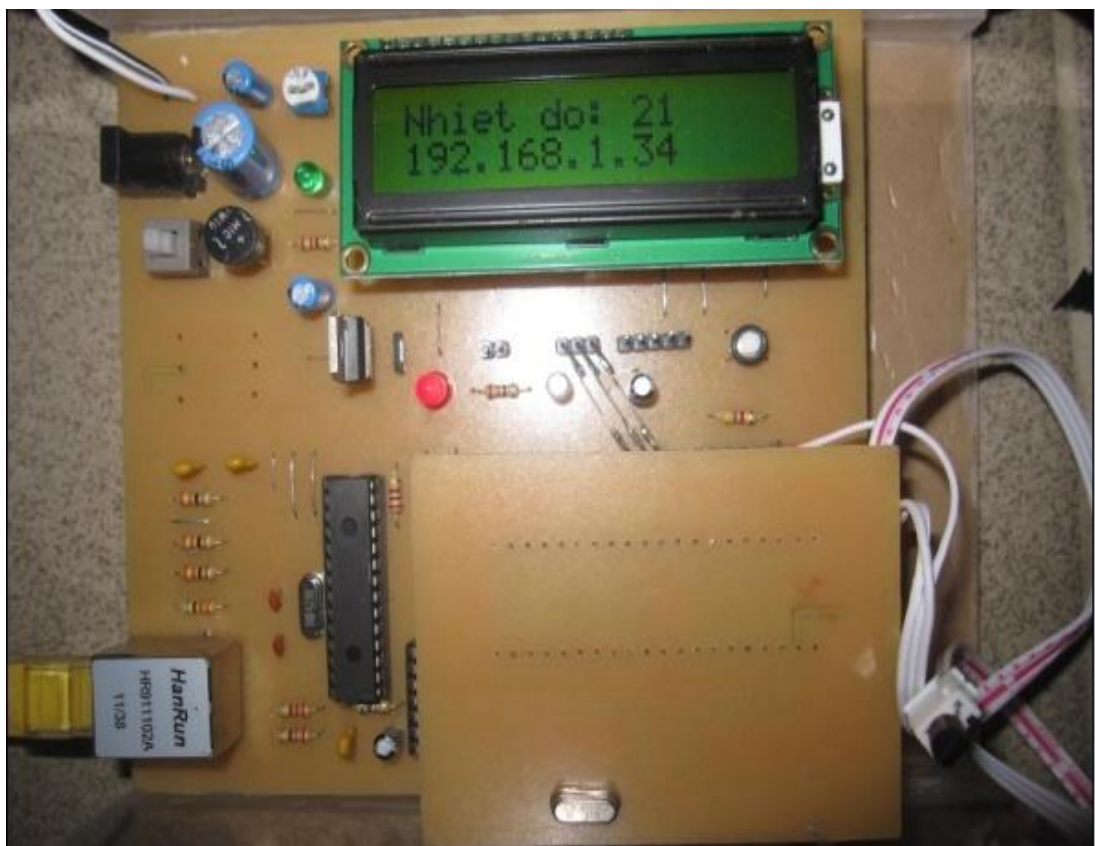
- LedA, LedB: có chức năng hỗ trợ tự động phát hiện và Reset.

-Khi khởi động khối giao tiếp mạng, vi điều khiển 18F4680 sẽ đọc cấu hình từ trong bộ nhớ, cấu hình này được load từ các thư viện. Cấu hình này sẽ kiểm tra xem có dữ liệu SPI gửi từ IC ENC28J60, các dữ liệu đó là địa chỉ MAC, IP. Sau khi đã cấu hình xong, vi điều khiển sẽ gửi tín hiệu nhiệt độ tới IC

IC ENC28J60 theo giao diện SPI, IC này lại gửi tín hiệu vừa nhận được lên webserver thông qua modem mạng.



Hình 3.13.Module khi chưa cắm dây mạng



Hình 3.14.Module khi đã kết nối mạng

KẾT LUẬN

Trong thời gian nghiên cứu và thiết kế được sự giúp đỡ tận tình của Th.s Trần Thị Phương Thảo em đã thu được nhiều kiến thức về cảm biến nhiệt độ, cấu trúc , cấu tạo của vi điều khiển , ứng dụng vào thực tế, hoàn thành cơ bản yêu cầu đề án. Mạch thi công xong hiện thi được nhiệt độ , hiển thị địa chỉ IP kết nối với internet. Tuy nhiên vẫn còn những khuyết điểm như sai số, thi thoảng nhảy giá trị, sử dụng mang day vì vậy hướn phát triển của đề tài kết nối với internet qua wifi. Tuy nhiên do thời gian có hạn cũng như trình độ của bản thân còn nhiều hạn chế nên đề tài thực hiện còn nhiều thiếu sót em rất mong nhận được sự chỉ bảo, sửa chữa đóng góp ý kiến của các thầy cô giáo, các bạn trong lớp để em có thể thực hiện và hoàn thành đề tài được tốt hơn.

Một lần nữa em xin chân thành cảm ơn sự chỉ bảo, hướng dẫn tận tình của Th.S Trần Thị Phương Thảo, các thầy cô trong khoa, các bạn bè trong lớp đã giúp đỡ em trong quá trình thực hiện đề tài.

Em xin chân thành cảm ơn!

Hải Phòng, ngày 25 tháng 11 năm 2012

Sinh viên

Nguyễn Văn Hiệp

TÀI LIỆU THAM KHẢO

1. BÀI GIẢNG CẢM BIẾN NHIỆT ĐỘ GIÁO TRÌNH CẢM BIẾN CÔNG NGHIỆP Hoàng Minh Công (nhà xuất bản xây dựng năm 2007)
2. CÁC VI ĐIỀU KHIỂN PIC TAP2 (nhà xuất bản khoa học và kỹ thuật)
3. GIÁO TRÌNH HOÀN TRỈNH VI ĐIỀU KHIỂN PIC 16
4. WWW.DIENTUVIETNAM.NET
5. WWW.LINHKIENDIENTU.COM
6. TAILIEU.VN