

**TRƯỜNG ĐẠI HỌC TÔN DỨC THẮNG
KHOA ĐIỆN – ĐIỆN TỬ
NGÀNH TỰ ĐỘNG ĐIỀU KHIỂN**



ĐỒ ÁN 1
ĐỀ TÀI : KHẢO SÁT NHIỆT ĐỘ ẨM CÓ CHỨC NĂNG
HIỂN THỊ TRONG PHÒNG.

Giáo viên hướng dẫn	: ThS. Đỗ Phạm Hoàng
Sinh viên thực hiện	: Nguyễn Triệu Vỹ
Mã số sinh viên	: 41203202
Lớp	: 12040301
Khoá	: 16

HCM 04 – 2015

CHƯƠNG 1 : TỔNG QUAN VỀ ĐỀ TÀI

GIỚI THIỆU ĐỀ TÀI

1.1 Tính thực tế và ý nghĩa của đề tài.

Trong cuộc sống hiện nay có rất nhiều yếu tố ảnh hưởng đến quá trình sản xuất và con người. Trong đó nhiệt độ và độ ẩm cũng là yếu tố được đề cập tới rất nhiều, vì thế mạch đo nhiệt độ và độ ẩm ra đời là sự tất yếu. Với sự phát triển của công nghệ hiện nay việc sản xuất mạch đo nhiệt độ và độ ẩm đơn giản mà độ chính xác cao là điều khá đơn giản. Việc áp dụng trong thực tế càng phổ biến hơn khi nhiệt độ độ ẩm đóng vai trò lớn ảnh hưởng đến con người, hàng hoá, máy móc, độ bền ... Vì vậy nhiệt độ được sử dụng hầu hết các nhà máy sản xuất, kho chứa hàng hoá, bảo quản máy móc cũng như điều kiện vận hành. Trong y tế nó được sử dụng rộng rãi trong các phòng cách ly, phòng điều trị cho bệnh nhân cũng như áp dụng hầu hết trong các dây chuyền cũng như các công nghệ sản xuất tùy theo nhu cầu mà chúng ta có thể tùy biến thêm ngoài chức năng của đề tài này để phù hợp với đúng nhu cầu trong quá trình hoạt động ngoài chức năng chính là hiển thị nhiệt độ độ ẩm của các khu vực cần khảo sát nhiệt độ và độ ẩm.

Với đề tài này 1 bộ mạch “Đo nhiệt độ và độ ẩm” ta có thể đo cùng lúc nhiệt độ và độ ẩm có thể phát triển hiệu quả hơn, nhưng đề tài chỉ trong phạm vi là Đồ án 1 nên tính hiệu quả thực tế cũng như tính chính xác của mạch không cao.

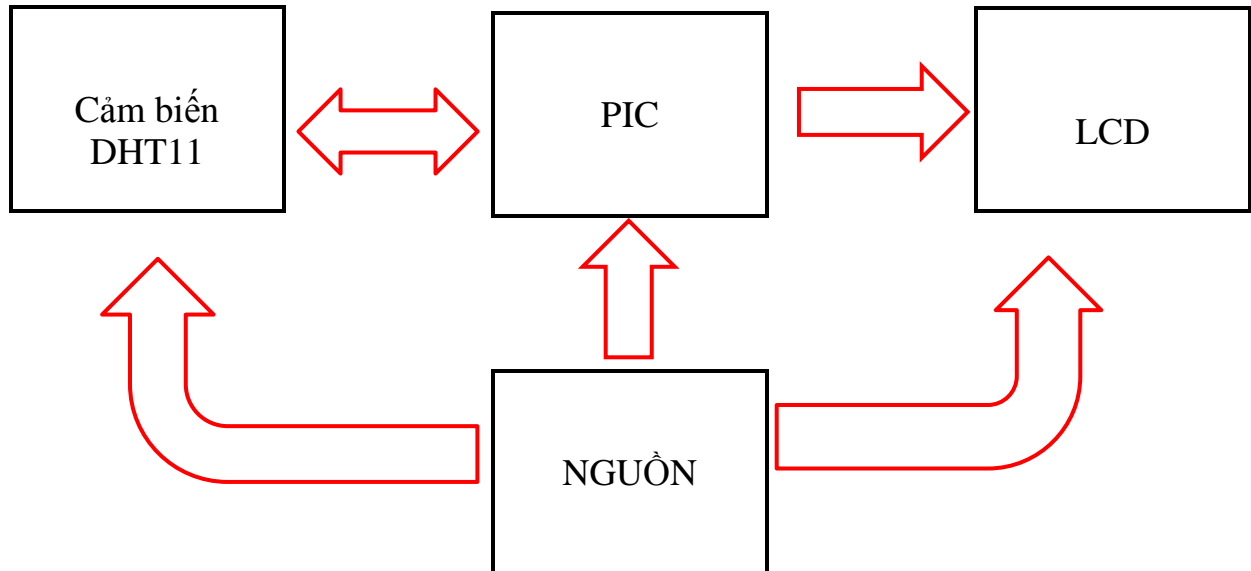
1.2 Sơ lược về hệ thống và sơ đồ khối của hệ thống.

1.2.1 Sơ lược hệ thống

Bộ mạch được điều khiển bởi vi điều khiển PIC16F628A đóng vai trò điều khiển và nhập xuất dữ liệu từ các thiết bị giao tiếp với nó, điển hình như cảm biến nhiệt độ và độ ẩm DHT11 được giao tiếp với vi điều khiển pic16f628a và xuất nhập dữ liệu đọc nhiệt độ và độ ẩm từ cảm biến sau đó giao tiếp với thiết bị LCD mã hoá dữ liệu để hiển thị kết quả ra màn hình với dữ liệu của cảm biến có dải nhiệt độ từ $0-50^{\circ}\text{C}$ và độ ẩm từ 20%- 95%.

Phần cứng sau khi thi công hoàn thành thì vi điều khiển được nạp chương trình qua các lệnh được thiết kế bằng ngôn ngữ C như những tập lệnh điều khiển vi điều khiển để mạch hoạt động một cách trơn tru.

1.2.2 Sơ đồ khối



Muốn cảm biến gửi tín hiệu đầu tiên Vi điều khiển PIC sẽ gửi tín hiệu muốn đo đến cảm biến DHT11 sau khi xác nhận được..

Khối cảm biến sẽ nhận nhiệt độ và độ ẩm từ môi trường bên ngoài để gửi tín hiệu dạng sóng 5byte dữ liệu và nhiệt độ đo được đến vi điều khiển PIC.

Sau đó Vi điều khiển PIC sẽ đọc dữ liệu và gửi trực tiếp đến bộ màn hình LCD, qua bộ giải mã được tích hợp trong LCD thì nhiệt độ và độ ẩm được hiển thị trên màn hình.

Nguồn được dùng để cung cấp áp cho hoạt động của Vi điều khiển, Cảm biến, Màn hình LCD hoạt động ổn định.

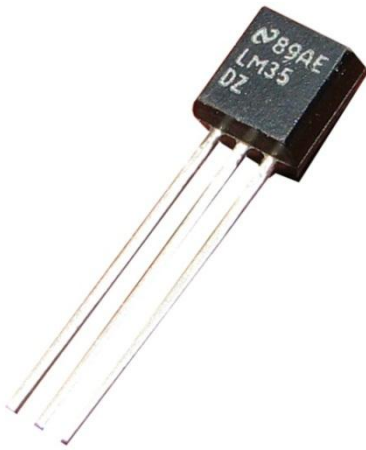
CHƯƠNG 2 : THIẾT KẾ

2.1 CÁC CHỨC NĂNG CỦA KHỐI.

2.1.1 khối cảm biến.

Hiện nay trên thị trường có rất nhiều cảm biến đo nhiệt độ và độ ẩm như:

- LM35, LM355, PT100... : cảm biến đo nhiệt độ.

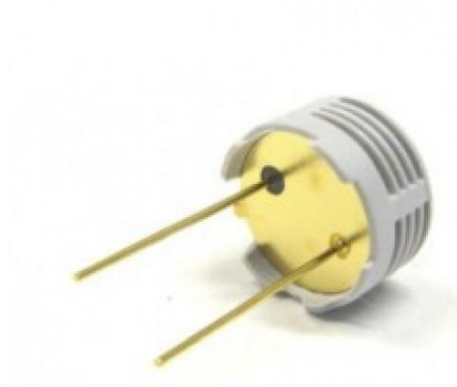


Độ chính xác thực tế: $1/4^{\circ}\text{C}$ ở nhiệt độ phòng và $3/4^{\circ}\text{C}$ ngoài khoảng -55°C tới 150°C

LM35 có hiệu năng cao, công suất tiêu thụ là $60\mu\text{A}$, giá thành rẻ.

Hình 2.1 cảm biến LM35

- HS1011, HS220... : cảm biến đo độ ẩm.



HS1101 cảm biến độ ẩm có độ chính xác khá cao.

Hình 2.2 cảm biến hs1101

- DHT11, DHT21, SHT11 : tích hợp của cả cảm biến nhiệt độ và độ ẩm



SHT11 có độ sai số nhỏ, phù hợp với khảo sát ở những nơi cần độ chính xác cao như nhà máy, phòng thí nghiệm.

Hình 2.3 cảm biến SHT11

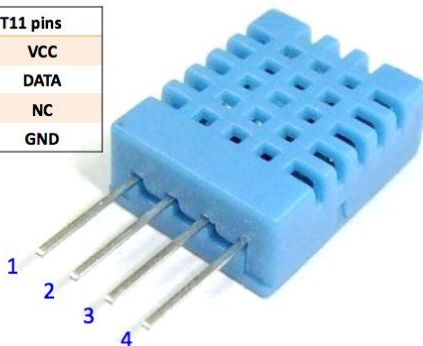
Nhưng ở trong phạm vi Đồ Án 1, để phù hợp và tiện dụng hơn thì ta sử dụng cảm biến tích hợp nhiệt độ và độ ẩm là DHT11, cũng như giá thành rẻ hơn nhưng thiết bị khác nhưng có khuyết điểm là tính chính xác tương đối, mà ở Đồ Án 1 chỉ là minh họa qua thiết bị và tìm hiểu chức năng cũng như ý nghĩa của đề tài, không sử dụng vào mục đích thực tiễn nên không cần độ chính xác cao.

Vì vậy chọn DHT11 là một lựa chọn có thể đảm bảo hai chức năng mà giá thành tương đối.

DHT11

Cảm biến độ ẩm và nhiệt độ DHT11 là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1 wire (giao tiếp digital 1 dây truyền dữ liệu duy nhất). Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không phải qua bất kỳ tính toán nào.

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



Nguồn: 3 -> 5 VDC.

Dòng sử dụng: 2.5mA max (khi truyền dữ liệu).

Đo tốt ở độ ẩm 20-80%RH với sai số 5%.

Đo tốt ở nhiệt độ 0 to 50°C sai số $\pm 2^{\circ}\text{C}$.

Tần số lấy mẫu tối đa 1Hz (1 giây 1 lần)

Kích thước 15mm x 12mm x 5.5mm.

4 chân, khoảng cách chân 0.1".

Hình 2.4 cảm biến DHT11.

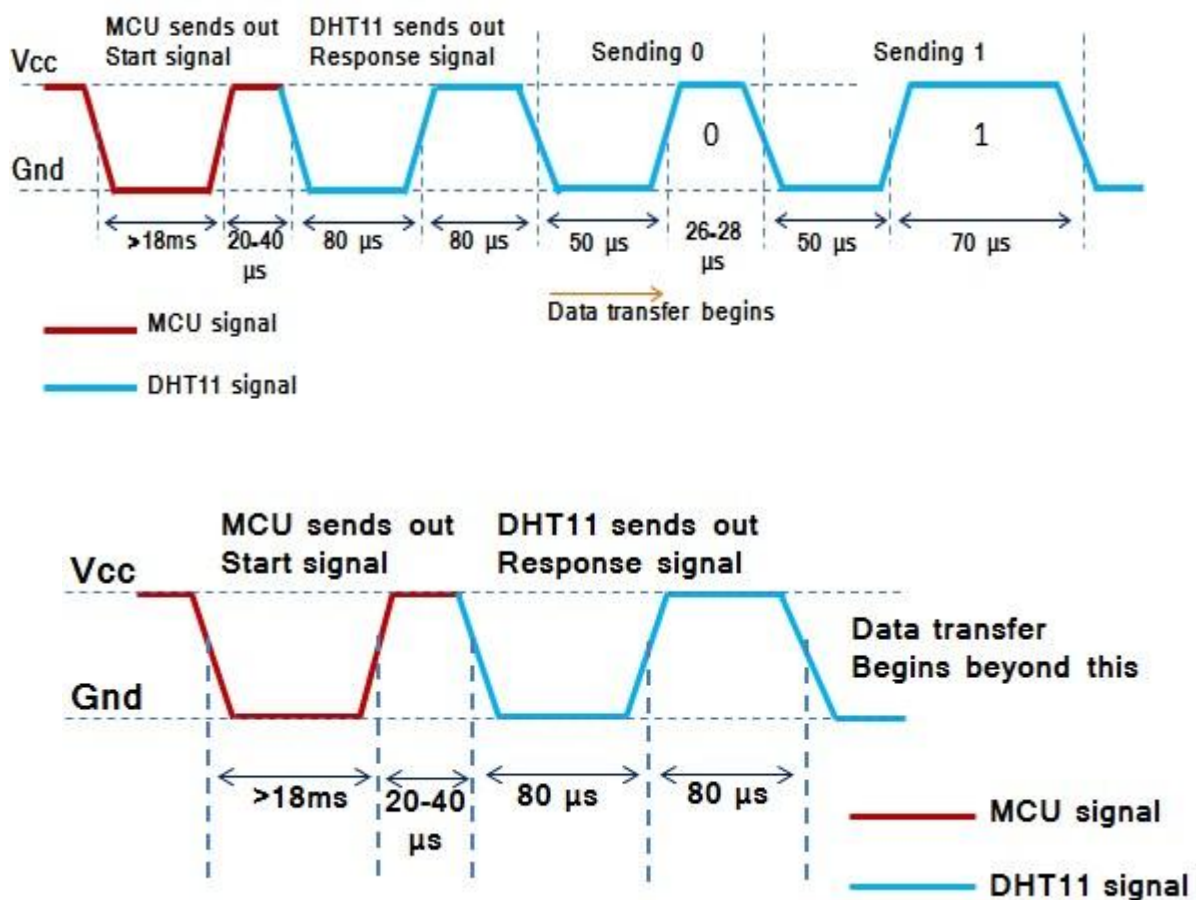
Nguyên lý hoạt động

Để có thể giao tiếp với DHT11 theo chuẩn 1 chân vi xử lý thực hiện theo 2 bước:

Gửi tín hiệu muốn đo (Start) tới DHT11, sau đó DHT11 xác nhận lại.

Khi đã giao tiếp được với DHT11, Cảm biến sẽ gửi lại 5 byte dữ liệu và nhiệt độ đo được.

- Bước 1: gửi tín hiệu Start



Hình 2.5 dạng tín hiệu giao tiếp với cảm biến.

MCU thiết lập chân DATA là Output, kéo chân DATA xuống 0 trong khoảng thời gian $>18\text{ms}$. Trong Code mình để 25ms . Khi đó DHT11 sẽ hiểu MCU muốn đo giá trị nhiệt độ và độ ẩm.

MCU đưa chân DATA lên 1, sau đó thiết lập lại là chân đầu vào.

Sau khoảng $20\text{--}40\text{ }\mu\text{s}$, DHT11 sẽ kéo chân DATA xuống thấp. Nếu $>40\text{ }\mu\text{s}$ mà chân DATA không được kéo xuống thấp nghĩa là không giao tiếp được với DHT11.

Chân DATA sẽ ở mức thấp $80\text{ }\mu\text{s}$ sau đó nó được DHT11 kéo lên cao trong $80\text{ }\mu\text{s}$. Bằng việc giám sát chân DATA, MCU có thể biết được có giao tiếp

được với DHT11 không. Nếu tín hiệu đo được DHT11 lên cao, khi đó hoàn thiện quá trình giao tiếp của MCU với DHT.

- Bước 2: đọc giá trị trên DHT11

DHT11 sẽ trả giá trị nhiệt độ và độ ẩm về dưới dạng 5 byte. Trong đó:

Byte 1: giá trị phần nguyên của độ ẩm (RH%)

Byte 2: giá trị phần thập phân của độ ẩm (RH%)

Byte 3: giá trị phần nguyên của nhiệt độ (TC)

Byte 4 : giá trị phần thập phân của nhiệt độ (TC)

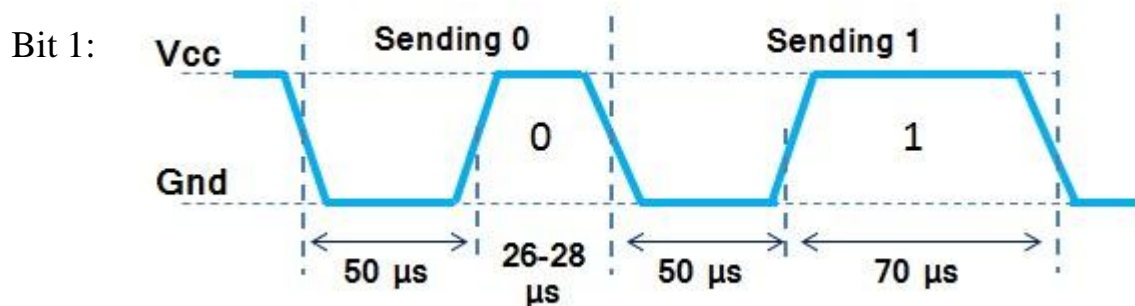
Byte 5 : kiểm tra tổng.

Nếu Byte 5 = (8 bit) (Byte1 +Byte2 +Byte3 + Byte4) thì giá trị độ ẩm và nhiệt độ là chính xác, nếu sai thì kết quả đo không có nghĩa.

Đọc dữ liệu:

Sau khi giao tiếp được với DHT11, DHT11 sẽ gửi liên tiếp 40 bit 0 hoặc 1 về MCU, tương ứng chia thành 5 byte kết quả của Nhiệt độ và độ ẩm.

Bit 0:



Hình 2.6 dạng tín hiệu khi ở bit 0.

Sau khi tín hiệu được đưa về 0, ta đợi chân DATA của MCU được DHT11 kéo lên 1. Nếu chân DATA là 1 trong khoảng 26-28 us thì là 0, còn nếu tồn tại 70us là 1. Do đó trong lập trình ta bắt sườn lên của chân DATA, sau đó delay 50us. Nếu giá trị đo được là 0 thì ta đọc được bit 0, nếu giá trị đo được là 1 thì giá trị đo được là 1. Cứ như thế ta đọc các bit tiếp theo.

2.1.2 Khối điều khiển

Ngày nay, những ứng dụng của vi điều khiển đã đi sâu vào đời sống sinh hoạt và sản xuất của con người. Thực tế hiện nay là hầu hết các thiết bị điện hiện nay đều có sự góp mặt của vi điều khiển và vi xử lý. Ứng dụng vi điều khiển trong thiết kế hệ thống làm giảm chi phí thiết kế và hạ giá thành sản phẩm đồng thời nâng cao tính ổn định của thiết bị và hệ thống.

Trên thị trường hiện nay có nhiều họ vi điều khiển để lựa chọn như: 8051, Motola68HC, AVR, ARM, Pic...và có lẽ 8051 là dòng mà chúng ta được làm quen nhiều nhất trong môi trường đại học nhưng tại sao chúng ta

chọn dòng vi điều khiển Pic để thực hiện ứng dụng và phát triển không ngoài những nguyên nhân sau:

- Họ vi điều khiển Pic phát triển và sử dụng phổ biến ở nước ta
=> việc tìm mua và trao đổi kinh nghiệm là hết sức thuận lợi.
- Giá thành các dòng Pic là không quá mắc.
- Các dòng Pic có đầy đủ tính năng để hoạt động độc lập.
- Là sự bổ sung hợp lý về kiến thức cũng như ứng dụng cho họ vi điều khiển 8051
- Có sự hỗ trợ cao của nhà sản xuất về các công cụ lập trình, trình biên dịch, mạch nạp Pic từ đơn giản tới phức tạp. Không những vậy các tính năng đa dạng của các dòng Pic không ngừng được phát triển.
- Có nhiều bộ phận ngoại vi ngay trên chip, bao gồm: Cổng và/ra số, bộ biến đổi ADC, bộ nhớ EEPFROM, bộ định thời, bộ điều chế độ rộng xung (PWM)...
- Bộ nhớ chương trình và bộ nhớ dữ liệu được tích hợp ngay trên chip. Đây là họ VĐK được chế tạo theo kiến trúc RISC (Reduced Intruction Set Computer) có cấu trúc khá phức tạp. Ngoài các tính năng như các họ VĐK khác, nó còn tích hợp nhiều tính năng mới rất tiện lợi cho người thiết kế và lập trình.

Các dòng và cách lựa chọn

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 PIC 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 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

PIC16F628A nằm trong dòng sản phẩm PIC16F627A/628A/648A của nhà sản xuất Microchip với đặc điểm 28/40/44 -Pin Enhanced Flash Microcontrollers with 10- Bit A/D and nanoWatt Technology. Dòng sản phẩm này có nhiều cải tiến đáng kể về tính năng so các dòng Pic trước đó như :

- Bộ nhớ chương trình được tăng cường
- Tăng cường modul CCP.

2.1.2.a Sơ lược về PIC16F628A



Hình 2.7 PIC16f628A

Vi điều khiển Pic16F628A có các đặc điểm cơ bản:

- Sử dụng công nghệ nanoWatt: Hiệu năng cao, tiêu thụ năng lượng ít
- Kiến trúc RISC
 - + 75 lệnh mạnh, hầu hết các lệnh thực hiện trong bốn chu kì xung.
 - + Tốc độ thực hiện lên tới 10 triệu lệnh trong 1s với tần số

40Mhz

+ Có bộ nhân cứng .

- Các bộ nhớ chương trình và dữ liệu cố định

+ 3.5 Kbytes bộ nhớ flash có khả năng tự lập trình trong hệ thống có thể thực hiện được 100.000 lần ghi/xóa

+ 128 bytes EEPROM có thể thực hiện được 1.000.000 lần ghi/xóa-+224 bytes SRAM

- Những ngoại vi tiêu biểu

+ 4 bộ định thời/bộ đếm 8 bit với các chế độ tỉ lệ đặt trước và chế độ so sánh.

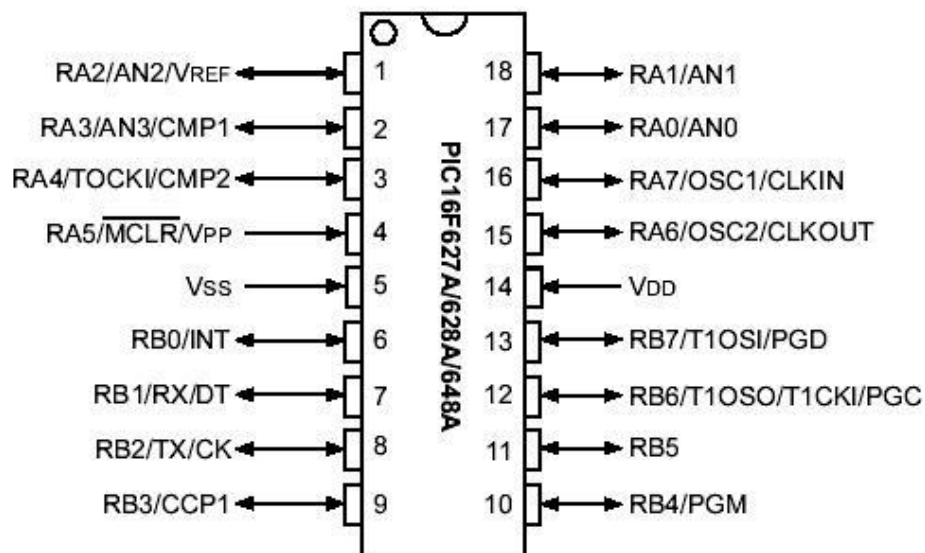
+ Bộ đếm thời gian thực với bộ tạo dao động riêng biệt + 2 kênh PWM

+ 13 kênh ADC 10 bit

+ Bộ truyền tin nối tiếp USART khả trình

+ Watchdog Timer khả trình với bộ tạo dao động bên trong riêng biệt + Bộ so sánh tương tự

2.1.2.b Sơ đồ chân của Vi điều khiển và chức năng



Hình 2.8 sơ đồ chân vi điều khiển

Sau đây là giới thiệu cấu tạo chân loại 18 chân (18 Pin PDIP):

• Chân 4(RA5/MCLR/Vpp/RE3) :

- *MCLR* là đầu vào Master Clear (reset) hoạt động ở mức thấp để reset toàn bộ thiết bị.
- *VPP* dùng để thay đổi điện áp đầu vào. - *RE3* đầu vào số.

Các chân thuộc cổng vào ra Port A

- Chân 17(RA0/AN0): với RA0 là cổng vào ra số, AN0 là đầu vào tương tự Input0.
- Chân 18(RA1/AN1): RA1 là cổng vào ra số, AN1 là đầu vào tương tự Input1.
- Chân 19(RA2/AN2/VREF+): RA2 là cổng vào ra số, AN2 là đầu vào tương tự Input2. VREF+ đầu vào tương tự chuyển đổi A/D điện áp tham chiếu(mức thấp), còn CVREF là đầu ra tương tự để so sánh điện áp chuẩn.
- Chân 20(RA3/AN3/CMP1): RA3 là cổng vào ra số, AN3 là đầu vào tương tự Input3.
- Chân 21(RA4/T0CKI/CMP2): RA4 là đầu vào ra số, T0CKI đầu vào xung bên ngoài của Timer0, C1OUT là đầu ra bộ so sánh 1.
- Chân 22(OSC1/CLKI/RA7): với OSC1 là đầu vào bộ dao động thạch anh hoặc là đầu vào nguồn xung từ bên ngoài, khi ta nối dây với các thiết bị tương tự thì đầu vào này dạng ST(Schmitt Trigger input with CMOS levels).CLKI là đầu vào CMOS cho nguồn xung bên ngoài và luôn được ghép nối với chân OSC1. Còn RA7 là chân vào ra sử dụng chung .
- Chân 23(OSC2/CLKO/RA6): OSC2 là đầu ra bộ dao động thạch anh được nối với thạch anh hoặc bộ công hưởng để lựa chọn dạng bộ dao động thạch anh. CLK0 có tần số bằng $\frac{1}{4}$ tần số của OSC1 độ rộng chu kỳ lệnh, RA6 là đầu vào ra chung.

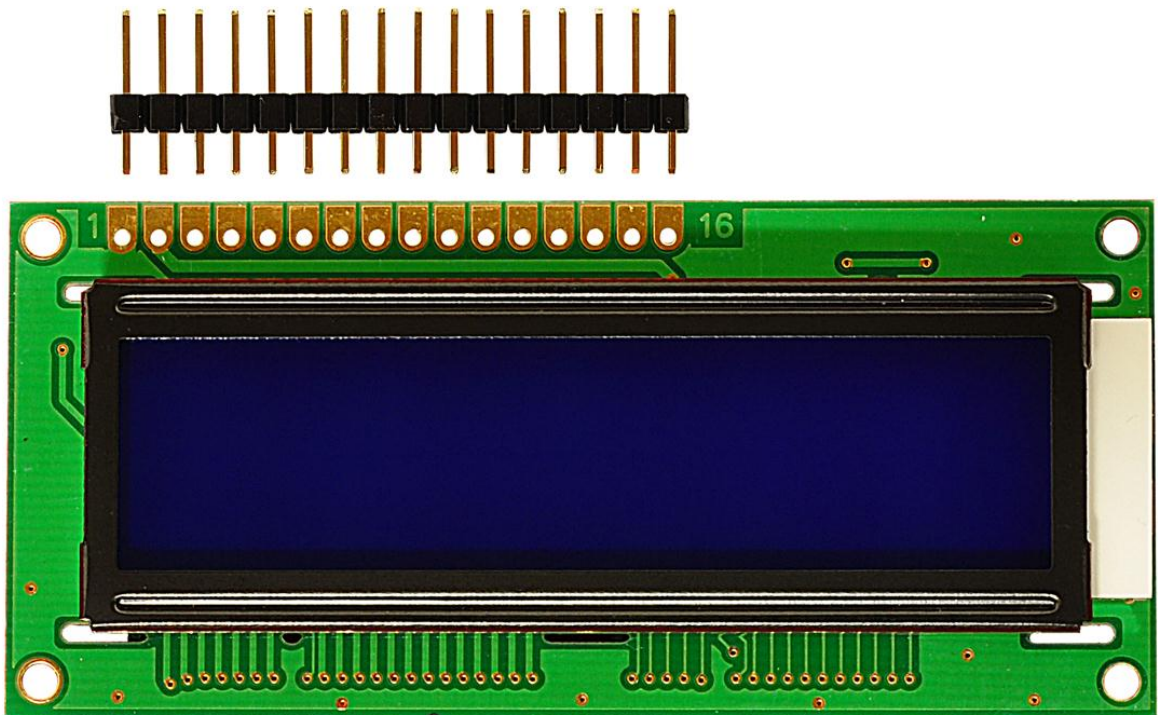
Các chân cổng vào ra hai chiều Port B. Port B có thể lập trình bằng phần mềm khi cho kéo đầu vào bên trong yếu lên trên toàn bộ đầu vào.

- Chân 6(RB0/INT0): Với RB0 là cổng vào ra số, INT0 là đầu vào ngắt ngoài Interrupt 0, FLT0 là đầu vào báo lỗi PWM được tăng cường CCP1, AN12 đầu vào tương tự Input 12.
- Chân 7(RB1/RX/DT): RB1 là đầu vào ra số
- Chân 8(RB2/DX/CK): RB2 là đầu vào ra số
- Chân 9 (RB3/CCP1): RB3 là đầu vào ra số.
- Chân 10(RB4/PGM): RB4 là đầu vào ra số.
- Chân 11(RB5): RB5 đầu vào ra số

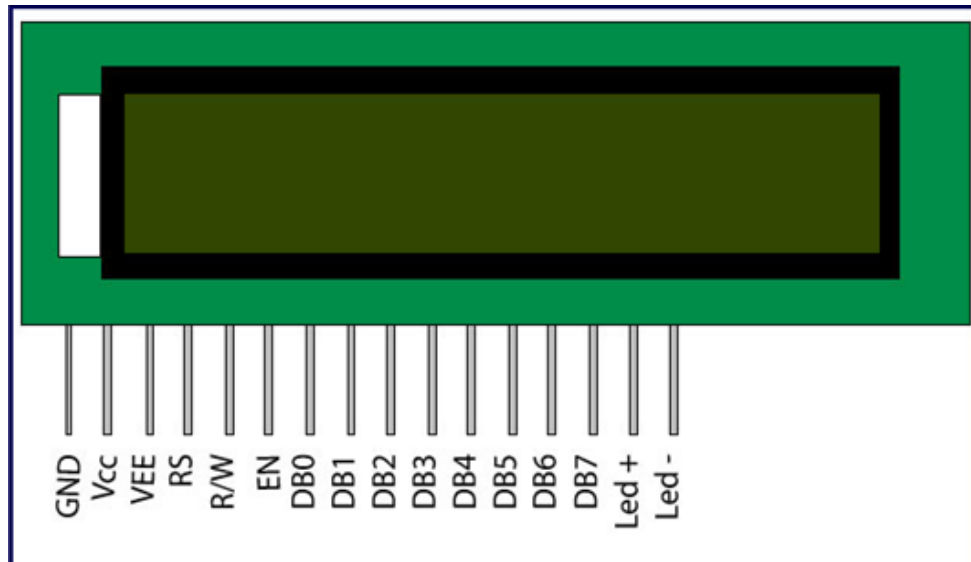
- Chân 12(RB6/T1OSO/PGC): RB6 là đầu vào ra số, KBI2 thay đổi mở ngắt, PGC chân dùng trong mạch chạy và xung lập trình ICSP.
- Chân 13(RB7/TIOSI/PGD): RB7 đầu vào ra số, KBI3 thay đổi mở ngắt, PGD chân dùng trong mạch chạy và xung lập trình ICSP.
- Chân (Vss) nối đất chuẩn cho I/O và logic.
- Chân (Vdd) cung cấp nguồn dương cho I/O và logic

2.1.3. Khối hiển thị

Khối hiển thị LCD 2x16(2 dòng, 16cột)



Hình 2.9 màn hình LCD thực tế
LCD làm việc chế độ 8 bit ghép nối Port D của vi điều khiển trung tâm



Hình 2.10 Sơ đồ chân LCD

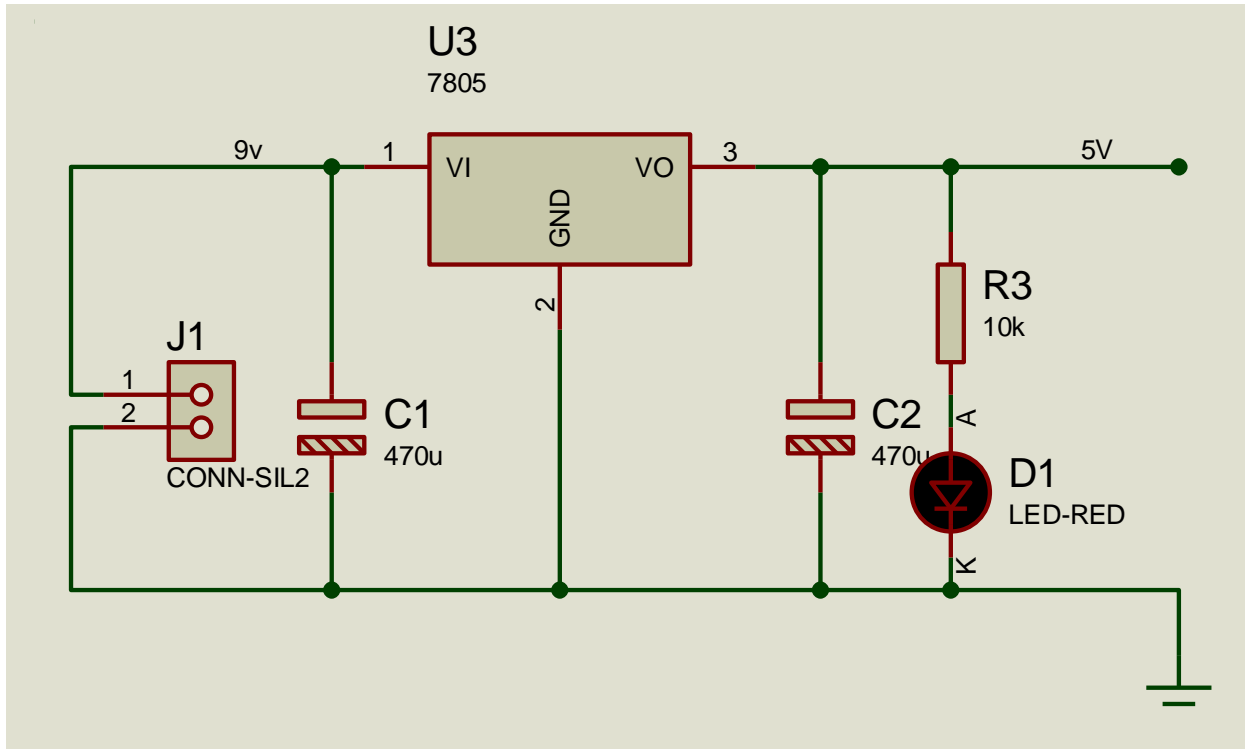
- Sử dụng một biến trở 10K điều chỉnh độ tương phản của LCD.

Chân	Kí hiệu	Chức Năng
1	VSS	Mass
2	VDD	Nguồn cung cấp cho LCD
3	VEE	Chỉnh độ tương phản
4	RS	Chọn thanh ghi trong LCD
5	RW	Đọc và ghi dữ liệu
6	E	Cho phép chọn LCD
7	D0	Bit 0 của byte dữ liệu
8	D1	Bit 1 của byte dữ liệu
9	D2	Bit 2 của byte dữ liệu
10	D3	Bit 3 của byte dữ liệu
11	D4	Bit 4 của byte dữ liệu
12	D5	Bit 5 của byte dữ liệu
13	D6	Bit 6 của byte dữ liệu
14	D7	Bit 7 của byte dữ liệu
15	BL-	Nguồn cho LED nền LCD
16	BL+	Mass cho LED nền LCD

Bảng 2.1 Các kí hiệu và ý nghĩa chân của LCD

- Chỉ dùng LCD để hiển thị (Write) nên chân R/W được nối mass.

2.1.4 Khối nguồn.

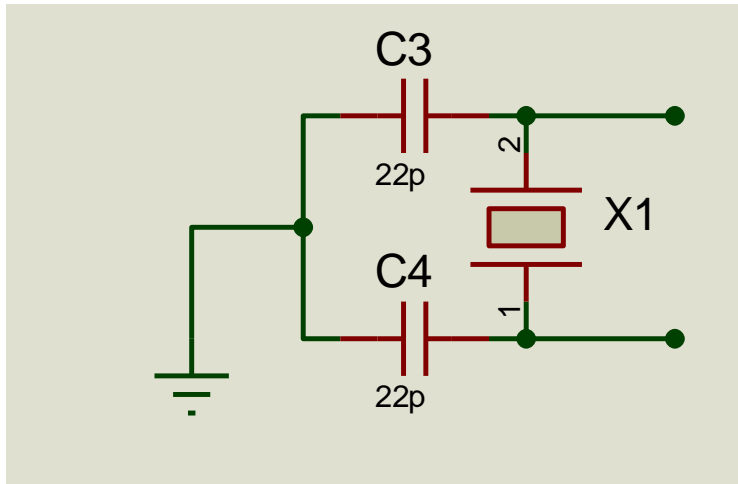


Hình 2.11 khối nguồn nuôi.

Trong mạch sử dụng nguồn pin 9V chuyển đổi qua LM7805 xuống thành nguồn 5V cung cấp cho mạch.

- Nguồn ra được đấu song song với tụ 470uF để chống nhiễu.
- Với led được mắc song song để dễ dàng biết khối nguồn đã hoạt động cũng như đã có nguồn nuôi linh kiện cho mạch

2.1.5. Khối cấp xung



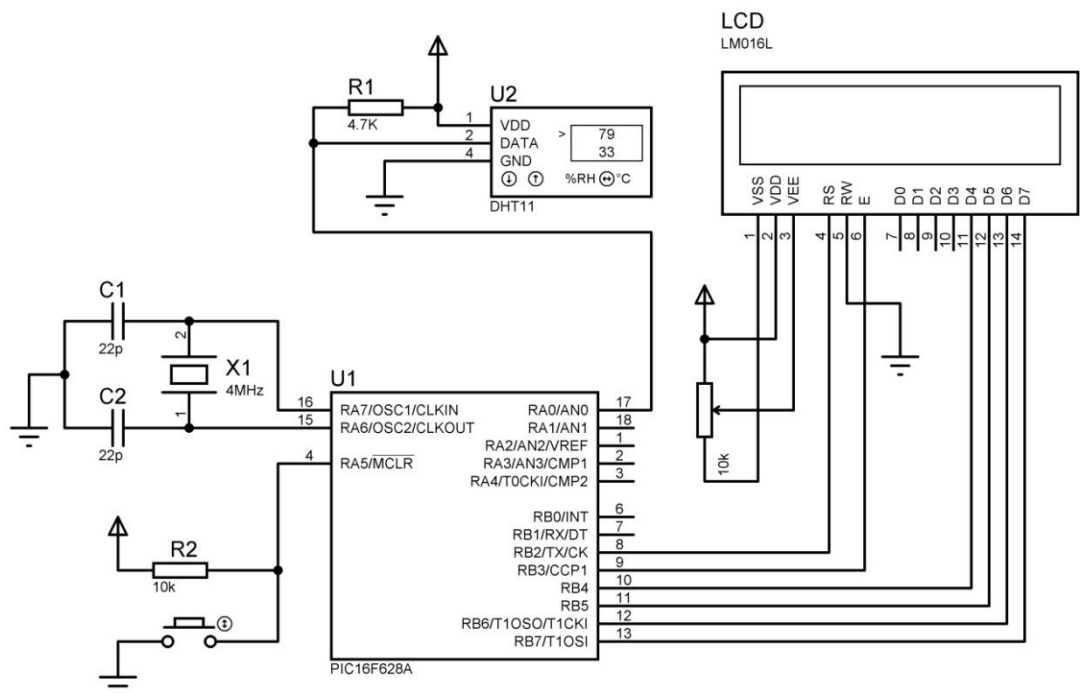
Hình 1.12 Khối cấp xung

Khối cấp xung là khối thiết yếu để cho vi điều khiển hoạt động với tụ thạch anh 4 Mhz được mắc cùng với 2 tụ ổn định 22pF.

Hai ngõ ra được mắc vào hai chân 15(osc 1) và 16(osc2) để cung cấp xung cho vi điều khiển làm việc.

2.2. Nguyên Lý hoạt động.

Mạch được mô phỏng



Hình

2.14 mạch nguyên lý

Thiết kế như trên hình, với nguyên lý hoạt động cũng khá đơn giản. Các khối nguồn và mạch tạo xung hoạt động và cấp nguồn 5V để giúp giúp linh kiện trên mạch hoạt động đặc biệt là vi điều khiển. Mạch hoạt động dựa trên nguyên lý:

- Khi có nguồn nuôi mạch, vi điều khiển sẽ xuất dữ liệu đến cảm biến cũng như tương tác với cảm biến sau đó cảm biến sẽ gửi dữ liệu qua vi điều khiển và vi điều khiển xử lý và mã hoá sau đó xuất tín hiệu đến màn hình hiển thị LCD.

CHƯƠNG 3 : MÔ PHỎNG

3.1 Mô phỏng trên phần mềm Protues.

Phần mềm protues là phần mềm chuyên dụng cho việc thiết kế mạch điện tử, cũng như mô phỏng các đề tài về điện tử.

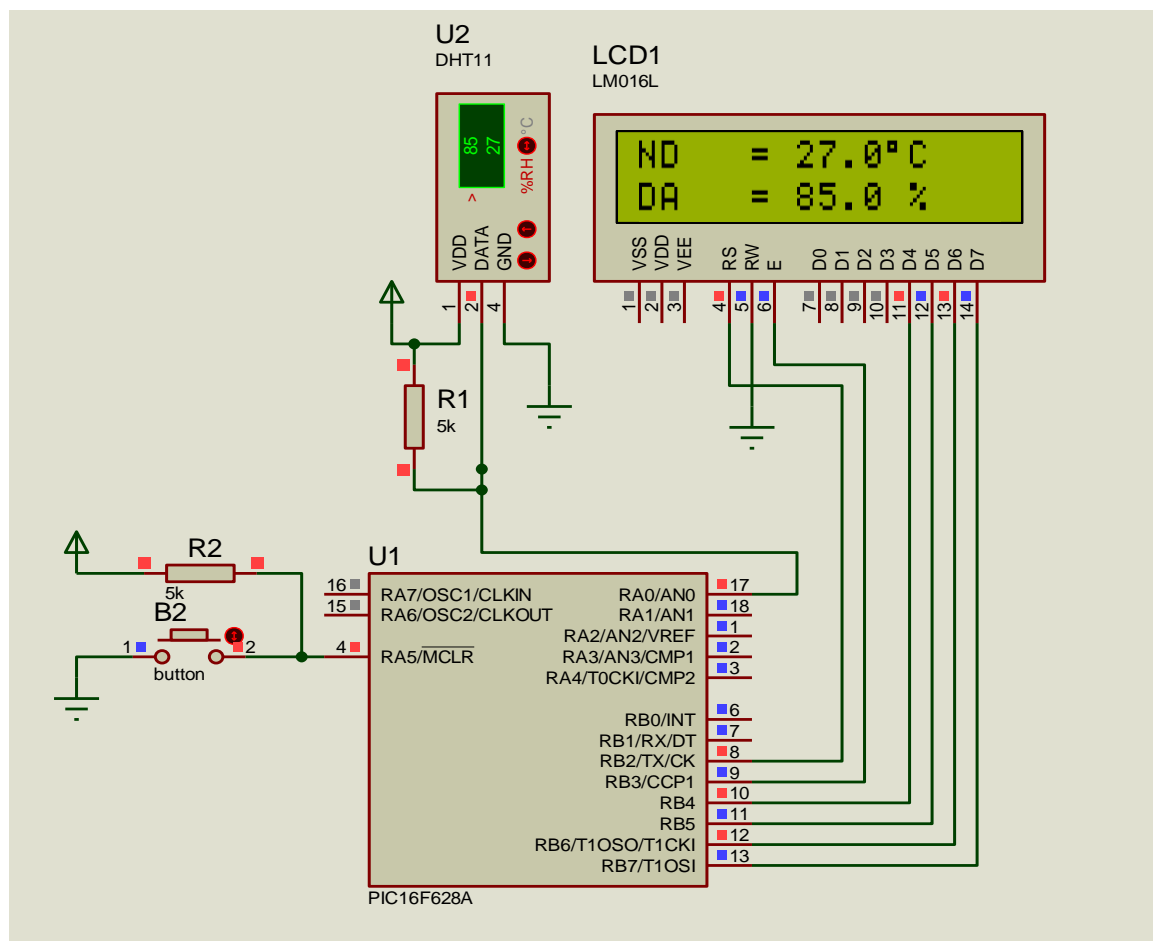
Một số ưu điểm của phần mềm này như:

- Thư viện linh kiện phong phú.
- Hỗ trợ các thiết bị đo.
- Hỗ trợ thiết kế mạch in.
- Cho phép chạy mô phỏng vi điều khiển họ AT89, PIC...

Vì vậy Protues là phần mềm có ưu điểm vượt trội để dễ dàng hỗ trợ thực hiện các thiết kế, mô phỏng trước khi lắp đặt nhằm khắc phục tối đa các lỗi có thể xảy ra.

Dưới đây là mạch mô phỏng đo nhiệt độ và độ ẩm.

3.1.1. Mạch mô phỏng khi làm việc bình thường.

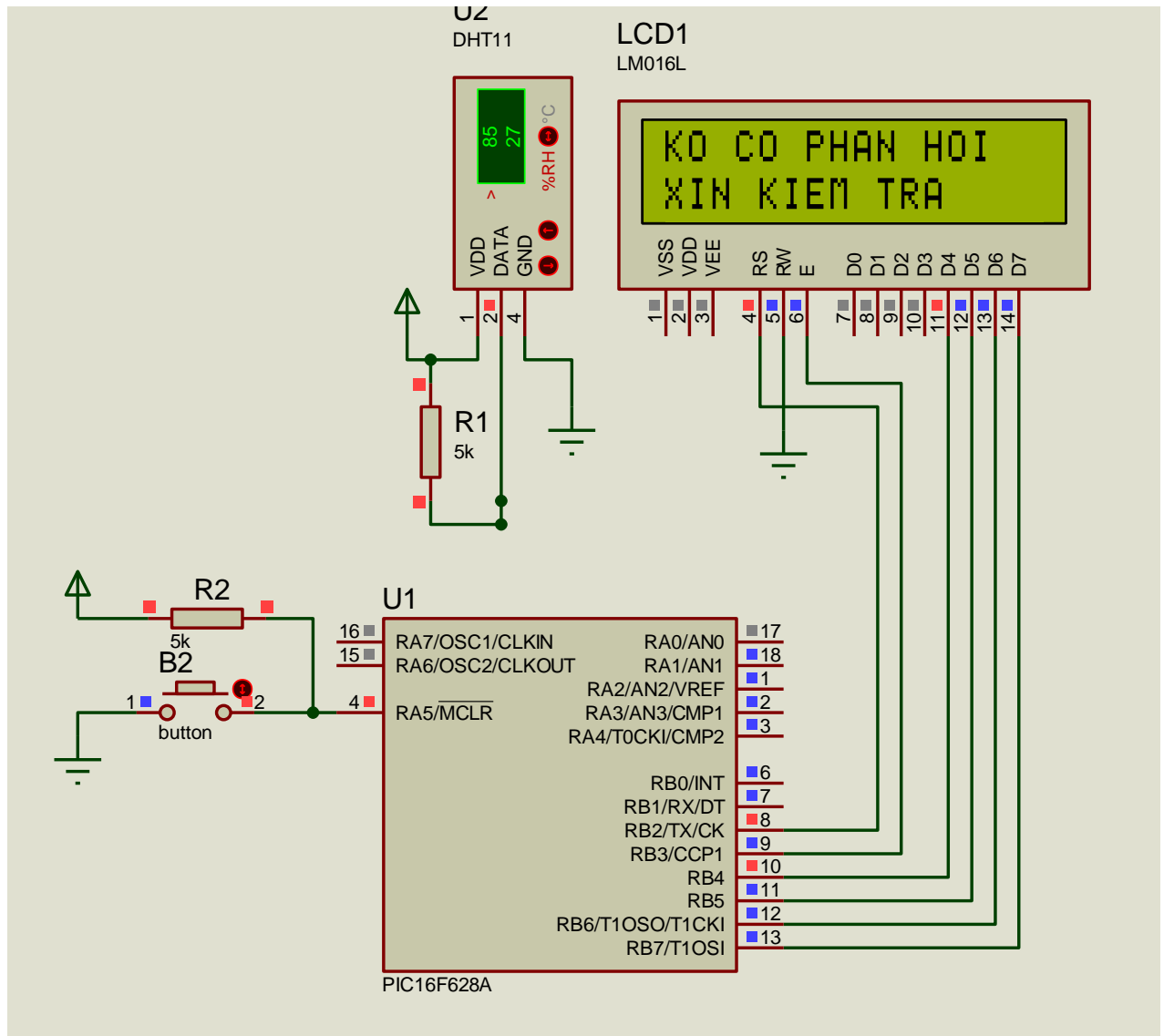


Hình 3.1 Mạch mô phỏng khi hoạt động.

Khi mạch hoạt động bình thường sẽ xuất kết quả ra màn hình.

3.1.2. khi mạch xảy ra sự cố

Có thể lỗi là do hỏng thiết bị cảm biến không dẫn hay các lỗi khác liên quan đến kết nối với cảm biến thì mạch sẽ thông báo lên màn hình.



Hình 3.2 Khi bị lỗi

Và cho chúng ta biết phải kiểm tra thiết bị, vì thiết bị gặp sự cố, cần được kiểm tra và bảo dưỡng.

CHƯƠNG 4 : THI CÔNG LẮP ĐẶT

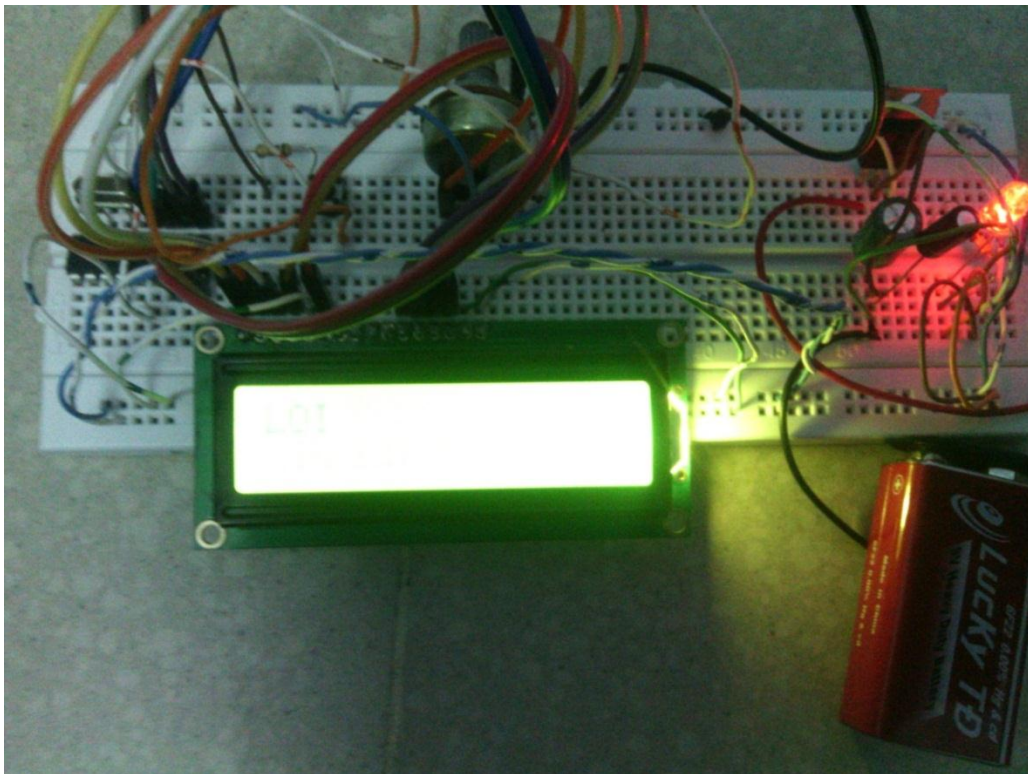
4.1. Trước khi thi công và lắp đặt

4.1.1. Tìm hiểu và chuẩn bị linh kiện

Tìm hiểu các linh kiện và chuẩn bị những linh kiện cần thiết để tiến hành bước kiểm tra trên testboard và khi thi công lắp đặt. Đảm bảo linh kiện đầy đủ và có chất lượng tốt để không xảy ra lỗi đáng có, gây chậm quá trình cũng như quá trình kiểm tra linh kiện còn tốt trước khi thực hiện.

Một số linh kiện cần thiết cho việc lắp đặt như :

- | | |
|------------------|---------------------|
| - VDK PIC16f628A | - Nút nhấn |
| - LCD LM16x2 | - Tụ gốm 22pF |
| - Led | - Tụ thạch anh 4Mhz |
| - Biến Trở 10k | - Tụ 470uF |
| - Cảm biến DHT11 | - Bảng đồng |
| - Biến áp 7805 | - Nguồn pin 9v |
| - Điện trở | - Bột sắt |



Hình 4.1 Thử trên test board

Mạch sau khi mô phỏng hoạt động trên testboard, và kiểm tra đã hoạt động bình thường. Và chuẩn bị cho quá trình thi công lắp đặt.

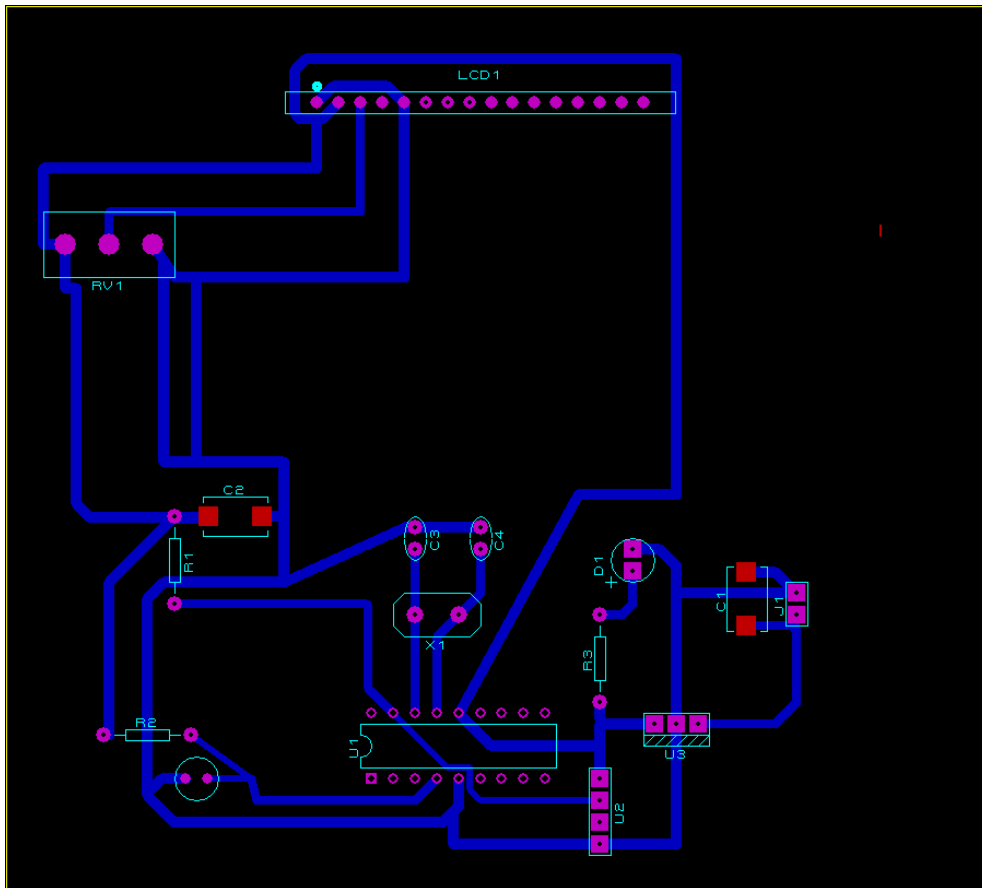
4.2.Thi công lắp đặt

Một số công đoạn lắp mạch như:

- Làm mạch in.
- Kiểm tra so sánh
- Lắp đặt linh kiện và thi công mạch

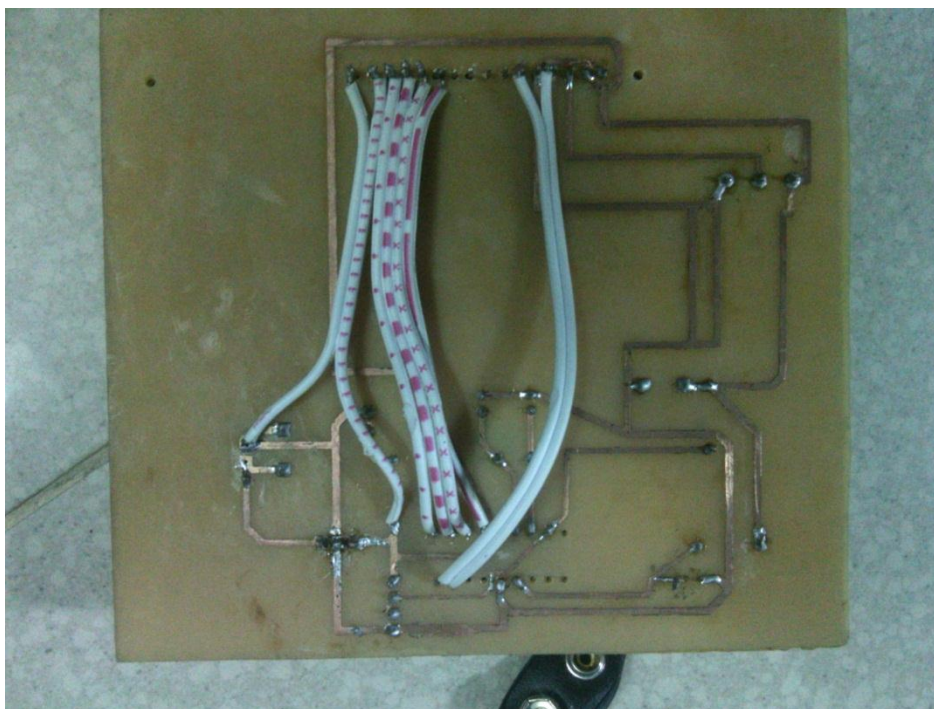
+ **Lưu ý :**

- Mạch in được thiết kế và vẽ trên các phần mềm protues hoặc orcad. Và dùng để làm mạch và lắp đặt linh kiện.

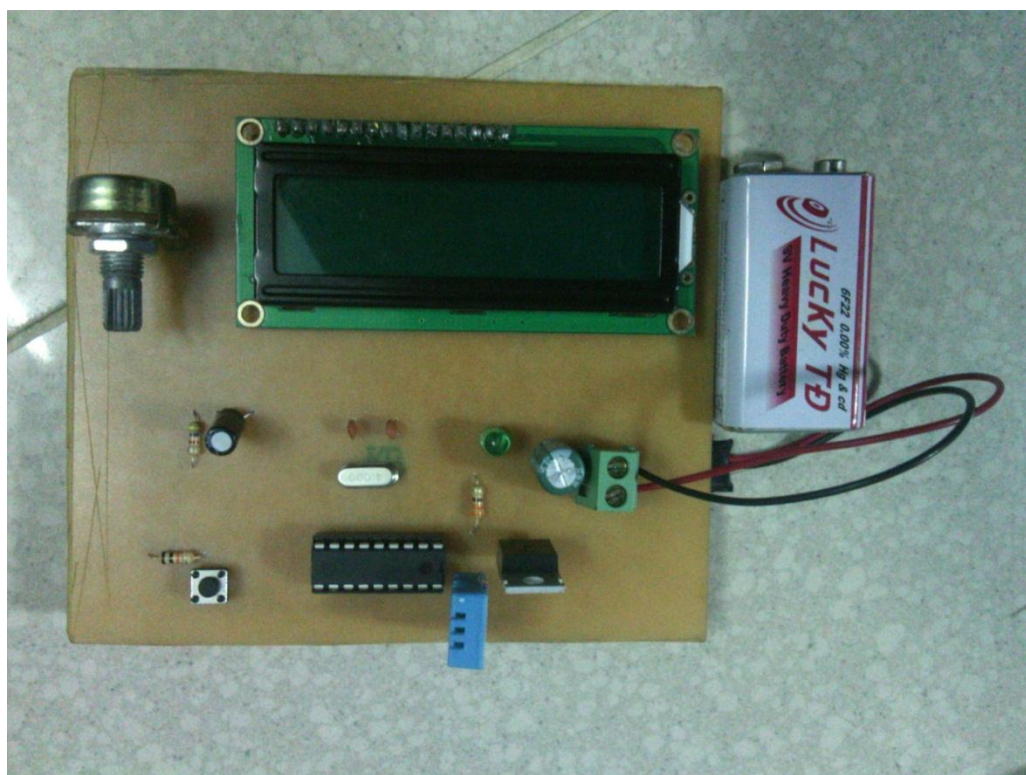


Hình 4.2 Mạch in.

- Kiểm tra và so sánh mạch khi làm có bị sai lỗi do quá trình làm, và kiểm tra hệ thống
- Khi đã kiểm tra, bắt đầu nối mạch, hàn mạch.



Hình 4.3 Mạch sau đã thi công.



Hình 4.4 Mặt trước đã thi công.

CHƯƠNG 5 : LẬP TRÌNH, CHẠY THỬ VÀ ĐÁNH GIÁ.

5.1. Giới thiệu

Lập trình cho vi điều khiển là một điều hết sức quan trọng để mạch có thể vận hành và đạt được mục đích khi thiết kế. Vì vậy sau khi thi công điều quan trọng đó là lập trình cho mạch, đặc biệt là lập trình đúng theo hoạt động cũng như theo sơ đồ khi lập trình.

Vì vi điều khiển sử dụng là PIC628a nên hiện tại có rất nhiều phần mềm lập trình phù hợp với vi điều khiển như là : CCS, ASM, flowcode 4, Mikro for Pic là những phần mềm thông dụng lập trình cho pic.

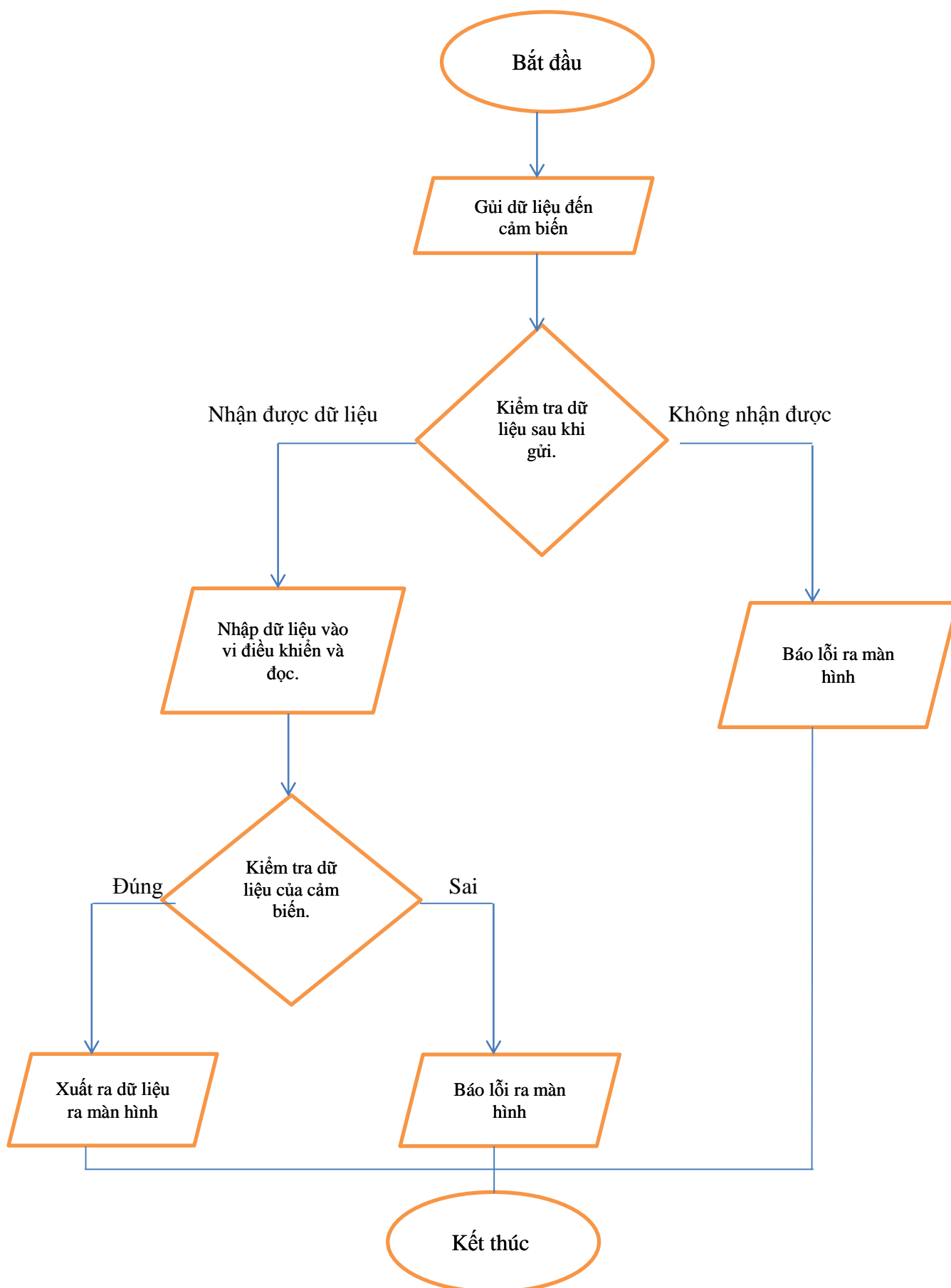
Ở đồ án này, Để lập trình cho dòng vi điều khiển Pic này thì em chọn phần mềm chuyên dành riêng lập trình cho vi điều khiển Pic đó là phần mềm MikroC for Pic phiên bản 6.4.0 .

Tại sao lại chọn phần mềm này? Bởi vì phần mềm khá nhiều ưu điểm :

- Dễ sử dụng bởi có bản hỗ trợ chi tiết cho người sử dụng giúp có thể dễ dàng hiểu và sử dụng.
- Thư viện khá phong phú và đầy đủ.
- Các câu lệnh khá đơn giản, thư viện có sẵn giúp người sử dụng dễ dàng tiếp cận cũng như rút ngắn thời gian thiết kế.

5.2. Phần lập trình

5.2.1 Sơ đồ giải thuật



Sơ Đồ khối phần giải thuật

5.2.2 .Mã chương trình

Chương trình được viết dưới dạng ngôn ngữ MikroC và được mã hoá thành chương trình có đuôi .hex để nạp vào vi điều khiển.

```
sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB3_bit;
sbit LCD_D4 at RB7_bit;
sbit LCD_D5 at RB6_bit;
sbit LCD_D6 at RB5_bit;
sbit LCD_D7 at RB4_bit;

sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB7_bit;
sbit LCD_D5_Direction at TRISB6_bit;
sbit LCD_D6_Direction at TRISB5_bit;
sbit LCD_D7_Direction at TRISB4_bit;

sbit Data at RA0_bit;
sbit DataDir at TRISA0_bit;

char message1[] = "ND  = 00.0 C";
char message2[] = "DA  = 00.0 %";

unsigned short TOUT = 0, CheckSum, i;

unsigned short T_Byte1, T_Byte2, DA_Byte1, DA_Byte2;
```

```
void StartSignal(){
    DataDir = 0;
    Data = 0;
    Delay_ms(25);
    Data = 1;
    Delay_us(30);
    DataDir = 1;
}

unsigned short CheckResponse(){
    TOUT = 0;
    TMR2 = 0;
    T2CON.TMR2ON = 1;
    while(!Data && !TOUT);
    if (TOUT) return 0;
    else {
        TMR2 = 0;
        while(Data && !TOUT);
        if (TOUT) return 0;
        else {
            T2CON.TMR2ON = 0;
            return 1;
        }
    }
}
```

```
unsigned short ReadByte(){
    unsigned short num = 0, t;
    DataDir = 1;
    for (i=0; i<8; i++){
        while(!Data);
        TMR2 = 0;
        T2CON.TMR2ON = 1;
        while(Data);
        T2CON.TMR2ON = 0;
        if(TMR2 > 40) num |= 1<<(7-i);
    }
    return num;
}
```

```
void interrupt(){
    if(PIR1.TMR2IF){
        TOUT = 1;
        T2CON.TMR2ON = 0;
        PIR1.TMR2IF = 0;
    }
}
```

```
void main() {
    unsigned short check;
```



```
TRISB = 0b00000000;
PORTB = 0;
TRISA = 0b00100001;
CMCON = 7;
INTCON.GIE = 1;
INTCON.PEIE = 1;

PIE1.TMR2IE = 1;
T2CON = 0;
PIR1.TMR2IF = 0;
TMR2 = 0;
Lcd_Init();
Lcd_Cmd(_Lcd_Clear);
Lcd_Cmd(_LCD_CURSOR_OFF);

do {
    Delay_ms(1000);
    StartSignal();
    check = CheckResponse();
    if (!check) {
        Lcd_Cmd(_Lcd_Clear);
        Lcd_Out(1, 1, "LOI");
        Lcd_Out(2, 1, "XIN KIEM TRA");
    }
    else{
```

```
DA_Byte1 = ReadByte();
```

```
DA_Byte2 = ReadByte();
```

```
T_Byte1 = ReadByte();
```

```
T_Byte2 = ReadByte();
```

```
Checksum = ReadByte();
```

```
if (Checksum == ((DA_Byte1 + DA_Byte2 + T_Byte1 + T_Byte2) & 0xFF))
```

```
{
```

```
    message1[7] = T_Byte1/10 + 48;
```

```
    message1[8] = T_Byte1%10 + 48;
```

```
    message1[10] = T_Byte2/10 + 48;
```

```
    message2[7] = DA_Byte1/10 + 48;
```

```
    message2[8] = DA_Byte1%10 + 48;
```

```
    message2[10] = DA_Byte2/10 + 48;
```

```
    message1[11] = 223;
```

```
    Lcd_Cmd(_Lcd_Clear);
```

```
    Lcd_Out(1, 1, message1);
```

```
    Lcd_Out(2, 1, message2);
```

```
}
```

```
else{
```

```
    Lcd_Cmd(_Lcd_Clear);
```

```
    Lcd_Out(1, 1, "KIEM TRA LAI!");
```

```
Lcd_Out(2, 1, "VA THU LAI...");  
  
}  
  
}  
}while(1);  
  
}
```

5.3. Đánh giá và kết luận

Các kết quả đạt được

+ Trên cơ sở thiết kế phần cứng và phần mềm khi thực hiện đồ án đã đạt được một số kết quả sau:

- Về phần cứng: Tìm hiểu về vi điều khiển Pic các linh kiện cũng như cách thiết kế và làm mạch thủ công.
- Đi sâu tìm hiểu một số thuật toán điều khiển như bộ điều khiển PIC.
- Nhận dạng và mô phỏng đối tượng điều khiển .

+ Một số điểm hạn chế

- Trong gian đoạn làm đồ án, chúng em đã rất cố gắng nghiên cứu, thiết kế để có được những kết quả đã nêu trên. Tuy nhiên do thời gian và kiến thức của chúng em có hạn, mặt khác một số điều kiện về thiết bị không cho phép nên trong đồ án còn những hạn chế và thiếu sót.
- Pic 16F628A là dòng vi điều khiển 4bit nên có tốc độ cũng như bộ nhớ thấp, cũng như ít chân hơn nên hạn chế trong lập trình và điều khiển.

Về đề tài :

- Đề tài có ứng dụng khá nhiều trong cuộc sống và thực tế.
- Đề tài có hướng phát triển sẽ có thể điều khiển được nhiệt độ trong phòng bằng cách tự động tắt mở quạt, máy lạnh mở cửa sổ, đặc biệt hơn là báo cháy và tự động dập lửa.
- Do lí do về thời gian những lí do khách quan khác nên đoán được thực hiện vẫn còn nhiều khiếm khuyết. Chúng em rất mong nhận được những ý kiến đóng góp quý báu của các thầy cô.
- Một lần nữa, chúng em xin chân thành cảm ơn Thầy đã hướng dẫn chúng em tận tình trong quá trình thực hiện đồ án này.

-----o0o-----