BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

ĐỒ ÁN TỐT NGHIỆP

NGÀNH ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP

Sinh viên : Đặng Văn Thoan

Giảng viên hướng dẫn: ThS. Đỗ Anh Dũng

Hải Phòng -2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

THIẾT KẾ HỆ THỐNG VI ĐIỀU KHIỂN PIC 16F877A ĐO NHIỆT ĐỘ DÙNG CẢM BIẾN BÁN DẪN

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

Sinh viên thực hiện: Đặng Văn Thoan
Giảng viên hướng dẫn: ThS Đỗ Anh Dũng

Hải Phòng - 2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Đặng Văn Thoan - **MSV** : 2103102013

Lớp : DCL 2401

Ngành: Điện Tự Động Công Nghiệp

Tên đề tài : **Thiết kế hệ thống Vi điều khiển PIC 16F877A đo nhiệt độ dùng cảm biến bán dẫn**

NHIỆM VỤ ĐỀ TÀI

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

……
……
……
……
……
……

2. Các số liệu cần thiết để tính toán.

……
……
……
……
……
……
……
……

3.Địa điểm thực tập tốt nghiệp.

……

CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên :

Học hàm, học vị :

Cơ quan công tác : Trường Đại học quản lý và công nghệ Hải Phòng

Nội dung hướng dẫn:

…….....

…….....

…….....

Đề tài tốt nghiệp được giao ngày 20 tháng 6 năm 2022

Yêu cầu phải hoàn thành xong trước ngày 10 tháng 9 năm 2022

Đã nhận nhiệm vụ ĐTTN	Đã giao nhiệm vụ ĐTTN
Sinh viên	*Giảng viên hướng dẫn*

Hải Phòng, ngày tháng năm 2022

TRƯỞNG KHOA

TS. Đoàn Hữu Chức

Cộng hòa xã hội chủ nghĩa Việt Nam

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên: ThS Đỗ Anh Dũng

Đơn vị công tác: Trường Đại Học Quản lý và Công nghệ Hải Phòng

Họ và tên sinh viên: Đặng Văn Thoan

Chuyên ngành: Điện Tự Động Công Nghiệp

Nội dung hướng dẫn : Toàn bộ đề tài

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

.....

.....

.....

.....

2. Đánh giá chất lượng của đồ án/khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận, thực tiễn, tính toán số liệu...)

.....

.....

.....

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Được bảo vệ ☐ Không được bảo vệ ☐ Điểm hướng dẫn ☐

Hải Phòng, ngày.....tháng.....năm 2022

Giảng viên hướng dẫn

(ký và ghi rõ họ tên)

Cộng hòa xã hội chủ nghĩa Việt Nam

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên ……

Đơn vị công tác:.....

Họ và tên sinh viên: .....Chuyên ngành:.....

Đề tài tốt nghiệp: .....

.....

1. Phần nhận xét của giảng viên chấm phản biện

.....

.....

.....

.....

2. Những mặt còn hạn chế

.....

.....

.....

.....

3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ ☐ Không được bảo vệ ☐ Điểm phản biện ☐

Hải Phòng, ngày.....tháng.....năm 2022

Giảng viên chấm phản biện

(ký và ghi rõ họ tên)

MỤC LỤC

Nội dung	Số trang
Chương 1. Cảm biến nhiệt độ	1
1.1 Khái niệm	1
1.2 Cảm biến nhiệt điện trở	1
a. Cảm biến nhiệt điện trở kim loại	1
b. Cảm biến nhiệt điện trở bán dẫn	2
Chương 2. Vi điều khiển PIC 16F877A	4
2.1 Giới thiệu Vi điều khiển PIC	
a. Tổng quan	
b. Sơ đồ khối vi điều khiển pic16f877a	6
2.2 Đặc điểm biến đổi ADC	7
2.3 Phần mềm thực hiện chương trình nạp cho PIC.	9
2.3.1 Chương trình dịch CCS	
a) Giới thiệu về CCS .	10
b) Tạo một PROJECT sử dụng PIC Wizard	
c) Tab General	11
d) Tab SPI and LCD	13
e) Tab Timer	14
f) Tab Analog	15
g) Tab Interrupts và Tab Driver	18
Chương 3. Thiết kế hệ thống đo nhiệt độ dùng cảm biến LM335	20
3.1 Thiết kế phần cứng khối xử lý tín hiệu	21
3.2. Thiết kế phần cứng giao tiếp LM335 và Vi điều khiển PIC16F877A	22
3.3. Khối hiển thị	23
3.4 Khối nguồn	25
3.5 Sơ đồ nguyên lý và mạch in hệ thống đo	26
2.6 Lưu đồ thuật toán	27
Kết luận	28
Lời cảm ơn	29
Phụ lục 1	30
Phụ lục 2	36
Phụ lục 3	39
Tài liệu tham khảo	41

Chương 1. Cảm biến nhiệt độ

1.1 Khái niệm

Nhiệt độ là đại lượng vật lý được quan tâm nhiều nhất vì nó đóng vai trò quyết định đến nhiều tính chất của vật chất.

Để đo nhiệt độ trong hệ thống tự động có nhiều biện pháp khác nhau. Trên cơ sở đó người ta sử dụng các bộ cảm biến nhiệt độ với nguyên lý làm việc khác nhau. VD: nhiệt điện trở, nhiệt ngẫu, quang…

Thang nhiệt độ:

Được xác định từ các định luật nhiệt động .

Thang nhiệt độ nhiệt động tuyệt đối: Thang Kenvin (0K) là nhiệt độ cân bằng của điểm cân bằng 3 trạng thái nước, nước đá và hơi.

$$T(^0K) = t(^0C) + 273.15 \qquad (1\text{-}1)$$

Thang Celcius : thang nhiệt độ bách phân (0C)

Thang Farenheit :

$$T(^0C) = \left[T(^0F) - 32\right]\frac{5}{9} \qquad (1\text{-}2)$$

$$T(^0F) = \frac{9}{5}.T(^0C) + 32$$

1.2 Cảm biến nhiệt điện trở

Cảm biến nhiệt điện trở là cảm biến có điện trở biến đổi theo nhiệt độ

Kim loại điện trở biến đổi theo nhiệt độ, thể hiện qua α (hệ số nhiệt điện trở)

Phân loại: 3 loại

+ Cảm biến nhiệt điện trở kim loại

+ Cảm biến nhiệt điện trở bán dẫn

+ Nhiệt điện trở

a. Cảm biến nhiệt điện trở kim loại:

- Dây kim loại : gồm một sợi dây kim loại được dán trên bìa cách điện. Vật liệu thường dùng là Pt, Ni, W, Cu.

Khoảng nhiệt độ đo được: Pt (2000C ÷ 12000C)

Ni (-1900C ÷ 2500C)

Cu (-500C ÷ 1800C)

Khi nhiệt độ θ tăng thì dẫn đến R tăng theo. Qua R đo được ta xác định được nhiệt độ qua công thức:

$$R_\theta = R_0(1+\alpha\theta) \qquad (1\text{-}3)$$

Để cảm biến có độ nhạy cao ta phải chọn kim loại có điện trở suất (ρ) lớn $R = \rho.\frac{l}{q}$ khi R tăng thì l tăng, q giảm

Điện trở R càng lớn thì độ nhạy càng cao và dải đo càng hẹp.

Bảng 1.1 Tính chất của một số kim loại

Các đặc tính vật lý	Cu	Ni	Pt	W
Nhiệt độ nóng chảy (^{0}C)	1083	1453	1769	3380
Tỷ nhiệt ở 20^{0}C(J^{0}C^{-1}kg^{-1})	400	450	135	125
Độ dẫn nhiệt (W^{0}C^{-1} m^{-1})	400	90	73	120
Hệ số dãn nở tuyến tính (^{0}C^{-1})	$16,7.10^{-6}$	$1,28.10^{-6}$	$8,9.10^{-6}$	6.10^{-6}
Điện trở suất ở 20^{0}C (Ω.m)	$1,72.10^{-8}$	$10,6.10^{-6}$	$10,6.10^{-6}$	$5,52.10^{-6}$
Hệ số nhiệt điện trở suất ở 20^{0}C (^{0}C^{-1})	$3,9.10^{-3}$	$4,7.10^{-3}$	$3,9.10^{-3}$	$4,5.10^{-3}$

- Màng mỏng: dùng để đo nhiệt độ trên bề mặt vật rắn. Khi đo người ta dán màng mỏng lên bề mặt vật cần đo (mỏng cỡ μm)

b. Cảm biến nhiệt điện trở bán dẫn

Cảm biến nhiệt điện trở silic (bán dẫn)

Các vật liệu bán dẫn rất nhạy cảm với nhiệt độ. Do đó người ta dùng vật liệu bán dẫn để chế tạo cảm biến đo nhiệt độ.

Silic tinh khiết có hệ số nhiệt điện trở $\alpha < 0$, nhưng khi được tác động ở một dải nhiệt độ nào đó thì $\alpha > 0$

$\theta < 2000C$ thì $\alpha > 0$

$\theta > 2000C$ thì $\alpha < 0$

$$R_T = R_0\left[1 + A(T - T_0) + B(T - T_0)^2\right] \qquad (1\text{-}4)$$

Trong đó : R0, T0 là điện trở, nhiệt độ ở điểm chuẩn (00K)

A = 0,007874 (K-1)

B = 1,874.10-5 (K-2)

Đo nhiệt độ bằng cảm biến LM35

Các vật liệu bán dẫn rất nhạy cảm với nhiệt độ. Do đó người ta dùng vật liệu bán dẫn để chế tạo cảm biến đo nhiệt độ. Trong báo cáo này tác giả sử dụng cảm biến LM35 để thiết kế và mô phỏng.

Những thông số quan trọng của LM35 như sau:

- Điện áp hoạt động: 4~20VDC
- Công suất tiêu thụ: khoảng 60uA
- Khoảng đo: -55°C đến 150°C
- **Điện áp tuyến tính theo nhiệt độ: 10mV/°C**
- Sai số: 0.25°C
- Kiểu chân: TO92
- Kích thước: 4.3 × 4.3mm

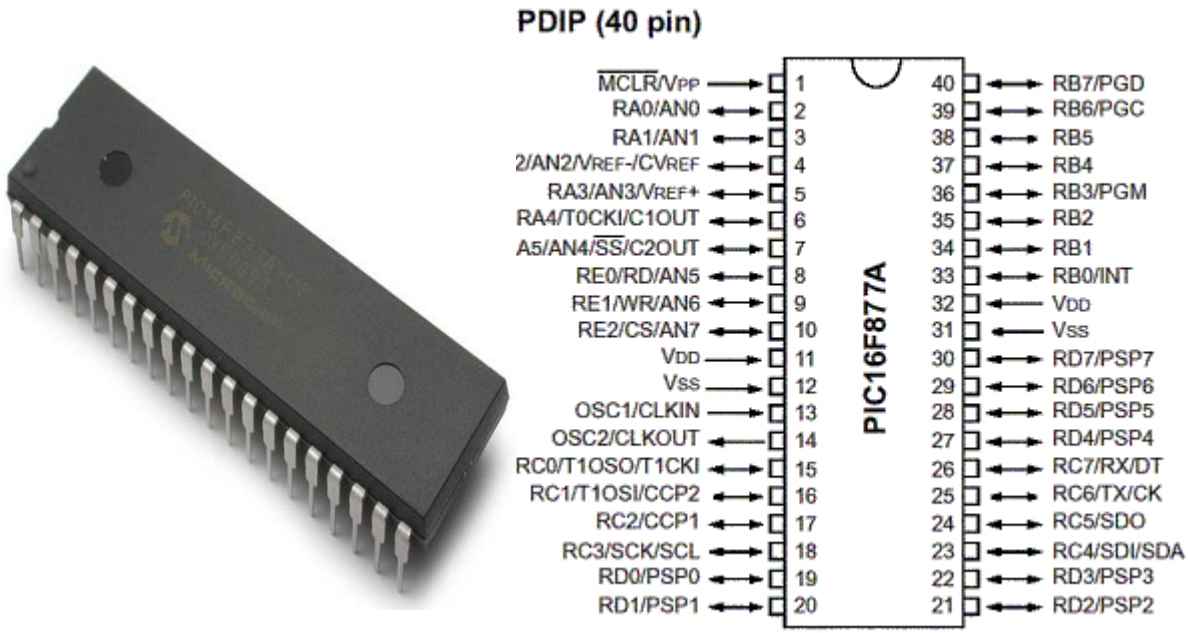
Tín hiệu ra của LM35 là điện áp thay đổi liên tục theo nhiệt độ. Vì vậy, khi thiết kế đo nhiệt độ cần sử dụng bộ biến đổi tương tự - số (ADC) của vi điều khiển để biến đổi tín hiệu liên tục nhận được thành tín hiệu số (Vi điều khiển chỉ xử lý với tín hiệu số). Do đó, trong thiết kế cần sử dụng các chân của cổng A hoặc cổng E làm lối vào của bộ biến đổi ADC.

Chương 2. Vi điều khiển PIC 16F877A

2.1 Giới thiệu Vi điều khiển PIC

a. Tổng quan

Khối xử lý tín hiệu làm nhiệm vụ nhận dòng dữ liệu gửi từ cảm biến nhiệt độ qua bộ biến đổi số tương tự và điều khiển hiển thị nhệt độ được lên màn hình LCD. Do đó cần lựa chọn loại Vi điều khiển có hỗ trợ biến đổi từu tín hiệu tương tự sang tín hiệu số. Có nhiều vi điều khiển có hỗ trợ giao tiếp này, tuy nhiên em chọn vi điều khiển PIC16F877A, vi điều khiển này có giá thành rẻ, sẵn có trên thị trường.

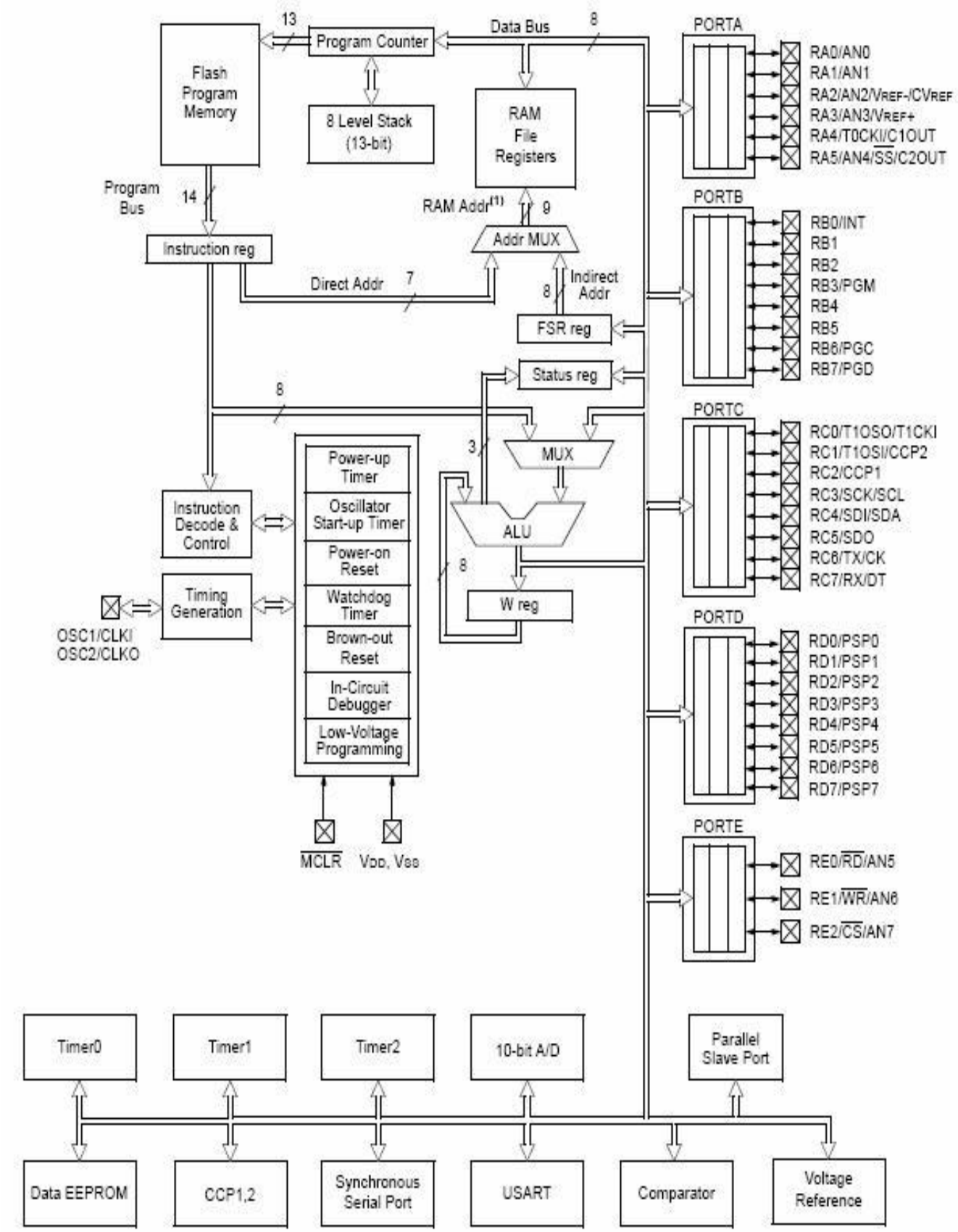


Hình 2.1. Sơ đồ chân PIC 16F877A.

Đây là vi điều khiển thuộc họ PIC16Fxxx với tập lệnh gồm 35 lệnh có độ dài 14 bít.Mỗi lệnh đều được thực thi trong một chu kì xung clock. Tốc độ hoạt động tối đa cho phép là 20 MHz với một chu kì lệnh là 200ns. Bộ nhớ chương trình 8Kx14 bit, bộ nhớ dữ liệu 368x8 byte RAM và bộ nhớ dữ liệu EEPROM với dung lượng 256x8 byte. Số PORT I/O là 5 với 33pin I/O.

- Các đặc tính ngoại vi bao gồm các khối chức năng sau:
+ Timer0: bộ đếm 8 bit với bộ chia tần số 8 bit.
+ Timer1: bộ đếm 16 bit với bộ chia tần số, có thể thực hiện chức năng đếm dựa vào xung clock ngoại vi ngay khi vi điều khiển hoạt động ở chế độ sleep.
+ Timer2: bộ đếm 8 bit với bộ chia tần số, bộ postcaler.
- Hai bộ Capture/so sánh/điều chế độ rông xung.
- Các chuẩn giao tiếp nối tiếp SSP (Synchronous Serial Port), SPI và I2C.
- Chuẩn giao tiếp nối tiếp USART với 9 bit địa chỉ.
- Cổng giao tiếp song song PSP (Parallel Slave Port) với các chân điều khiển RD, WR,CS ở bên ngoài.
- Các đặc tính Analog:
+ 8 kênh chuyển đổi ADC 10 bit.
+ Hai bộ so sánh.
- Bên cạnh đó là một vài đặc tính khác của vi điều khiển như:
+ Bộ nhớ flash với khả năng ghi xóa được 100.000 lần.
+ Bộ nhớ EEPROM với khả năng ghi xóa được 1.000.000 lần.
+ Dữ liệu bộ nhớ EEPROM có thể lưu trữ trên 40 năm.
+ Khả năng tự nạp chương trình với sự điều khiển của phần mềm.
+ Nạp được chương trình ngay trên mạch điện ICSP (In Circuit Serial Programming) thông qua 2 chân.
+ Watchdog Timer với bộ dao động trong.
+ Chức năng bảo mật mã chương trình.
+ Chế độ Sleep.
+ Có thể hoạt động với nhiều dạng Oscillator khác nhau.

b) Sơ đồ khối vi điều khiển PIC16F877A.

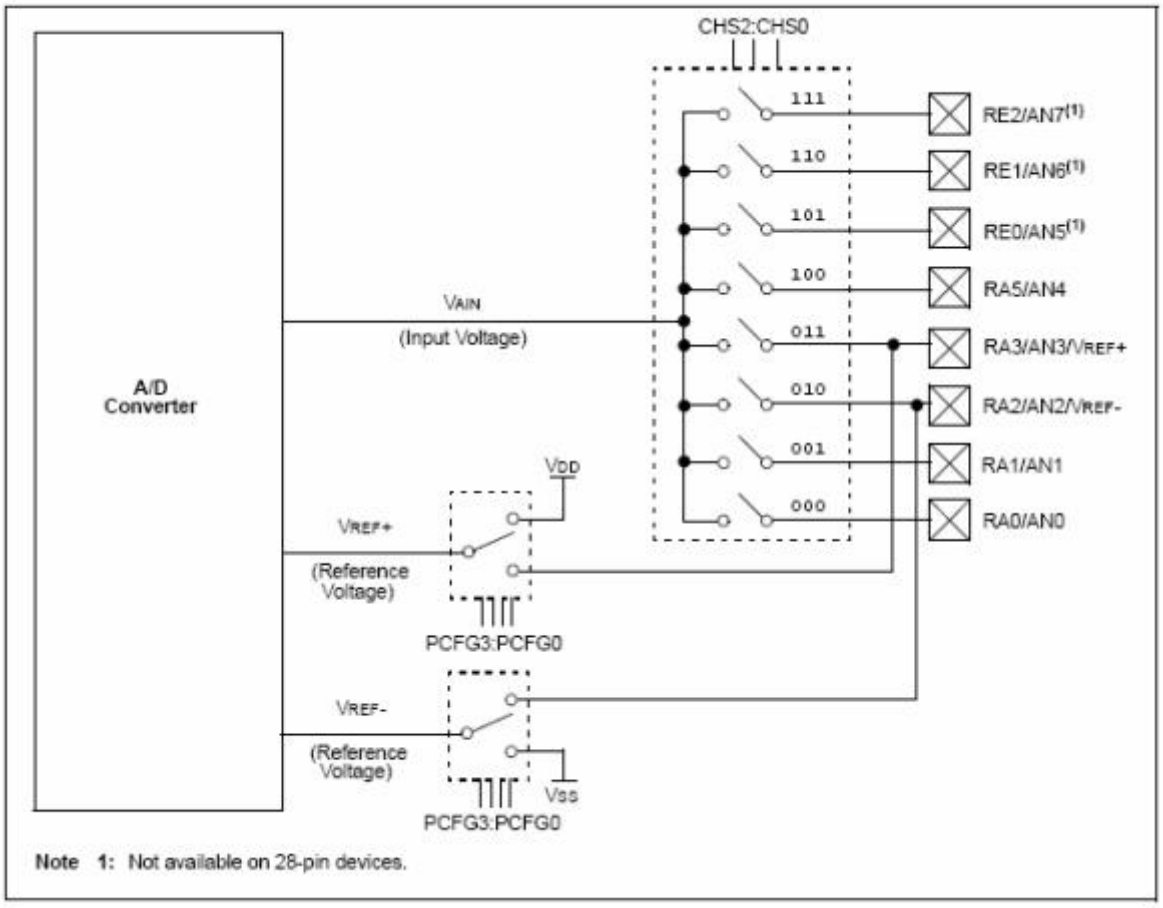


Hình 2.2. Sơ đồ khối PIC16F877A

2.2 Đặc điểm biến đổi ADC.

Vi điều khiển PIC 16F877A có bộ chuyển đổi tín hiệu tương tự sang tín hiệu số ADC 10 bit đa hợp 8 kênh và PIC 16F887 có 14 kênh. Mạch ADC dùng cho các ứng dụng giao tiếp với tín hiệu tương tự có thể nhận từ các cảm biến như cảm biến nhiệt độ LM35, cảm biến áp suất, cảm biến độ ẩm, cảm biến khoảng cách, …

Phần này sẽ khảo sát chi tiết khối ADC của PIC, các thanh ghi của khối ADC, trình tự thực hiện chuyển đổi, tập lệnh lập trình C cho ADC và ứng dụng ADC để đo nhiệt độ.



Hình 2.3. Sơ đồ khối ADC của PIC 16F877A.

Chức năng các thành phần:

AN0 đến AN7 (analog) là 4 ngõ vào của 8 kênh tương tự được đưa đến mạch đa hợp.

CHS<3:0> là các ngõ vào chọn kênh của bộ đa hợp tương tự.

Tín hiệu kênh tương tự đã chọn sẽ được đưa đến bộ chuyển đổi ADC.

Điện áp tham chiếu dương Vref+ có thể lập trình nối với nguồn cung cấp dương AVDD hoặc điện áp tham chiếu bên ngoài nối với ngõ vào Vref+ của chân AN3, bit lựa chọn là VCFG0.

Điện áp tham chiếu âm Vref- có thể lập trình nối với nguồn cung cấp AV SS hoặc điện áp tham chiếu bên ngoài nối với ngõ vào Vre- của chân AN2, bit lựa chọn là VCFG1. Hai ngõ vào Vref+ và Vref- có chức năng thiết lập độ phân giải cho ADC.

Bit ADON có chức năng cho phép ADC hoạt động hoặc tắt bộ A DC khi không hoạt động để giảm công suất tiêu tán, ADON bằng 1 thì cho phép, bằng 0 tắt.

Kết quả chuyển đổi là số nhị phân 10 bit sẽ lưu vào cặp thanh ghi 16 bit có tên là ADRESH và ADRESL, 10 bit kết quả lưu vào thanh ghi 16 bit nên có dạng lưu là canh lề trái và canh lề phải tùy thuộc vào bit lựa chọn có tên ADFM.

ADC có 8 kênh nhưng mỗi thời điểm chỉ chuyển đổi 1 kênh và chuyển đổi kênh nào thì phụ thuộc vào 4 bit chọn kênh CHS4:CHS0. Hai ngõ vào điện áp tham chiếu dương và âm có thể lập trình nối với nguồn VDD và VSS hoặc nhận điện áp tham chiếu từ bên ngoài qua 2 chân RA3 và RA2.

Khối ADC độc lập với CPU nên có thể hoạt động khi CPU đang ở chế độ ngủ do xung cung cấp cho ADC lấy từ dao động RC bên trong của khối ADC.

Khối ADC có 4 thanh ghi:

Thanh ghi lưu kết quả byte cao: ADRESH (A/D Result High Register)

Thanh ghi lưu kết quả byte thấp: ADRESL (A/D Result Low Register)

Thanh ghi điều khiển ADC thứ 0: ADCON0 (A/D Control Register 0)

Thanh ghi điều khiển ADC thứ 1: ADCON1 (A/D Control Register 1)

Để thực hiện chuyển đổi ADC thì phải thực hiện các bước sau:

Bước 1: Cấu hình cho port:Cấu hình cho các port ở chế độ ngõ vào tương tự.Không được định cấu hình cho các port ở chế độ xuất dữ liệu.

Bước 2: Cấu hình cho module ADC: Chọn xung clock cho chuyển đổi ADC.Định cấu hình cho điện áp chuẩn.Chọn kênh ngõ vào tương tự cần chuyển đổi.Chọn định dạng cho 2 thanh ghi lưu kết quả.Mở nguồn cho ADC.

Bước 3: Thiết lập cấu hình ngắt ADC nếu sử dụng: Xóa cờ báo ngắt ADIF của ADC.Cho bit ADIE bằng 1 để cho phép ADC ngắt.Cho bit PEIE bằng 1 để cho phép ngắt ngoại vi.Cho bit GIE bằng 1 để cho phép ngắt toàn cục.

Bước 4: Chờ hết thời gian ổn định theo yêu cầu.

Bước 5: Bắt đầu chuyển đổi bằng cách cho bit GO/DONE lên 1.

Bước 6: Kiểm tra chuyển đổi ADC kết thúc bằng cách: Kiểm tra liên tục bit GO/DONE nếu về 0 thì quá trình chuyển đổi kết thúc. Nếu dùng ngắt thì chờ ngắt ADC xảy ra.

Bước 7: Đọc cặp thanh ghi kết quả (ADRESH: ADRESL), xóa bit ADIF nếu dùng ngắt.

Bước 8: Thực hiện chuyển đổi kế tiếp.

2.3 Phần mềm thực hiện chương trình nạp cho PIC.

2.3.1 Chương trình dịch CCS.

Sự ra đời của một loại vi điều khiển đi kèm với việc phát triển phần mềm ứng dụng cho việc lập trình cho con vi điều khiển đó. Vi điều khiển chỉ hiểu và làm việc với hai con số 0 và 1. Ban đầu để việc lập trình cho VĐK là làm việc với dãy các con số 0 và 1. Sau này khi kiến trúc của Vi điều khiển ngày càng phức tạp, số lượng thanh ghi lệnh nhiều lên, việc lập trình với dãy các số 0 và 1 không còn phù hợp nữa, đòi hỏi ra đời một ngôn ngữ mới thay thế. Và ngôn ngữ lập trình Assembly. Ở đây ta không nói nhiều đến Assmebly. Sau này khi ngôn ngữ C ra đời, nhu cầu dùng ngôn ngữ C đề thay cho ASM trong việc mô tả các lệnh lập trình cho Vi điều khiển một cách ngắn gọn và dễ hiểu hơn đã dẫn

đến sự ra đời của nhiều chương trình soạn thảo và biên dịch C cho Vi điều khiển : Keil C, HT-PIC, MikroC, CCS…Em chọn CCS vì CCS là một công cụ lập trình C mạnh cho Vi điều khiển PIC. Những ưu và nhược điểm của CCS sẽ được đề cập đến trong các phần dưới đây.

h) Giới thiệu về CCS .

CCS là trình biên dịch lập trình ngôn ngữ C cho Vi điều khiển PIC của hãng Microchip. Chương trình là sự tích hợp của 3 trình biên dich riêng biết cho 3 dòng PIC khác nhau đó

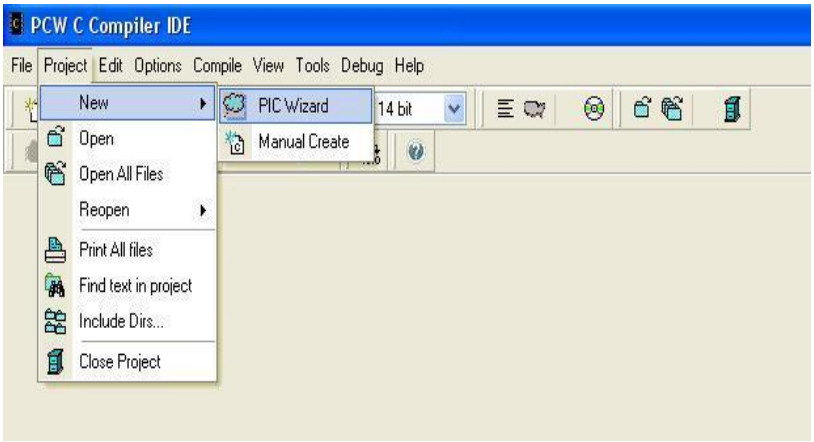
là:

- PCB cho dòng PIC 12-bit opcodes
- PCM cho dòng PIC 14-bit opcodes
- PCH cho dòng PIC 16 và 18-bit

Tất cả 3 trình biên dich này được tích hợp lại vào trong một chương trình bao gồm cả trình soạn thảo và biên dịch là CCS.

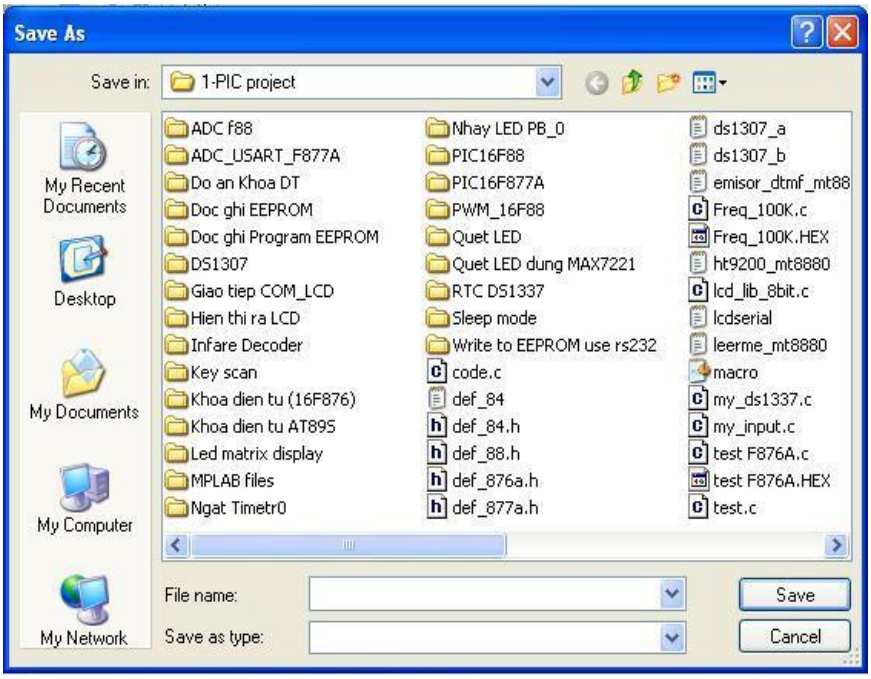
***i)* Tạo một PROJECT sử dụng PIC Wizard.**

Trước hết bạn khởi động chương trình làm việc PIC C Compiler. Từ giao diện chương trình bạn di chuột chọn ***Project-> New-> PIC Wizard*** nhấn nút trái chuột chọn.



Hình 2.4. Giao diện lưu trữ code

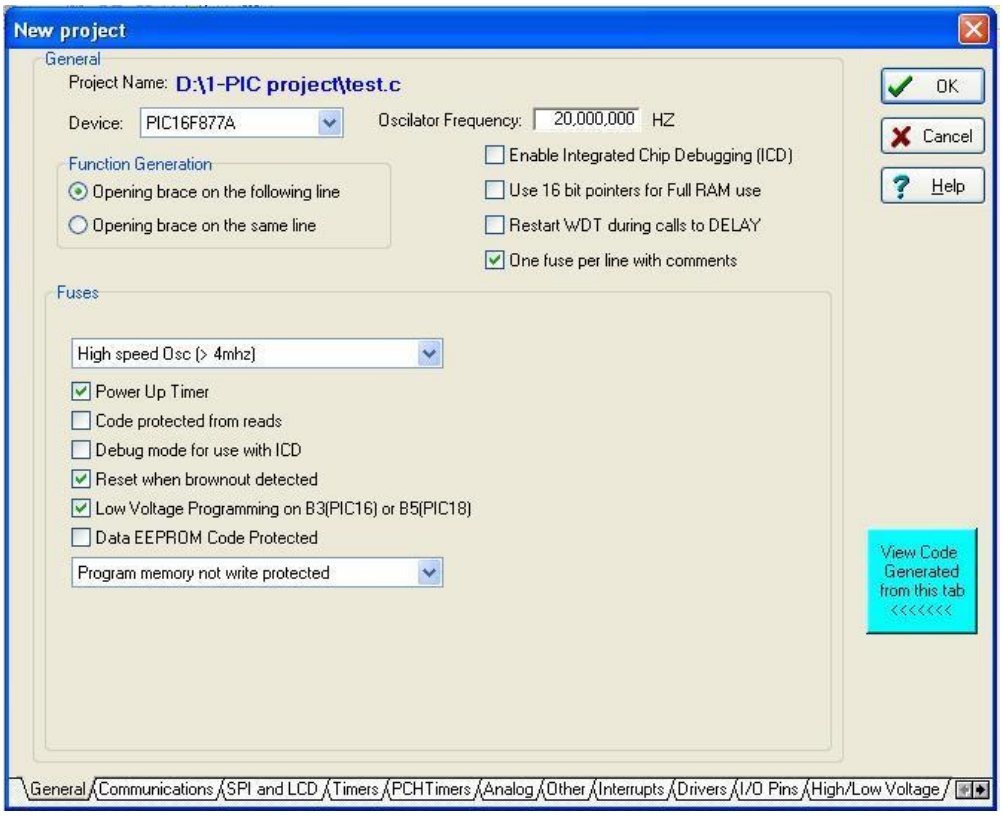
Sau khi nhấn chuột, một cửa sổ hiện ra yêu cầu ban nhập tên Files cần tạo. Bạn tạo một thư mục mới, vào thư mục đó và lưu tên files cần tạo tại đây.Như vậy là xong bước đầu tiên. Sau khi nhấn nút ***Save,*** một cửa sổ ***New Project*** hiện ra.Trong của sổ này bao gồm rất nhiều Tab, mỗi Tab mô tả về một vài tính năng của con PIC. Ta sẽ chọn tính năng sử dụng tại các Tab tương ứng. Dưới đây sẽ trình bày ý nghĩa từng mục chọn trong mỗi Tab. Các mục chọn này chính là đề cập đến các tính năng của một con PIC, tùy theo từng loại mà sẽ có các Tab tương ứng. Đối với từng dự án khác nhau, khi ta cần sử dụng tính năng nào của con PIC thì ta sẽ chọn mục đó. Tổng cộng có 13 Tab đẻ ta lưa chọn. Em giới thiệu những Tab chính thường hay được sử dụng.



Hình 2.5. Giao diện lưu chương trình.

j) Tab General

Tab General cho phép ta lựa chọn loại PIC mà ta sử dụng và một số lựa chọn khác như chọn tần số thạch anh dao động, thiết lập các bit CONFIG nhằm thiết lập chế độ hoạt động cho PIC.



Hình 2.6 Tab General

- **Device**: Liệt kê danh sách các loại PIC 12F, 16F, 18F… Ta sẽ chọn tên Vi điều khiển PIC mà ta sử dụng trong dự án. Lấy ví dụ chọn PIC16F877A
- **Oscilator Frequency:** Tần số thạch anh ta sử dụng, chọn 20 MHz (tùy từng loại)
- **Fuses:** Thiết lập các bit Config như: Chế độ dao động (HS, RC, Internal), chế độ bảo vệ Code, Brownout detected…
- Chọn kiểu con trỏ RAM là 16-bit hay 8-bit
- **Tab Communications**

Tab Communications liệt kê các giao tiếp nối tiếp mà một con PIC hỗ trợ, thường là RS232 và I2C, cùng với các lựa chọn để thiết lập chế độ hoạt động cho từng loại giao tiếp.

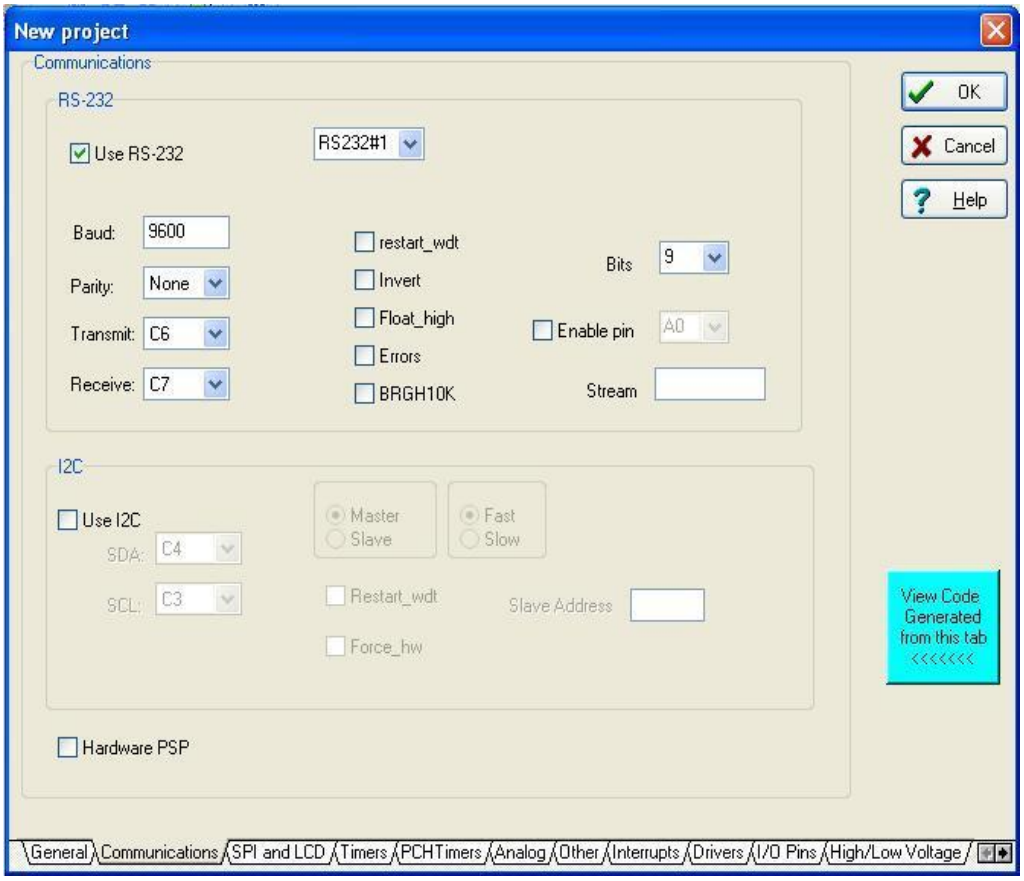
- **Giao tiếp RS232**

Mỗi một Vi điều khiển PIC hỗ trợ một cổng truyền thông RS232 chuẩn.

Tab này cho phép ta lựa chọn chân Rx, Tx, tốc độ Baud, Data bit, Bit Parity…

- **Giao tiếp I2C**

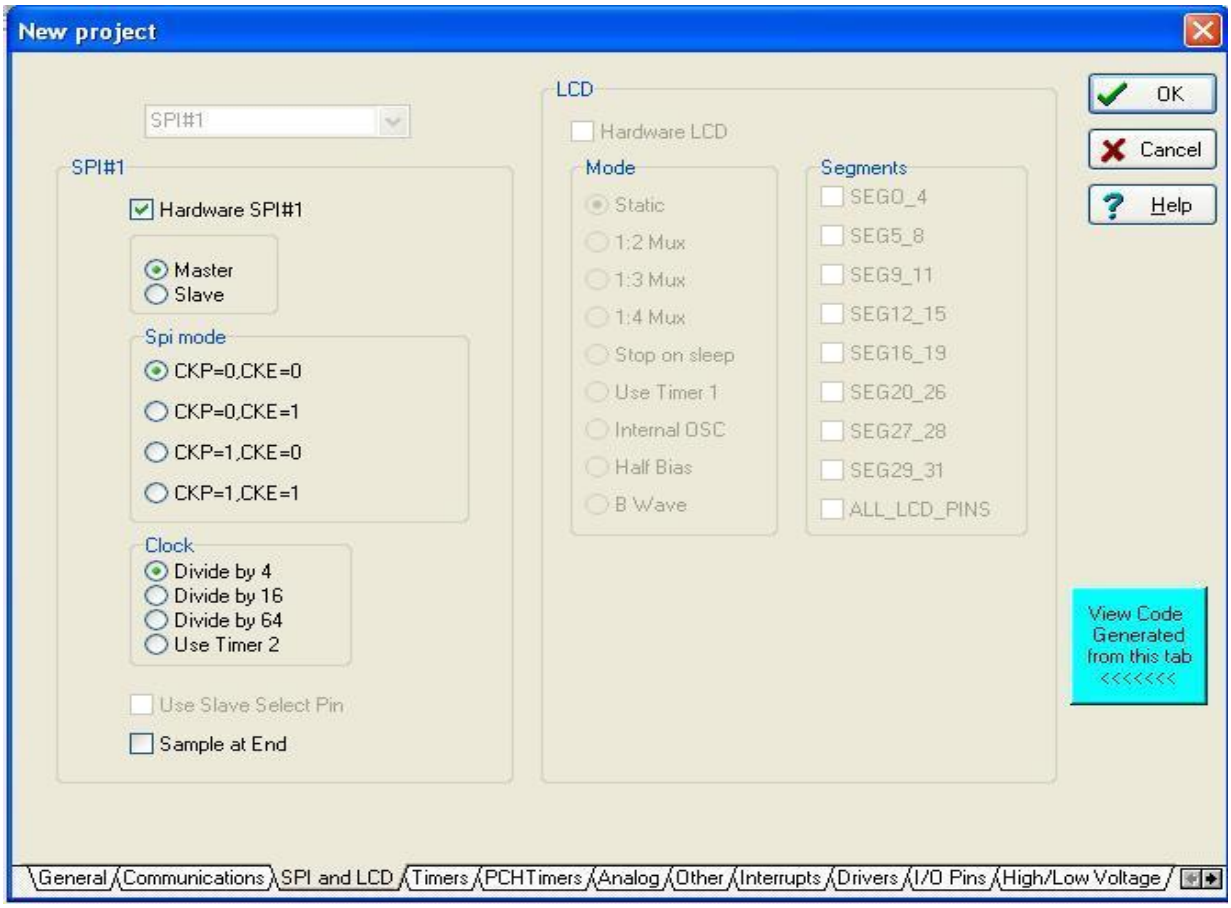
Để sử dụng I2C ta tích vào nút chọn Use I2C, khi đó ta có các lựa chọn: Chân SDA, SCL, Tốc độ truyền (Fast - Slow), chế độ Master hay Slave, địa chỉ cho Salve.



Hình 2.7 Tab Communications

***k)* Tab SPI and LCD**

Tab này liệt kê cho người dùng các lựa chọn đối với giao tiếp nối tiếp SPI, chuẩn giao tiếp tốc độ cao mà PIC hỗ trợ về phần cứng. Chú ý khi ta dùng I2C thì không thể dùng SPI và ngược lại. Để có thể sử dụng cả hai giao tiếp này cùng một lúc thì buộc một trong 2 giao tiếp phải lập trình bằng phần mềm (giống như khi dùng I2C cho các chip AT8051, không có hỗ trợ phần cứng SSP). Phần cấu hình cho LCD dành cho các chíp dòng 18F và 30F.

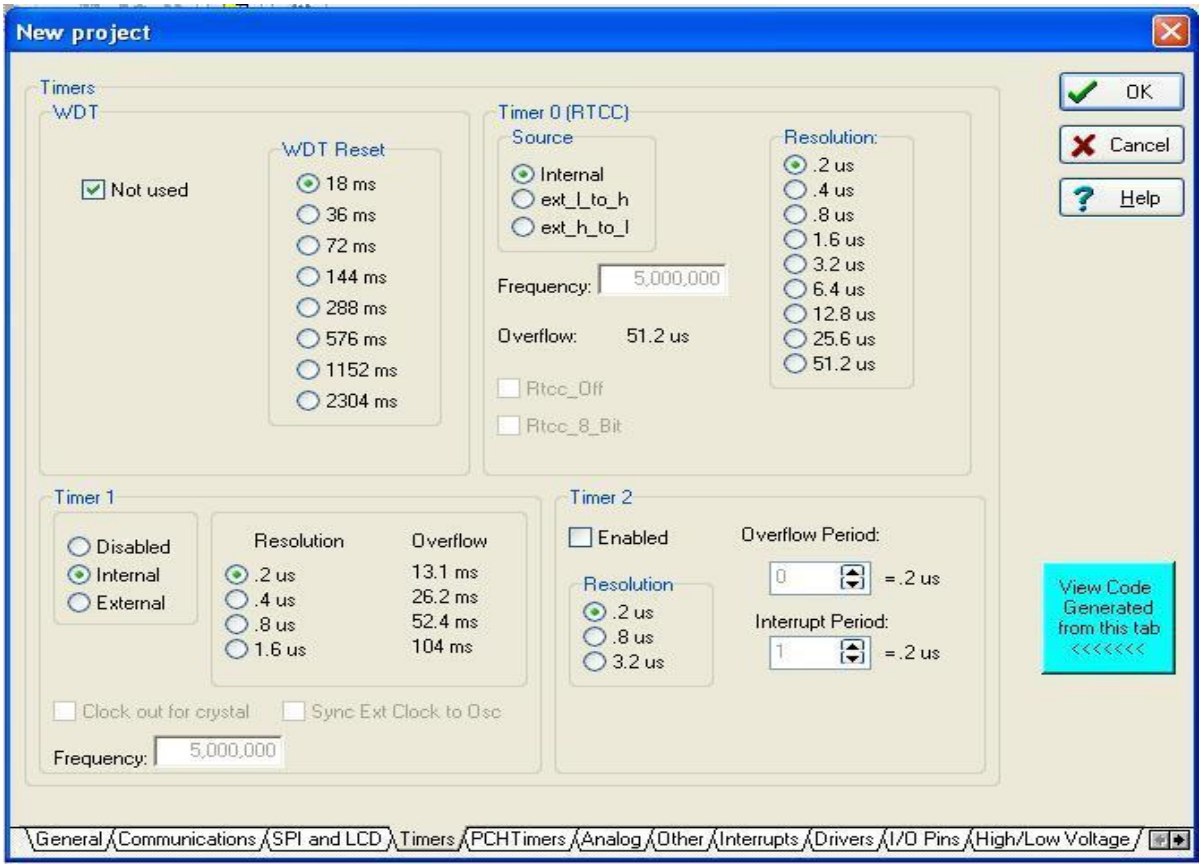


Hình 2.8 Tab SPI and LCD

l) Tab Timer

Liệt kê các bộ ***đếm/định thời*** mà các con PIC dòng Mid-range có: Timer0, timer1,timer2, WDT…

Trong các lựa chọn cấu hình cho các bộ đếm /định thời có: chọn nguồn xung đồng hồ (trong/ngoài), khoảng thời gian xảy ra tràn…

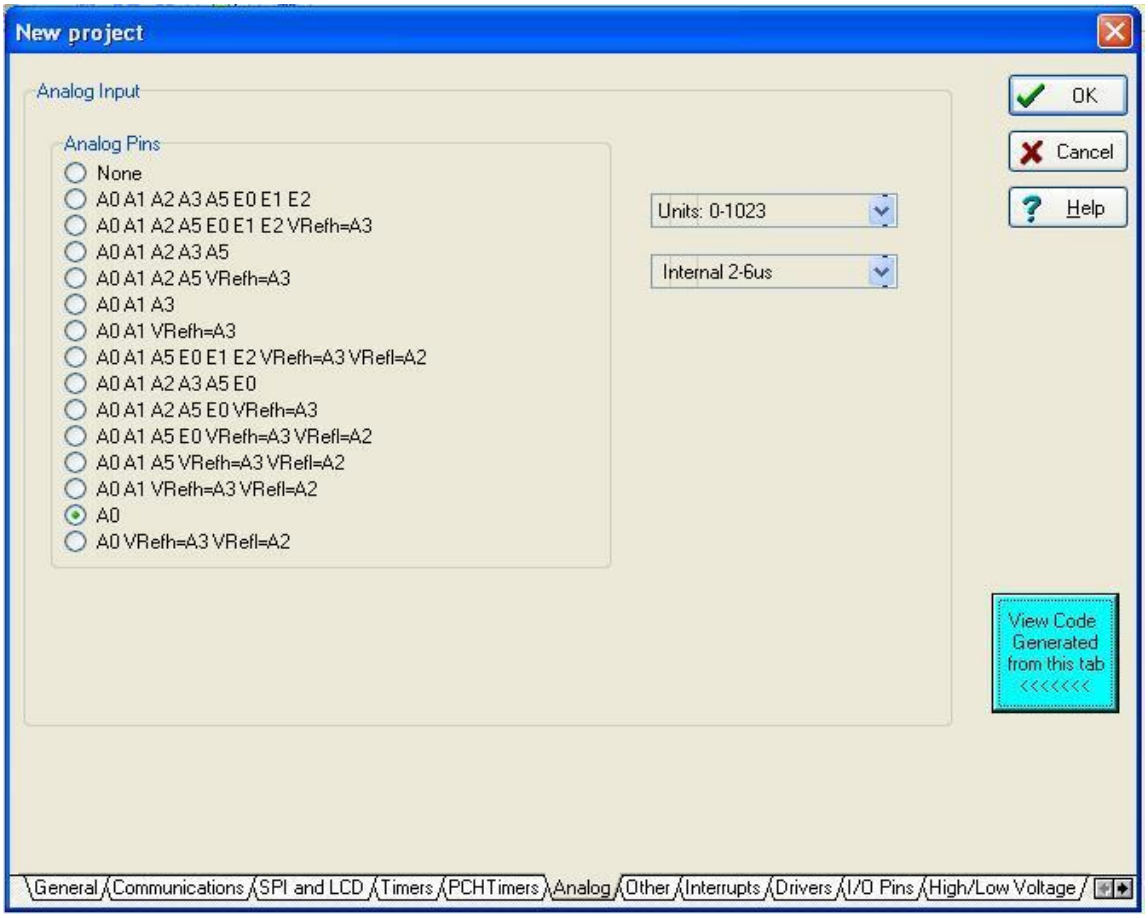


Hình 2.9 Tab Timer

m) Tab Analog

Liệt kê các lựa chọn cho bộ chuyển đổi tương tự/số (ADC) của PIC. Tùy vào từng IC cụ thể mà có các lựa chọn khác nhau, bao gồm:

- Lựa chọn cổng vào tương tự
- Chọn chân điện áp lấy mẫu (Vref)
- Chọn độ phân giải: 8-bit = 0 ~ 255 hay 10-bit = 0~1023
- Nguồn xung đồng hồ cho bộ ADC (trong hay ngoài), từ đó mà ta có được tốc độ lấy mẫu, thường ta chọn là ***internal 2-6 us.***
- Khi không sử dụng bộ ADC ta chọn **none**



Hình 2.10Tab Analog

Tab này cho phép ta thiết lập các thông số cho các bộ **Capture/Comparator/PWM.**

Capture - Bắt giữ

- Chọn bắt giữ xung theo sườn dương (**rising edge**) hay sườn âm (**falling edge**) của xung vào.
- Chọn bắt giữ sau 1, 4 hay 16 xung (copy giá trị của TimerX vào thanh ghi lưu trữ CCCPx sau 1, 4 hay 16 xung).

Compare - So sánh

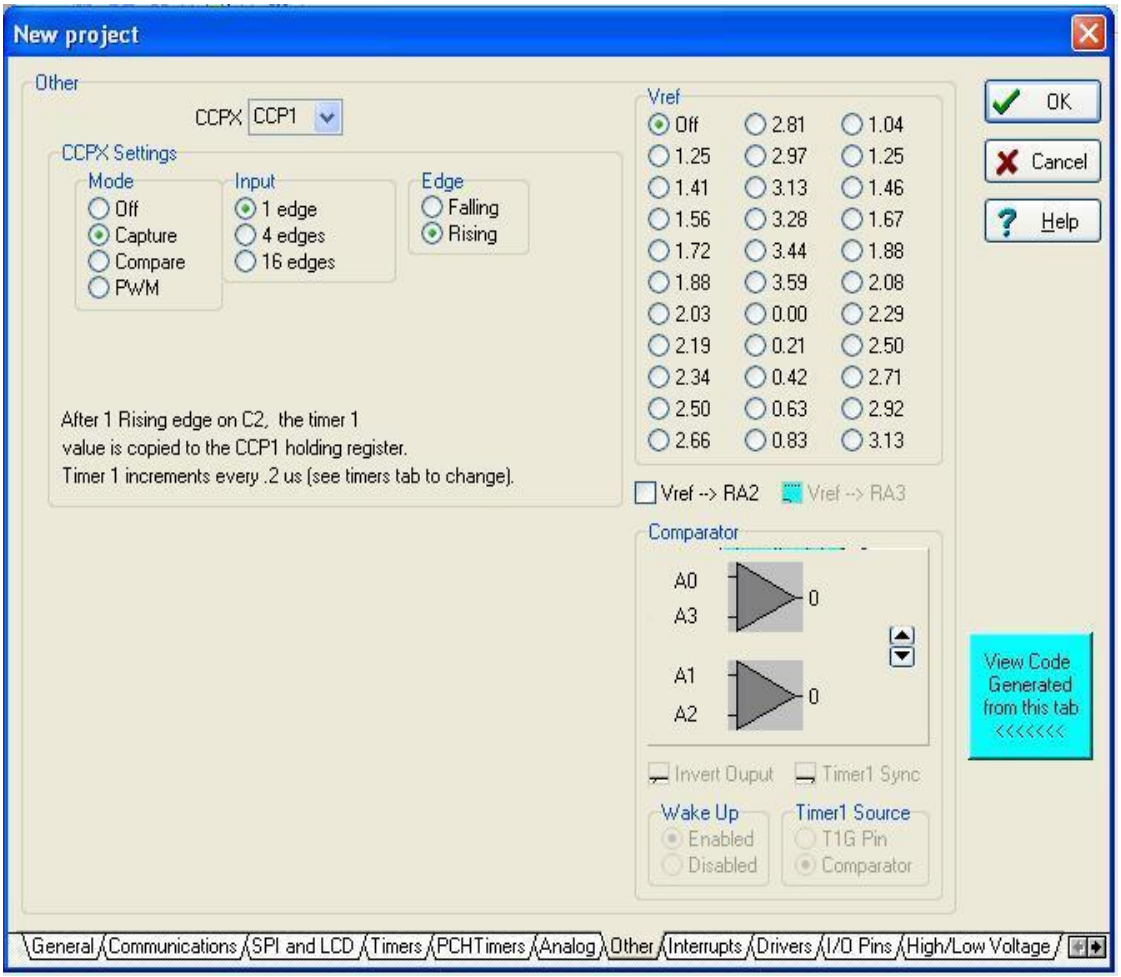
- Ta có các lựa chọn thực hiện lệnh khi xayư ra bằng nhau giữa 2 đối tượng so sánh là giá trị của Timer1 với giá trị lưu trong thanh ghi để so sánh. Bao gồm:
  - Thực hiện ngắt và thiết lập mức 0.
  - Thực hiện ngắt và thiết lập mức 1.
  - Thực hiện ngắt nhưng không thay đổi trạng thái của chân PIC.
  - Đưa Timer1 về 0 nhưng không thay đổi trạng thái chân.

PWM - Điều chế độ rộng xung

- Lựa chọn về tần số xung ra và duty cycle. Ta có thể lựa chọn sẵn hay tự chọn tần số, tất nhiên tần số ra phải nằm trong một khoảng nhất định.

Comparator - So sánh điện áp

- Lựa chọn mức điện áp so sánh Vref. Có rất nhiều mức điện áp để ta lựa chọn. Ngoài ra ta còn có thể lựa chọn cho đầu vào của các bộ so sánh.

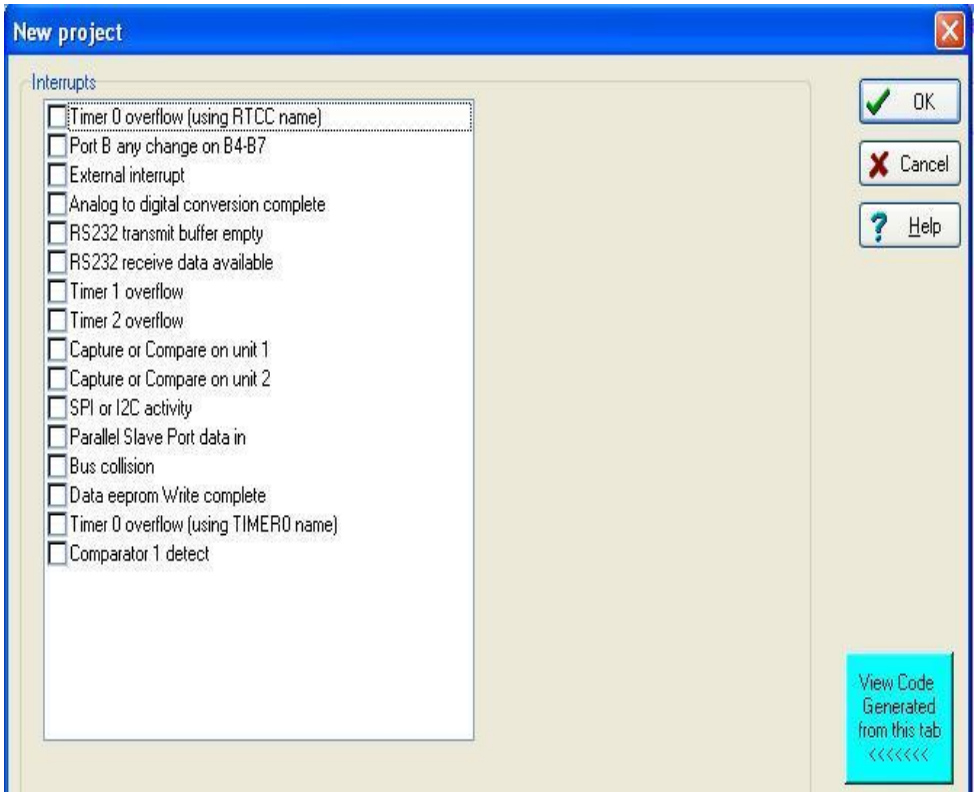


Hình 2.11 Tab Other

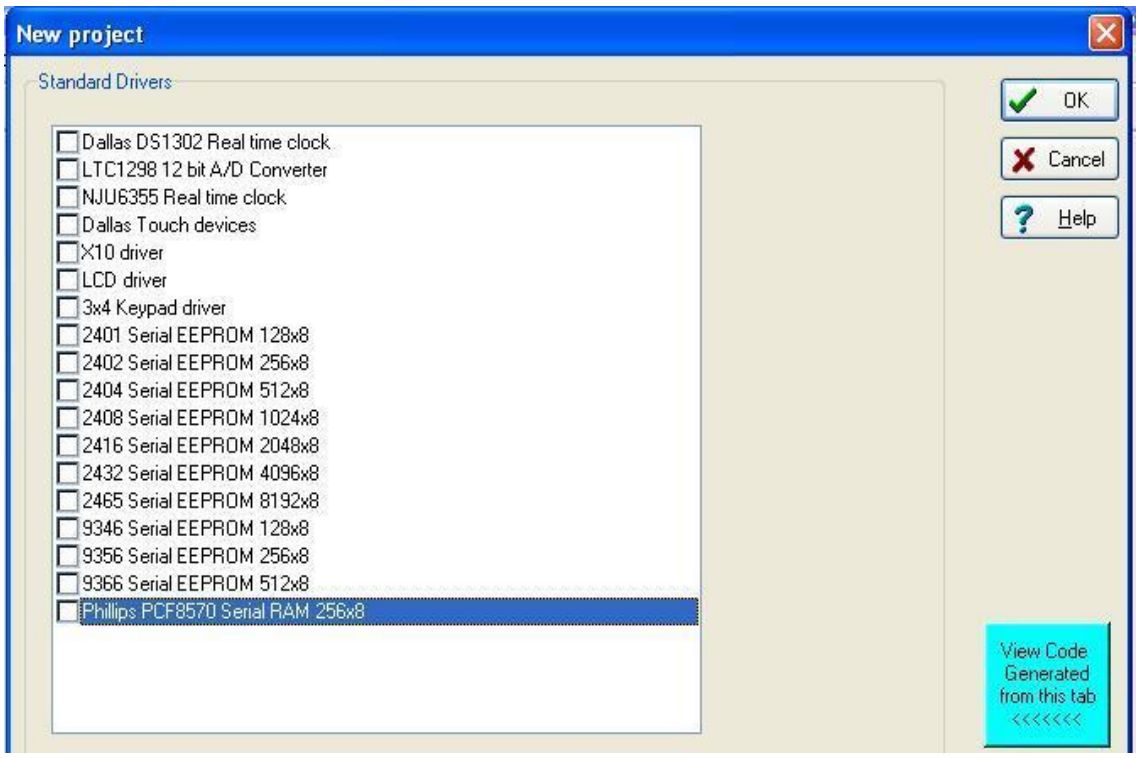
n) **Tab Interrupts và Tab Driver**

Tab Interrupts cho phép ta lựa chọn nguồn ngắt mà ta muốn sử dụng. Tùy vào từng loại PIC mà số lượng nguồn ngắt khác nhau, bao gồm: ngắt ngoài 0(INT0), ngắt RS232, ngắt Timer, ngắt I2C-SPI, ngắt onchange PORTB.v.v…

Tab Drivers được dùng để lựa chọn những ngoại vi mà trình dịch đã hỗ trợ các hàm giao tiếp. Đây là nhưng ngoại vi mà ta sẽ kết nối với PIC, trong các IC mà CCS hỗ trợ, đáng Chú ý là các loại EEPROM như 2404, 2416, 2432, 9346, 9356…Ngoài ra còn có IC RAM PCF8570, IC thời gian thực DS1302, Keypad 3x4, LCD, ADC… Chi tiết ta có thể xem trong thư mục Driver của chương trình**:** **\...\PICC\Drivers**



Hình 2.12 Tab Interrupts

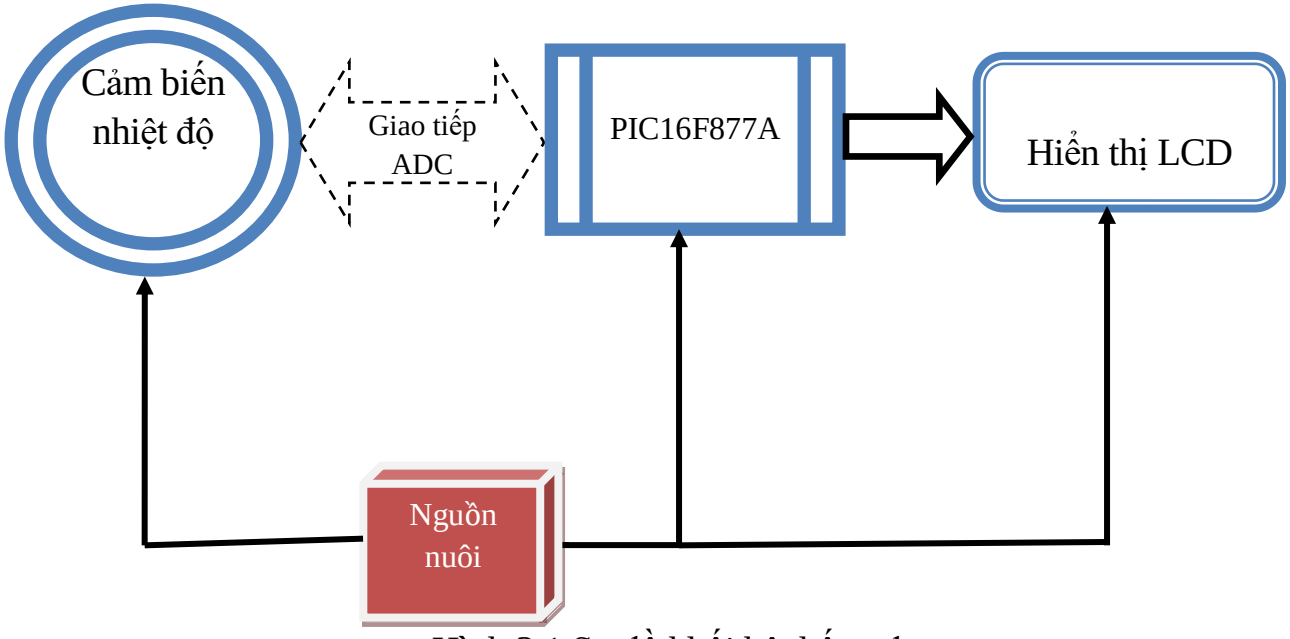


Hình 2.13 Tab Driver

Sau các bước chọn trên, ta nhấn OK để kết thúc quả trình tạo một Project trong CCS, một Files ***ten_project.c*** được tạo ra, chứa những khai báo cần thiết cho PIC trong một Files ***ten_project.h.***

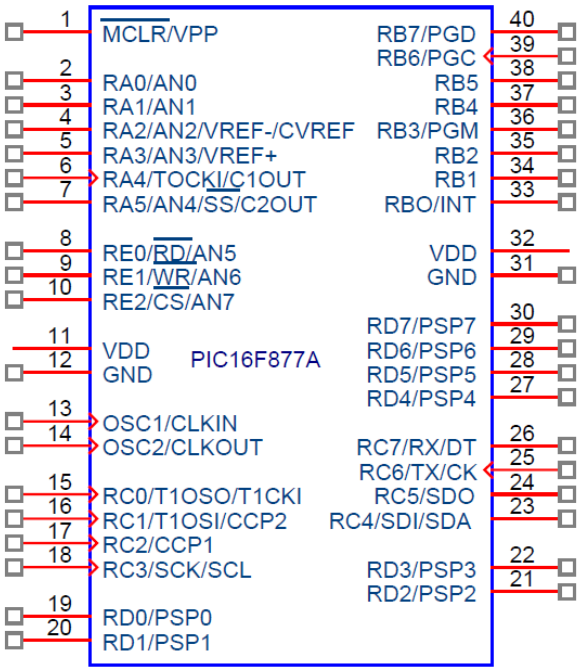
Chương 3. Thiết kế hệ thống đo nhiệt độ dùng cảm biến LM335

Phần này sẽ trình bày toàn bộ thiết kế hệ thống đo khoảng cách sử dụng cảm biến siêu âm SRF10.Sơ đồ khối hệ thống được trình bày dưới đây.



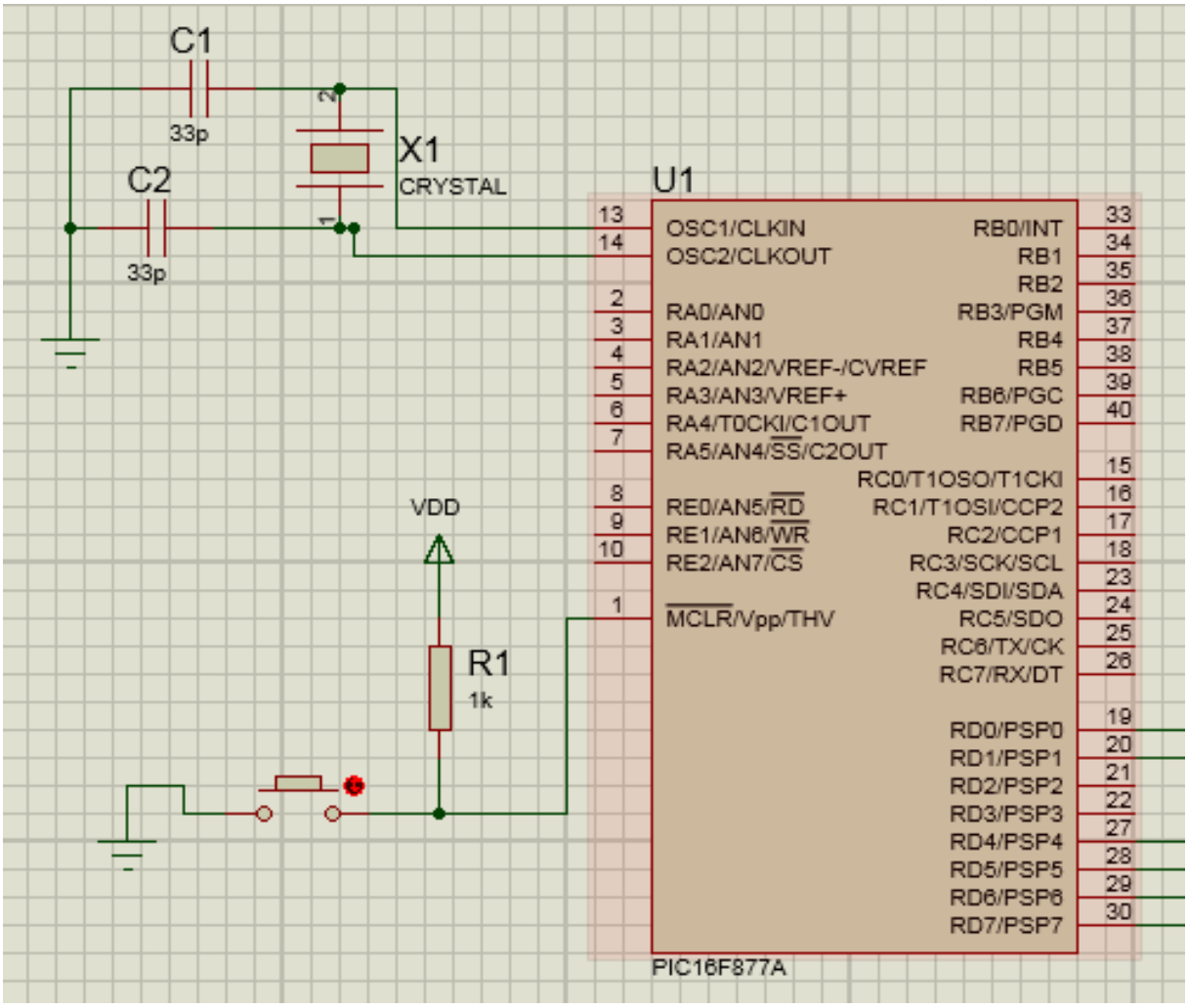
Hình 3.1 Sơ đồ khối hệ thống đo

3.1 Thiết kế phần cứng khối xử lý tín hiệu



Hình 21. Vi điều khiển PIC16F877A

Để vi điều khiển hoạt động ta cần cấp nguồn cho nó, PIC 16F877A có 4 chân cấp nguồn trong đó chân 11, 32 nối nguồn +5V, chân 12, 31 nối đất. Sau khi cấp nguồn ta cần cung cấp tiếp xung clock cho hoạt động của vi điều khiển. Ở đây ta sẽ dùng thạch anh làm nguồn xung để cấp cho PIC qua chân 13,14 của PIC. Tuy nhiên như ta đã biết, các xung dao động do thạch anh tạo ra cũng không thực sự ổn định một cách tuyệt đối, và cách khắc phục là gắn thêm các tụ lọc vào thạch anh. Thạch anh sử dụng ở đây là 12 MHz vì vậy giá trị tụ lọc tương ứng phù hợp với nó là 33pF. Vậy ta sẽ mắc được sơ đồ mạch nguyên lý của khối này như sau:



Hình 3.2 Sơ đồ nguyên lý của khối xử lý tín hiệu

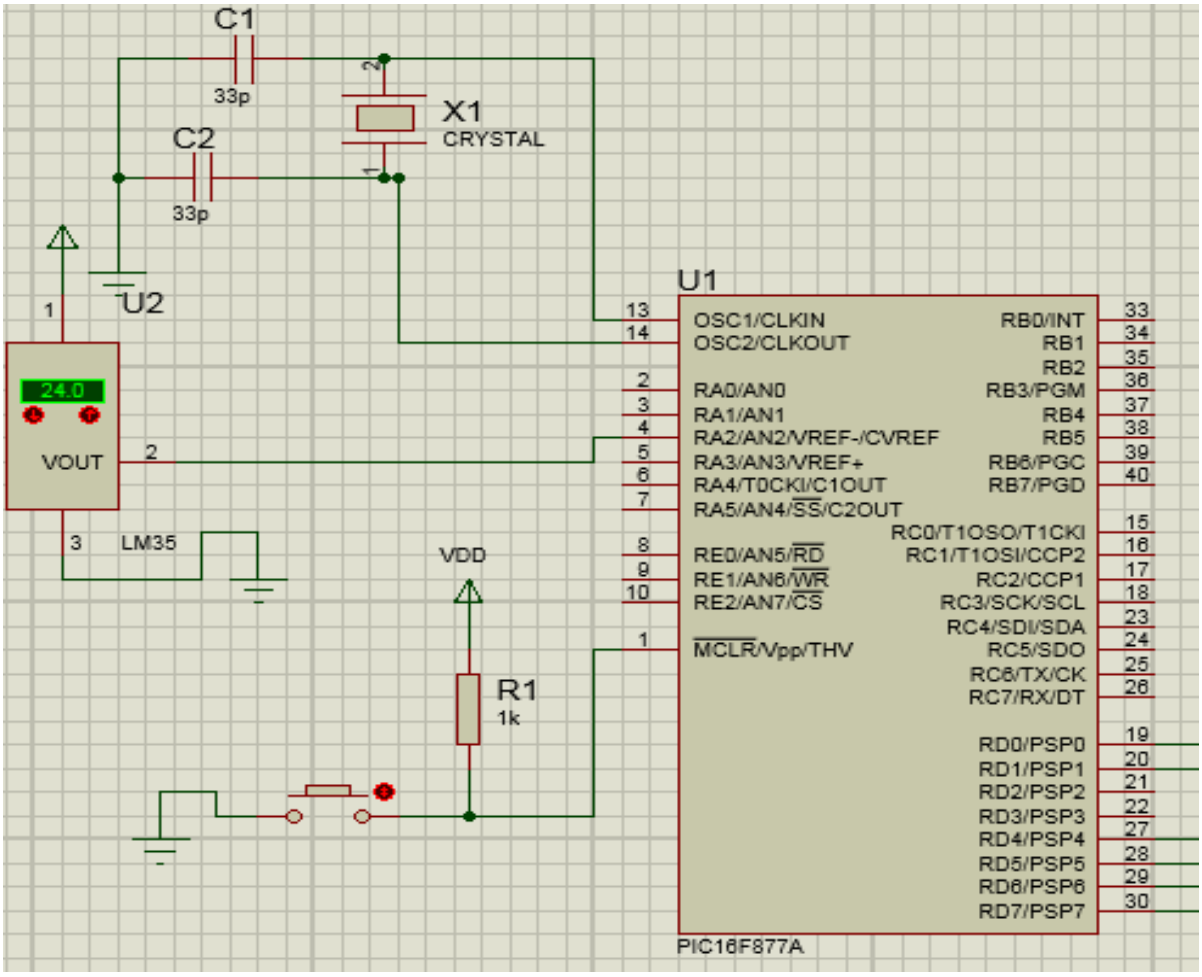
Chân số 1 MCLR được đấu nối thêm như trên đóng vai trò reset PIC, làm việc ở sườn xuống (mức 1 về 0). Khi SW1 mở điện áp vào chân số 1 là +5V (mức 1) PIC không được reset, khi SW1 đóng, mạch kín, chân số 1 nối đất, điện áp vào sẽ là 0V (mức 0) là mức kích hoạt, hoạt động của PIC được reset lại.

3.2. Thiết kế phần cứng giao tiếp LM335 và Vi điều khiển PIC16F877A

Cảm biến LM335 cho ta tín hiệu tương tự. Do đó,, để có thể két nối với Vi điều khiển chúng ta thiết kế giao tiếp cảm biến và PIC 16F877A qua một đường có

khả năng biến đổi ADC. Trong thiết kế này em chọn chân A0 để làm lối vào cho cảm biến đo nhiệt độ.

Sơ đồ chi tiết thiết kế phần cứng như trên hình 3.3.

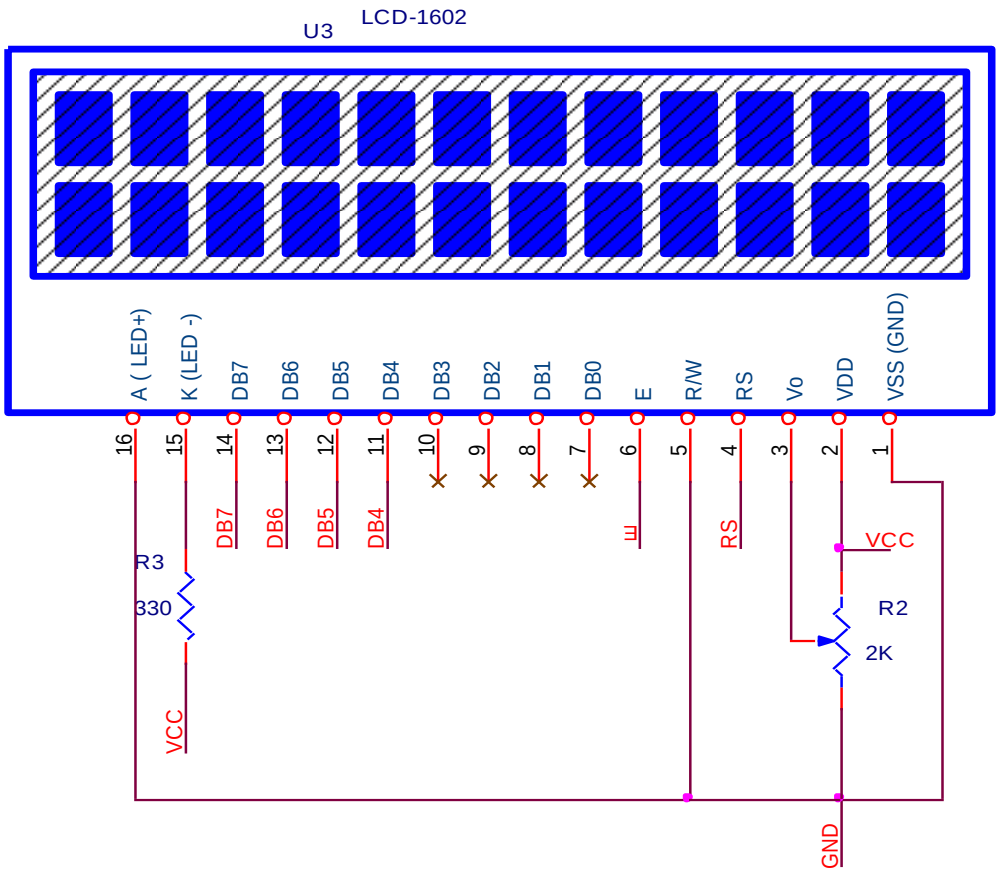


Hình 3.3 Sơ đồ nguyên lý của khối cảm biến và PIC 16F877A.

3.3. Khối hiển thị.

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

của Hitachi, một loại thiết bị hiển thị LCD rất thông dụng ở nước ta, cụ thể là sử dụng LCD_DM 1602A (1 dòng của HD44780).



Hình 3.4 Sơ đồ nguyên lý của khối hiển thị LCD

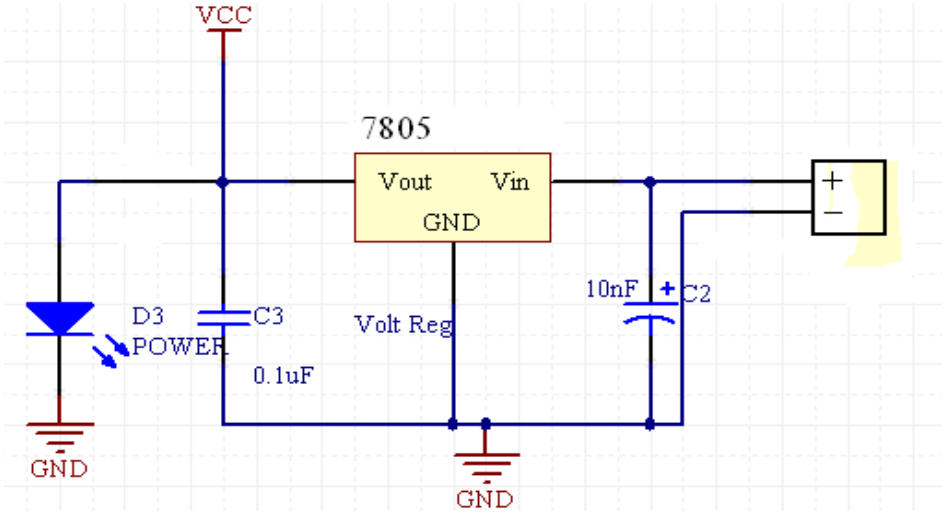
LCD1602 là loại 2 dòng, 16 kí tự, sử dụng nguồn nuôi thấp (từ 2,5 đến 5V). Có thể hoạt động ở hai chế độ 4 bit hoặc 8 bit. Với đầu vào 4 bit được lấy từ 4 chân D4→D7 của LCD nối từ RD4→RD7 của vi điều khiển PIC. Như vậy ở đây chúng ta chỉ sử dụng nửa byte cao của LCD. Chân RW đóng vai trò chọn chế độ đọc ghi cho LCD, mức logic "0" LCD hoạt động ở chế độ ghi, ngược lại ở chế độ đọc. Chân RS của LCD được nối với chân RD0 của Vi điều khiển. Chân E của LCD được nối với chân RD1 của Vi điều khiển. Các tín hiệu điều khiển cho phép hiển thị trên LCD được thực hiện thông qua lập trình.

3.4 Khối nguồn.

Làm nhiệm vụ cung cấp nguồn nuôi cho toàn bộ hệ thống. Sơ đồ khối bộ nguồn như trong hình 3.5.

Hình 3.5 Sơ đồ khối bộ nguồn

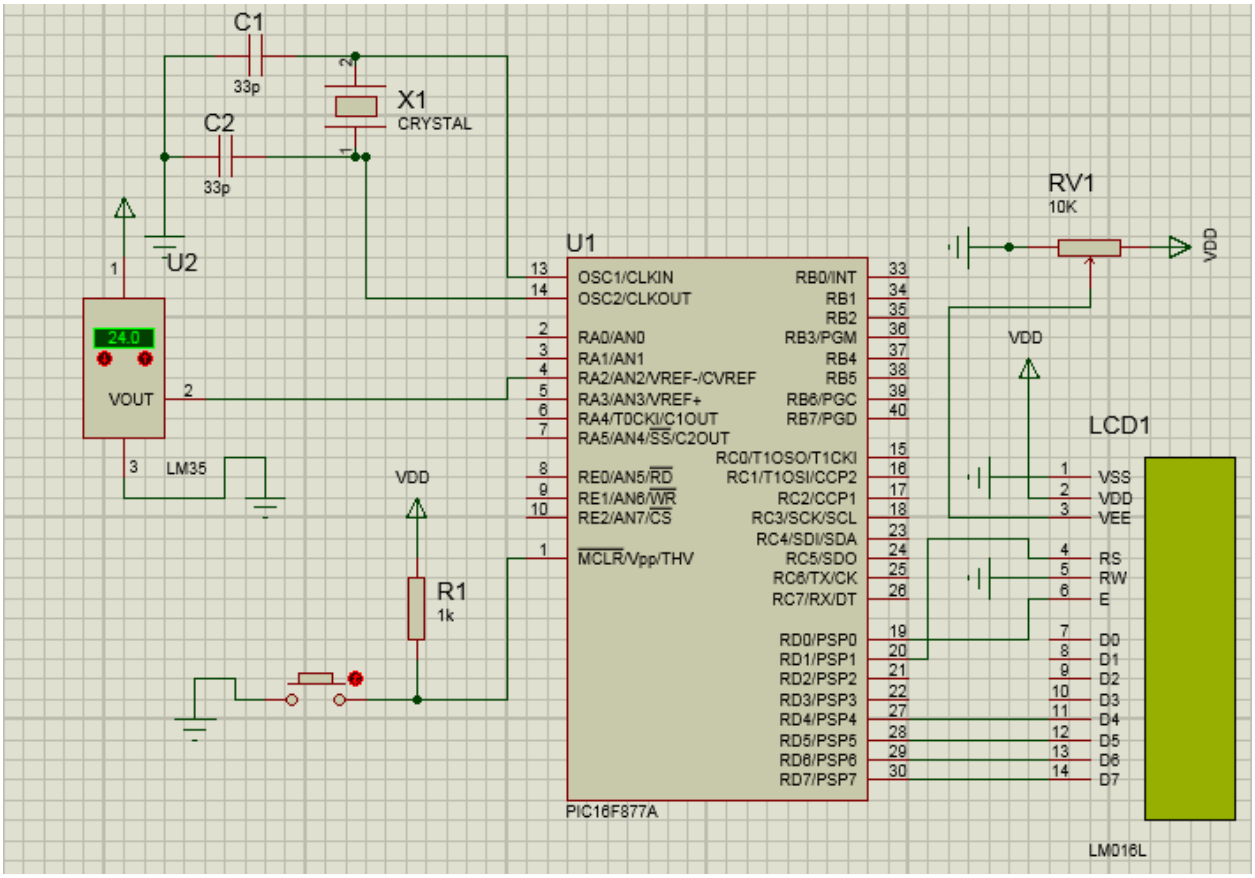
Ở đây bộ ổn áp dùng các IC 7805 để tạo các nguồn +5V ổn định cấp cho toàn mạch. Hình 3.6 là sơ đồ bộ nguồn +5V, tụ C2 và C3 để lọc nhiễu, diode D3 có nhiệm vụ báo nguồn.



Hình 3.6 Sơ đồ nguyên lý của bộ ổn áp

3.5 Sơ đồ nguyên lý và mạch in hệ thống đo

Sơ đồ nguyên lý của hệ thống đo nhiệt độ dùng cảm biến LM35 (LM335) được cho ở hình 3.7 dưới đây



Hình 3.7. Sơ đồ nguyên lý mạch đo nhiệt độ dùng LM335.

Cảm biến M335 thực hiện đo nhiệt độ, tín hiệu điện áp thay đổi sẽ gửi dòng dữ liệu kết quả đo về cho vi điều khiển. Vi điều khiển đọc giá trị đó, xuất giá trị này ra cổng PORTB dưới dạng mã nhị phân 8 bit và hiển thị giá trị nhiệt độ đo được lên màn hình LCD.

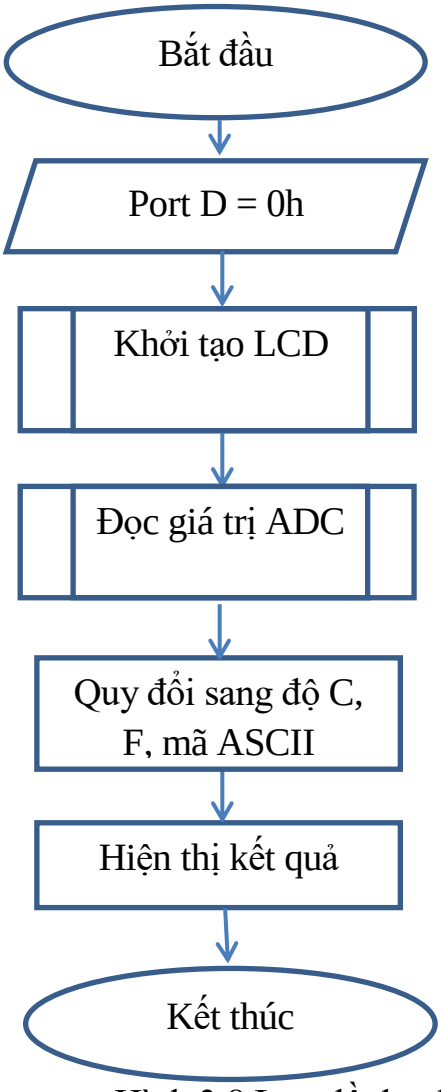
Sau thiết kế phần cứng để có thể điều khiển và đọc kết quả đo LM335 hoạt động tốt thì cần phải viết chương trình thỏa mãn những yêu cầu như:

- Thiết lập được giao tiếp LM335 với Vi điều khiển qua bộ biến đổi ADC.

- Nhận kết quả từ cảm biến và lọc nhiễu.
- Hiển thị kết quả nên LCD.

Từ đó ta có thể xây dựng lưu đồ thuật toán như sau:

2.6 Lưu đồ thuật toán.



Hình 3.8 Lưu đồ đo nhiệt độ.

Kết luận

Sau thời gian đồ án, kết quả em đã thực hiện được những nội dung chính như sau:

- Nghiên cứu tổng quan về Vi điều khiển và cảm biến nhiệt độ bán dẫn;
- Tìm hiểu phần mềm CCS;
- Tìm hiểu màn hình LCD 1602
- Thiết kế các thành phần của hệ thống đo nhiệt độ dùng cảm biến LM335;
- Xây dựng lưu đồ thuật toán và thực hiện viết chương trình cho hệ thống Vi điều khiển.

Lời cảm ơn

Sau thời gian nghiên cứu và được sự hướng dẫn tận tình của thầy Đỗ Anh Dũng em đã hoàn thành đồ án tốt nghiệp với đề tài : "**Thiết kế hệ thống Vi điều khiển PIC 16F877A đo nhiệt độ dùng cảm biến bán dẫn**" đúng thời gian quy định. Do thời gian thực hiện có hạn nên đồ án không tránh khỏi những thiếu sót. Em rất mong được sự góp ý của các thầy và bạn bè để đồ án được hoàn thiện hơn

Em xin chân thành cảm ơn thầy giáo Đoàn Hữu Chức đã tận tình hướng dẫn giúp đỡ em để em hoàn thành đồ án này. Trong thời gian học tập tại trường em xin chân thành cảm ơn tất cả các thầy trong Khoa Điện – Điện tử đã luôn tạo điều kiện giúp đỡ và truyền đạt nhiều kiến thức quý báu. Em xin cảm ơn về những sự giúp đỡ đó.

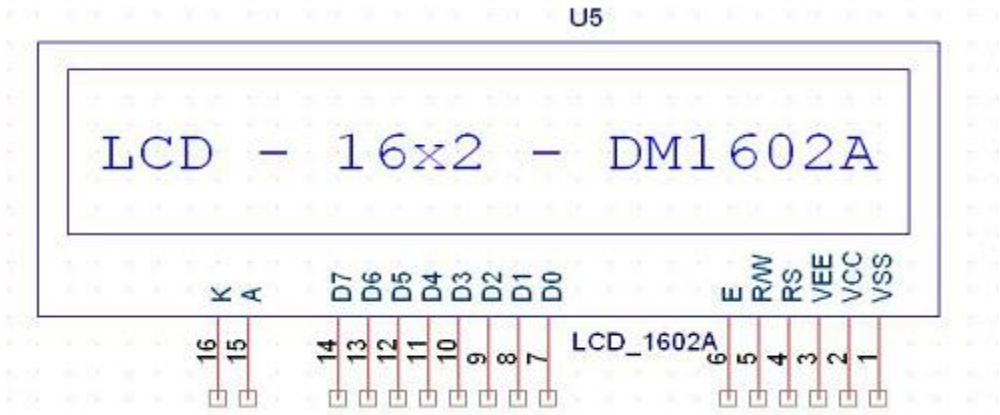
Em xin chân thành cảm ơn các thầy !

Hải phòng , ngày tháng năm 2022

Sinh viên thực hiện

Phụ lục 1 Cấu tạo LCD 1602A

Hình dáng của LCD 1602



Sơ đồ chân của LCD 1602

Các thanh ghi

Chân	Kí hiệu	Mức Logic	I/O	Chức năng
1	Vss	-	-	Nguồn (GND)
2	Vcc	-	-	Nguồn (+5V)
3	Vee	-	-	Chỉnh độ tương phản
4	RS	0/1	I	0 = Nhập lệnh 1 = Nhập dữ liệu
5	R/W	0/1	I	0 = Ghi dữ liệu 1 = Đọc dữ liệu
6	E	1, 1->0	I	Tín hiệu cho phép
7	DB0	0/1	I/O	Bus dữ liệu 0
8	DB1	0/1	I/O	Bus dữ liệu 1
9	DB2	0/1	I/O	Bus dữ liệu 2
10	DB3	0/1	I/O	Bus dữ liệu 3
11	DB4	0/1	I/O	Bus dữ liệu 4
12	DB5	0/1	I/O	Bus dữ liệu 5
13	DB6	0/1	I/O	Bus dữ liệu 6
14	DB7	0/1	I/O	Bus dữ liệu 7
15	Lamp-	-	-	Đèn LCD
16	Lamp+	-	-	Đèn LCD

- Thanh ghi IR: Mỗi lệnh được nhà sản xuất LCD đánh địa chỉ rõ ràng. Người dùng chỉ việc cung cấp địa chỉ lệnh bằng cách nạp vào thanh ghi IR.

Ví dụ:

Lệnh "hiển thị màn hình và con trỏ" có mã lệnh là 00001110

- Thanh ghi DR : Thanh ghi DR dùng để chứa dữ liệu 8 bit để ghi vào vùng RAM DDRAM hoặc CGRAM (ở chế độ ghi) hoặc dùng để chứa dữ liệu từ 2 vùng RAM này gửi ra cho MPU (ở chế độ đọc).

- **Cờ báo bận BF: (Busy Flag)**

Khi đang thực thi các hoạt động bên trong, LCD bỏ qua mọi giao tiếp với bên ngoài và bật cờ BF(thông qua chân DB7 khi có thiết lập RS=0, R/W=1) lên để cho biết nó đang "bận".

- **Bộ đếm địa chỉ AC : (Address Counter)**

Khi một địa chỉ lệnh được nạp vào thanh ghi IR, thông tin được nối trực tiếp cho 2

vùng RAM (việc chọn lựa vùng RAM tương tác đã được bao hàm trong mã lệnh). Sau khi ghi vào (đọc từ) RAM, bộ đếm AC tự động tăng lên (giảm đi) 1 đơn vị.

Bộ nhớ LCD

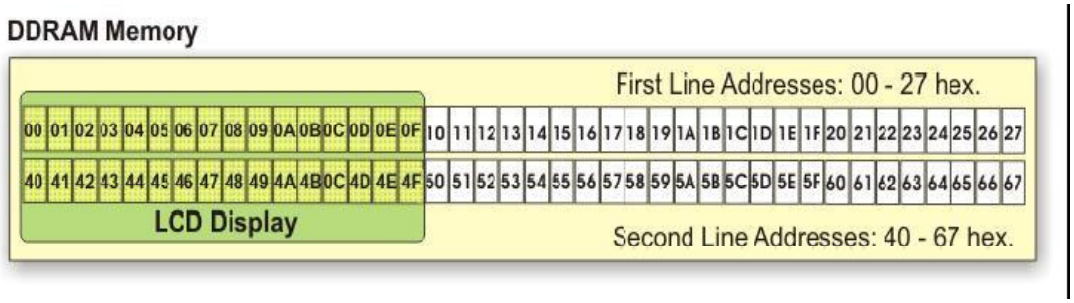
Vùng RAM hiển thị DDRAM : (Display Data RAM)

Vùng CGROM: Character Generator ROM

Vùng CGRAM : (Character Generator RAM)

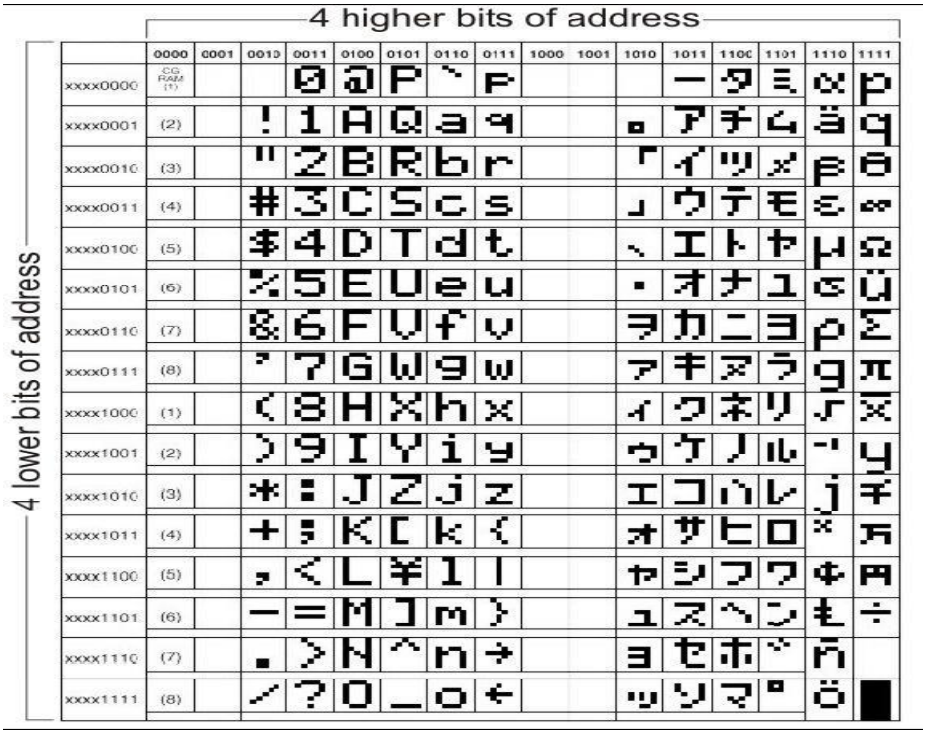
DDRAM

Đây là vùng RAM dùng để hiển thị, nghĩa là ứng với một địa chỉ của RAM là một ô kí tự trên màn hình.



CGROM

Chứa các mẫu kí tự loại 5x7 hoặc 5x10 điểm ảnh/kí tự, và định địa chỉ bằng 8 bit



A11 A10 A9 A8 A7 A6 A5 A4	A3 A2 A1 A0	O4 O3 O2 O1 LSB O0
	0 0 0 0	1 0 0 0 0
	0 0 0 1	1 0 0 0 0
	0 0 1 0	1 0 1 1 0
	0 0 1 1	1 1 0 0 1
	0 1 0 0	1 0 0 0 1
	0 1 0 1	1 0 0 0 1
	0 1 1 0	1 1 1 1 0
0 1 1 0 0 0 1 0	0 1 1 1	0 0 0 0 0

Mẫu kí tự đồ họa riêng.

Table 5 **Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data)**

For 5 × 8 dot character patterns

Character Codes (DDRAM data)	CGRAM Address	Character Patterns (CGRAM data)
7 6 5 4 3 2 1 0 (High … Low)	5 4 3 2 1 0 (High … Low)	7 6 5 4 3 2 1 0 (High … Low)
0 0 0 0 * 0 0 0	0 0 0	* * * 1 1 1 1 0 — Character pattern (1)
		* * * 0 0 0 0 0 — Cursor position
0 0 0 0 * 0 0 1	0 0 1	* * * 1 0 0 0 1 — Character pattern (2)
		* * * 0 0 0 0 0 — Cursor position
0 0 0 0 * 1 1 1	1 1 1	* * *

Notes:
1. Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).
2. CGRAM address bits 0 to 2 designate the character pattern line position. The 8th line is the cursor position and its display is formed by a logical OR with the cursor.
   Maintain the 8th line data, corresponding to the cursor display position, at 0 as the cursor display.
   If the 8th line data is 1, 1 bits will light up the 8th line regardless of the cursor presence.
3. Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).
4. As shown Table 5, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.
5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.

* Indicates no effect.

Tập lệnh:

Tập lệnh	Mã nhị phân										Mô tả	Thời gian thực thi
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Xoá hiển thị	0	0	0	0	0	0	0	0	0	1	Xoá hiện thị và đưa con trỏ về vị trí ban đầu (địa chỉ 0).	1.64mS
Con trỏ ban đầu	0	0	0	0	0	0	0	0	1	*	Trả con trỏ về vị trí ban đầu (địa chỉ 0). Ngoài ra đưa hiển thị đã bị dịch chuyển về vị trí ban đầu. Nội dung bộ nhớ hiển thị dữ liệu(DDRAM) không thay đổi.	1.64mS
Thiết lập chế độ	0	0	0	0	0	0	0	1	I/D	S	Thiết lập hướng di chuyển của con trỏ tăng/giảm(I/D=0:giảm,ID=1:tăng), chỉ rõ dịch chuyển hiển thị (S=0:không dịch chuyển hiển thị,S=0 dịch chuyển hiển thị). Hoạt động này được thực hiện trong suốt quá trình đọc/ghi dữ liệu	40uS
Điều khiển hiển thị	0	0	0	0	0	0	1	D	C	B	Bật/tắt hiển thị (D=0:tắt,D=1,bật) nhưng dữ liệu vẫn lưu trong DDRAM, bật/tắt con trỏ(C=0:tắt,C=1:bật) và bật tắt con trỏ nhấp nháy tại vị trí của kí tự (B=0:tắt,B=1:bật).	40uS
Dịch chuyển con trỏ/hiển thị	0	0	0	0	0	1	S/C	R/L	*	*	Dịch chuyển con trỏ/hiển thị qua trái /phải mà không phải đọc/ghi lại dữ liệu (S/C=0:di chuyển con trỏ,S/C=1:di chuyển hiển thị), (R/L=0:dịch trái,R/L=1:dịch phải). nội dung DDRAM không thay đổi.	40uS
Thiết lập chức năng	0	0	0	0	1	DL	N	F	*	*	Khởi tạo giao diện của độ dài dữ liệu (DL=0:độ dài 4 bit,DL=1:8 bit), số hàng hiển thị(N=0:1 hàng ,N=1 :2 hàng) và phông chữ(F=0:5x7,F=1:5x10).	40uS
Thiết lập địa chỉ CGRAM	0	0	0	1	Địa chỉ CGRAM						Thiết lập địa chỉ bộ nhớ tạo kí tự (CGRAM.),dữ liệu được gửi/nhận sau thiết lập này	40uS

Tập lệnh	Mã nhị phân										Mô tả	Thời gian thực thi
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Thiết lập địa chỉ DDRAM	0	0	1	Địa chỉ DDRAM							Thiết lập địa chỉ bộ nhớ tạo kí tự (DDRAM.),dữ liệu được gửi/nhận sau thiết lập này	40uS
Đọc cờ bận	0	1	BF	Địa chỉ CGRAM / DDRAM							Đọc cờ bận Busy-flag (BF),kiểm tra xem hệ thống có đang thực thi 1 lệnh đã được nhận trước đó không.(BF=1:hệ thống đang thực hiện tác vụ bên trong,khi BF=0 thì lệnh tiếp theo mới được thực thi)	0uS
Ghi dữ liệu đến CGRAM/ DDRAM.	1	0	Ghi dữ liệu								Ghi dữ liệu đến CGRAM/ DDRAM.	40uS
Đọc dữ liệu từ CGRAM/ DDRAM.	1	1	Đọc dữ liệu								Đọc dữ liệu từ CGRAM/ DDRAM.	40uS

Mã (Hex)	Lệnh đến thanh ghi của LCD
1	Xoá màn hình hiển thị
2	Trở về đầu dòng
4	Giả con trỏ (dịch con trỏ sang trái)
6	Tăng con trỏ (dịch con trỏ sang phải)
5	Dịch hiển thị sang phải
7	Dịch hiển thị sang trái
8	Tắt con trỏ, tắt hiển thị
A	Tắt hiển thị, bật con trỏ
C	Bật hiển thị, tắt con trỏ
E	Bật hiển thị, nhấp nháy con trỏ
F	Tắt con trỏ, nhấp nháy con trỏ
10	Dịch vị trí con trỏ sang trái
14	Dịch vị trí con trỏ sang phải
18	Dịch toàn bộ hiển thị sang trái

PHỤ LỤC 2. THƯ VIỆN LCD

```
#include <stddef.h>
#use delay(clock=200000000)
#define LCD_RS          PIN_D0
//#define LCD_RW        PIN_D2
#define LCD_EN          PIN_D1
#define LCD_B4          PIN_D4
#define LCD_B5          PIN_D5
#define LCD_B6          PIN_D6
#define LCD_B7          PIN_D7
// misc display defines-
#define Line_1          0x80
#define Line_2          0xC0
#define Clear_Scr       0x01
// prototype statements
#separate void LCD_Init ( void );// ham khoi tao LCD
#separate void LCD_SetPosition ( unsigned int cX );//Thiet lap vi tri con tro
#separate void LCD_PutChar ( unsigned int cX );// Ham viet1kitu/1chuoi len LCD
#separate void LCD_PutCmd ( unsigned int cX) ;// Ham gui lenh len LCD
#separate void LCD_PulseEnable ( void );// Xung kich hoat
#separate void LCD_SetData ( unsigned int cX );// Dat du lieu len chan Data
// D/n Cong
#use standard_io ( B )
#use standard_io (A)
//khoi tao LCD**********************************************
#separate void LCD_Init ( void )
```

```
    {
    LCD_SetData ( 0x00 );
    delay_ms(200);       /* wait enough time after Vdd rise >> 15ms */
    output_low ( LCD_RS );// che do gui lenh
    LCD_SetData ( 0x03 );   /* init with specific nibbles to start 4-bit mode */
    LCD_PulseEnable();
    LCD_PulseEnable();
    LCD_PulseEnable();
    LCD_SetData ( 0x02 );   /* set 4-bit interface */
    LCD_PulseEnable();      /* send dual nibbles hereafter, MSN first */
    LCD_PutCmd ( 0x2C );    /* function set (all lines, 5x7 characters) */
    LCD_PutCmd ( 0b00001100);    /* display ON, cursor off, no blink */
    LCD_PutCmd ( 0x06 );    /* entry mode set, increment & scroll left */
    LCD_PutCmd ( 0x01 );    /* clear display */
    }
#separate void LCD_SetPosition ( unsigned int cX )
    {
    /* this subroutine works specifically for 4-bit Port A */
    LCD_SetData ( swap ( cX ) | 0x08 );
    LCD_PulseEnable();
    LCD_SetData ( swap ( cX ) );
    LCD_PulseEnable();
    }
#separate void LCD_PutChar ( unsigned int cX )
    {
    /* this subroutine works specifically for 4-bit Port A */
        output_high ( LCD_RS );
```

```
      LCD_PutCmd( cX );
      output_low ( LCD_RS );
   }
#separate void LCD_PutCmd ( unsigned int cX )
   {
   /* this subroutine works specifically for 4-bit Port A */
   LCD_SetData ( swap ( cX ) );     /* send high nibble */
   LCD_PulseEnable();
   LCD_SetData ( swap ( cX ) );     /* send low nibble */
   LCD_PulseEnable();
   }
#separate void LCD_PulseEnable ( void )
   {
   output_high ( LCD_EN );
   delay_us ( 3 );         // was 10
   output_low ( LCD_EN );
   delay_ms ( 3 );         // was 5
   }
#separate void LCD_SetData ( unsigned int cX )
   {
   output_bit ( LCD_B0, cX & 0x01 );
   output_bit ( LCD_B1, cX & 0x02 );
   output_bit ( LCD_B2, cX & 0x04 );
   output_bit ( LCD_B3, cX & 0x08 );
   }
```

Phụ lục 3 Chương trình đo và hiện thị kết quả đo

```
#include <DoNhietDo_LM35.h>
void main()
{
float GiaTriadc;
float Voltage;
float NhietDo;
set_tris_a(1);
set_tris_d(0);
lcd_init();
lcd_clear();
lcd_gotoxy(1,1);
lcd_putc("Do Nhiet Do--LM35");

setup_adc(ADC_CLOCK_INTERNAL);
setup_adc_ports(ALL_ANALOG);
set_adc_channel(0);
  while(TRUE)
  {
    //TODO: User Code
    GiaTriadc = read_adc();
    Voltage = GiaTriadc*5/1023;
    NhietDo = Voltage*100;
    /* ta co cong thuc:
    gia tri 5v duoc chia thanh 1023 muc adc
    nen moi muc adc = 5/1023(V)
    >>> so voltage do duoc = (5/1023).GiaTriadc
```

```
      vi LM35 co: 10mV = 1*C
      >>Nhiet Do = Voltage*100 */
          lcd_gotoxy(1,2);
      printf(lcd_putc,"Nhiet Do:%02.2f",NhietDo);
       if(NhietDo>=35)
      {
      output_high(PIN_B0);
      output_low(PIN_B1);
      }
      else
     {
      output_high(PIN_B1);
      output_low(PIN_B0);
     }
   }
}
```

Tài liệu tham khảo

1. Ngô Diên Tập(2006), Vi điều khiển với lập trình C, NXB Khoa học và Kỹ thuật.
2. Vũ Quý Điềm(2006), Cơ Sở Kỹ Thuật Đo Lường Điện Tử, NXB Khoa học và Kỹ thuật.
3. Nguyễn Vũ Quỳnh, Phạm Quang Huy, "Giáo trình đo lường cảm biến (Lý thuyết – Thực hành)", NXB Thanh Niên, 2020.
4. TS Đoàn Hữu Chức, Báo cáo tổng hợp kết quả nghiên cứu khoa học đề tài "Ứng dụng cảm biến qua Arduino".
5. Datasheet PIC 16F877A
6. Các bài viết trên các diễn đàn:

- ✓ http://www.picvietnam.com/forum/
- ✓ http://www.dientuvietnam.net/forums/