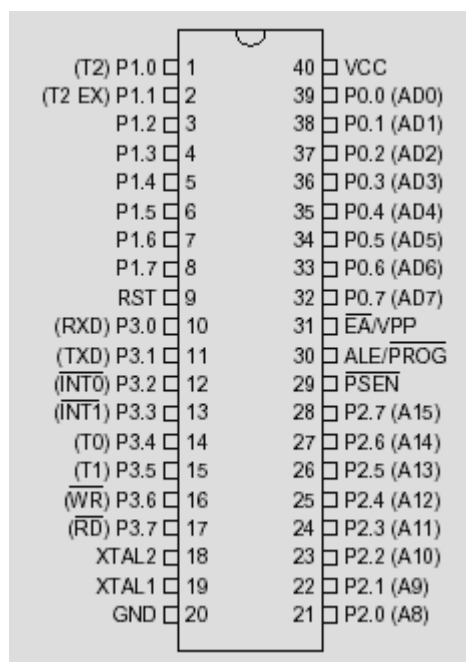


Phần 1 : Giới thiệu chung về vi điều khiển 8051



Vi điều khiển 8051

Khối điều khiển ngắt với 2 nguồn ngắt ngoài và 4 nguồn ngắt trong

Bộ lập trình (ghi chương trình lên Flash ROM) cho phép người sử dụng có thể nạp các chương trình cho chip mà không cần đến bộ nạp chuyên dụng

Bộ chia tần số với hệ số chia là 12

4 cổng xuất nhập với 32 chân

1. Port 0 (P0.0 – P0.7) : Port 0 gồm 8 chân, ngoài chức năng xuất nhập, port 0 còn là bú dũ liệu và địa chỉ (AD0 – AD7), chức năng này sẽ được sử dụng khi 8051 giao tiếp với các thiết bị ngoài có kiến trúc bú ngữ các vi mạch nhớ ...
2. Port 1 (P1.0 – P1.7) : Có chức năng xuất nhập theo bit và theo byte. Bên cạnh đó 3 chân P1.5, P1.6, P1.7 được dùng để nạp ROM theo chuẩn IP, 2 chân P1.0 và P1.1 được dùng cho bộ Timer.

3. Port2: là cổng vào/ra còn là byte cao của bus địa chỉ khi sử dụng bộ nhớ ngoài.
4. Port 3: ngoài chức năng xuất nhập còn có chức năng riêng

Bit	Tên	Chức năng
P3.0	RXD	Dữ liệu nhận cho port nối tiếp
P3.1	TXD	Dữ liệu truyền cho port nối tiếp
P3.2	INT0	Ngắt bên ngoài 0
P3.3	INT1	Ngắt bên ngoài 1
P3.4	T0	Ngõ vào của Timer/counter 0
P3.5	T1	Ngõ vào của Timer/counter 1
P3.6	/WR	Xung ghi nhớ dữ liệu ngoài
P3.7	/RD	Xung đọc bộ nhớ dữ liệu ngoài

5. Chân / PSEN (Program Store Enable) : là chân điều khiển đọc chương trình ở bộ nhớ ngoài nó được phép đọc các byte mã lệnh trên ROM ngoài/ PSEN sẽ ở mức thấp trong thời gian đọc lệnh. Mã lệnh được đọc từ bộ nhớ ngoài qua bus dữ liệu (port 0) thanh ghi lệnh để được giải mã. Khi thực hiện chương trình ROM nội thì /PSEN ở mức cao.
6. Chân ALE (Address Latch Enable) : ALE là tín hiệu điều khiển chốt địa chỉ có tần số bằng 1/6 tần số dao động của vi điều khiển. Tín hiệu ALE được dùng để cho phép vi mạch chốt bên ngoài như 74373,74573 chốt byte địa chỉ thấp ra khỏi bus đa hợp địa chỉ/dữ liệu (port 0).
7. Chân /EA(External Access) : tín hiệu cho phép chọn bộ nhớ chương trình là bộ nhớ ngoài hay trong vi điều khiển. Nếu /EA ở mức cao (nối với V_{cc}) thì vi điều khiển thi hành chương trình trong ROM nội. Nếu /EA ở mức thấp (nối GND) thì vi điều khiển thi hành chương trình bộ nhớ ngoài.
8. XTAL1,XTAL2 : AT89S52 có một bộ dao động trên chip, nó thường nối với bộ dao động thạch anh có tần số lớn nhất là 33MHz, thông thường là 12MHz
9. V_{cc} ,GND : AT89S52 dùng nguồn chiều có độ dài điện áp từ 4V đến 5,5V được cấp qua chân 40 và 20.

Thanh ghi TMOD chứa 2 nhóm 4 bit dùng để đặt chế độ làm việc cho Timer 0 và Timer 1.

Thanh ghi TMOD

GATE1	C/#T1	M1	M0	GATE0	C/#T0	M1	M0
Bit	Ký hiệu	Chức năng					
7	GATE1	Bit điều khiển cổng. Khi set lên 1, bộ định thời chỉ hoạt động trong khi INT1 ở mức cao					
6	C/#T1	Bit chọn chức năng đếm hoặc định thời 1= đếm sự kiện 0= định thời trong 1 khoảng thời gian					
5	M1	Bit chọn chế độ thứ nhất					
4	M0	Bit chọn chế độ thứ 2 00 chế độ 0 – Timer 13 bit 01 chế độ 1 – Timer 16 bit 10 chế độ 2 – 8 bit tự động nạp lại 11 chế độ 3 – tách Timer					
3	GATE0	Bit điều khiển cổng cho bộ định thời 0					
2	C/#T0	Bit chọn chức năng đếm hoặc định thời cho bộ định thời 0					
1	M1	Bit chọn chế độ thứ nhất cho bộ định thời 0					
0	M0	Bit chọn chế độ thứ 2 cho bộ định thời 0					

Thanh ghi điều khiển Timer (TCON)

Thanh ghi TCON chứa các bit trạng thái và các bit điều khiển cho Timer 0 và Timer 1.

TF1	TR1	TF0	TR0	IT1	IE1	IT0	IE0
-----	-----	-----	-----	-----	-----	-----	-----

Bit	Ký hiệu	Chức năng
TCON		Điều khiển bộ định thời
TCON.7	TF1	Cờ tràn của bộ định thời 1. Cờ này được set bởi phần cứng khi tràn, được xóa bởi phần mềm, hoặc bởi phần cứng khi bộ xử lý trở đến trình phục vụ ngắt
TCON.6	TR1	Bit điều khiển hoạt động của bộ định thời 1. Bit này

		được set hay xóa bằng phần mềm để điều khiển bộ định thời hoạt động hay ngưng
TCON.5	TF0	Cờ tràn của bộ định thời 0
TCON.4	TR0	Bit điều khiển hoạt động của bộ định thời
TCON.3	IE1	Cờ ngắt bên ngoài 1 (kích khởi cạnh). Cờ này được set bởi phần cứng khi có cạnh âm (cuống) xuất hiện trên chân INT1, được xóa bởi phần mềm, hoặc phần cứng khi CPU trở đến trình phục vụ ngắt
TCON.2	IT1	Cờ ngắt bên ngoài 1 (kích khởi cạnh hoặc mức). Cờ này được set hay xóa bởi phần mềm khi xảy ra cạnh âm hoặc mức thấp ở chân ngắt ngoài
TCON.1	IE0	Cờ ngắt bên ngoài 0 (kích khởi cạnh)
TCON.0	IT0	Cờ ngắt bên ngoài 0 (kích khởi cạnh hoặc mức)

CÁC CHẾ ĐỘ ĐỊNH THỜI CỦA TIMER 0 VÀ 1

1. Chế độ 0 : là chế độ định thời 13 bit, chế độ này tương thích với các bộ vi điều khiển trước đó, trong các ứng dụng hiện nay chế độ này không còn thích hợp.
Trong chế độ này bộ định thời dùng 13 bit (8 bit của TH và 5 bit cao của TL) để chứa các giá trị đếm, 3 bit thấp của TL không được sử dụng.
2. Chế độ 1 : Trong chế độ này, bộ timer dùng cả 2 thanh ghi TH và TL để chứa giá trị đếm, vì vậy chế độ này còn được gọi là chế độ định thời 16 bit. Bit MSB sẽ là bit D7 của TH còn bit LSB sẽ là D0 của TL.
3. Chế độ 2 : Trong chế độ 2, bộ định thời dùng TL để chứa giá trị đếm và TH để chứa giá trị nạp vì vậy chế độ này còn gọi là chế độ tự nạp lại 8 bit. Sau khi đếm 255 sẽ xảy ra tràn, khi đó TF được đặt bằng 1 đồng thời giá trị của timer tự động được nạp lại bằng nội dung của TH.
4. Chế độ 3 : Trong chế độ 3, Timer 0 được tách thành 2 bộ Timer hoạt động độc lập chế độ này sẽ cung cấp cho bộ vi điều khiển thêm 1 Timer nữa.

TỔ CHỨC NGẮT Ở AT89S52

Bảng tóm tắt các ngắt trong AT89S52 như sau:

STT	Tên ngắt	Mô tả	Cờ ngắt	Thanh ghi	Vector ngắt
-----	----------	-------	---------	-----------	-------------

				chứa cờ	
1	INT0	Ngắt ngoài 0 khi có tín hiệu tích cực theo kiểu đã chọn ở chân P3.2	IE0	TCON	0x0003
2	Timer 0	Ngắt tràn timer 0 khi giá trị timer 0 tràn từ giá trị max về giá trị min	TF0	TCON	0x000B
3	INT1	Ngắt ngoài 1 khi có tín hiệu tích cực theo kiểu đã chọn ở chân P3.3	IE1	TCON	0x00013
4	Timer 1	Ngắt tràn timer 1 khi giá trị timer1 tràn từ giá trị max về giá trị min	TF1	TCON	0x001B
5	Serial Port	Ngắt cổng nối tiếp khi vi điều khiển nhận hoặc truyền xong 1 byte bằng cổng nối tiếp	TI, RI	SCON	0x0023
6	Timer2	Ngắt tràn timer 2 khi giá trị timer 2 tràn	TX2 hoặc EXT2	T2CON	002BH

Thanh ghi IE

EA	-	ET2	É	ET1	EX1	ET0	EX0
----	---	-----	---	-----	-----	-----	-----

Bit	Ký hiệu	Địa chỉ bit	Mô tả
IE.7	EA	AFH	Cho phép/ cấm toàn bộ
IE.6	-	AEH	Không được miêu tả
IE.5	ET2	ADH	Cho phép ngắt từ timer 2 (8052)
IE.4	ES	ACH	Cho phép ngắt từ port nối tiếp
IE.3	ET1	ABH	Cho phép ngắt từ Timer1
IE.2	EX1	AAH	Cho phép ngắt ngoài 1
IE.1	ET0	A9H	Cho phép ngắt từ Timer 0
IE.0	EX0	A8H	Cho phép ngắt ngoài 0

Ngắt do các timer

AT89S52 có 3 Timer là Timer 0 và Timer 2. Các Timer này đều là Timer 16 bit, giá trị đếm mã do đó bằng 65535 (đếm từ 0 đến 65535). Ba timer có nguyên lý hoạt động hoàn toàn giống nhau và độc lập.

Các ngắt do các bộ Timer xảy ra do sự kiện tràn ở các Timer, khi đó các cờ tràn TF_x sẽ được đặt bằng 1. Khi ISR được đáp ứng, các cờ TF_x sẽ tự động được xóa bởi phần mềm.

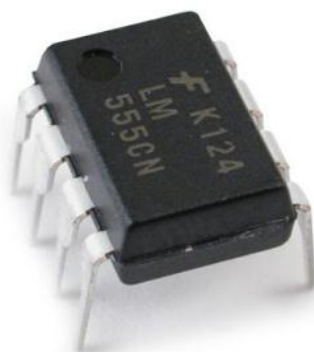
Ngắt do cổng nối tiếp

Ngắt do cổng nối tiếp xảy ra khi hoặc cờ phát ngắt (TI) hoặc cờ ngắt thu (RI) được đặt bằng 1, ngắt phát xảy ra khi bộ đệm truyền rỗng, ngắt thu xảy ra khi 1 ký tự đã được nhận xong và đang đợi trong SBUF để được đọc.

Các ngắt do cổng nối tiếp khác ngắt do timer, cờ gây ra ngắt do port nối tiếp không bị xóa bằng phần cứng khi CPU chuyển tới I do có 2 nguồn ngắt do cổng nối tiếp TI và RI, nguồn ngắt phải được xác định trong ISR và cờ tạo ngắt sẽ được xóa bằng phần mềm.

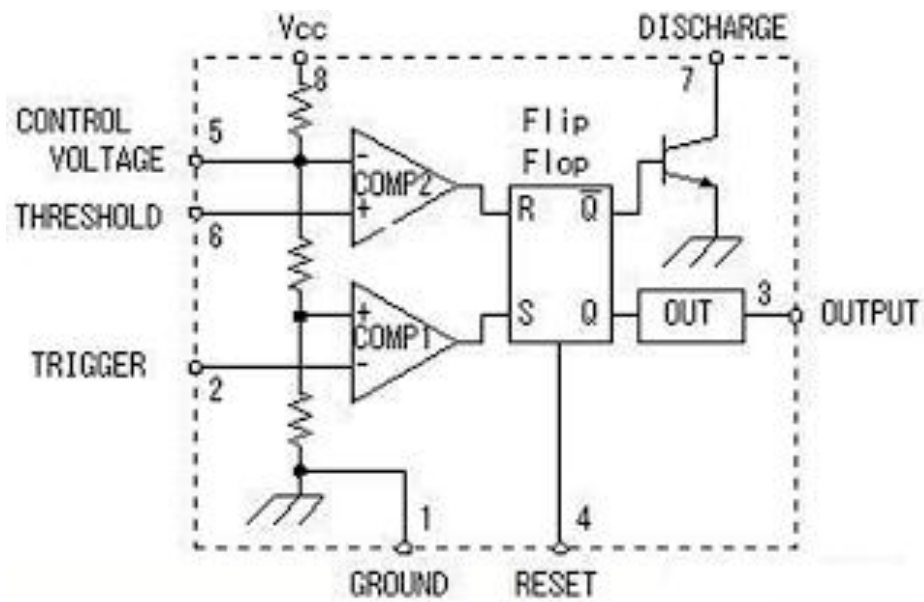
Phần 2 : Bộ tạo xung bằng IC NE 555

Khái quát về IC 555



1. Cấu tạo nguyên lý hoạt động của IC 55

Cấu tạo của NE555 gồm OP-amp so sánh điện áp, mạch lật và transistor để xả điện. cấu tạo của IC đơn giản nhưng hoạt động tốt. Bên trong gồm 3 điện trở mắc nối tiếp chia điện áp V_{cc} thành 3 phần. Cấu tạo này tạo nên điện áp chuẩn. Điện áp $1/3 V_{cc}$ nối vào chân dương của Op-amp 1 và điện áp $2/3 V_{cc}$ nối vào chân âm của Op-amp 2. Khi điện áp ở chân 2 nhỏ hơn $1/3 V_{cc}$, chân S= [1] và FF được kích. Khi điện áp ở chân 6 lớn hơn $2/3 V_{cc}$, chân R của FF= [1] và FF được reset.



Chân 1 : GND (nối đất) Chân 2 : Trigger Input

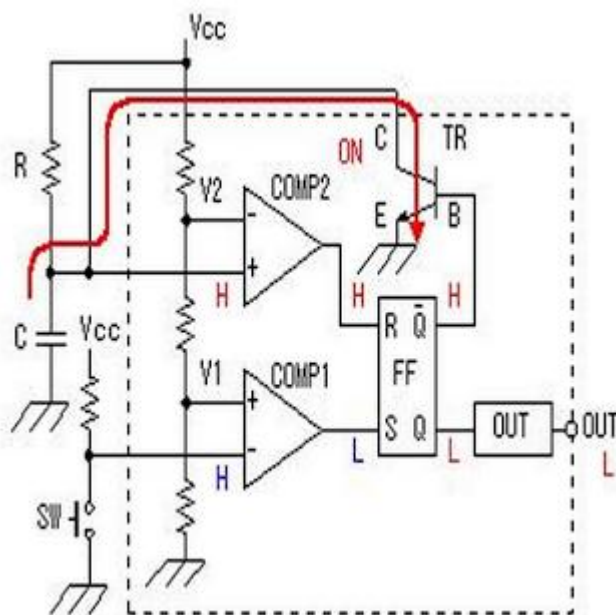
Chân 3 : Out put (ngõ ra) Chân 4 : Reset (hồi phục)

Chân 5 : Control Voltage (điện áp điều khiển)

Chân 6 : Threshold (thêm ngưỡng)

Chân 7 : Discharge (phóng điện) Chân 8 : +Vcc (nguồn dương)

- Giải thích sự dao động :



Ký hiệu mức 0 là mức thấp bằng 0V, 1 là mức cao gần bằng Vcc. Mạch FF là loại RS Flip-flop.

Khi $S = [1]$ thì $Q = [1]$ và $\bar{Q} = [0]$

Sau đó, khi $S = [0]$ thì $Q = [1]$ và $\bar{Q} = [0]$.

Khi $R = [1]$ thì $\bar{Q} = [1]$ và khi $R = [1]$ thì $Q = [0]$ bởi vì $\bar{Q} = [1]$, transistor mở dẫn, cực C nối đất. Cho nên điện áp không nạp vào tụ C, điện áp ở chân 6 không vượt quá V2. Do lỗi ra của Op-amp 2 ở mức 0, FF không reset.

Giai đoạn ngõ ra ở mức 1 :

Khi bấm công tắc khởi động, chân 2 ở mức 0.

Vì điện áp ở chân 2 (V-) nhỏ hơn V1(V+), ngõ ra của Op-amp 1 ở mức 1 nên $S = [1]$. $Q = [1]$ và $\bar{Q} = [0]$. Ngõ ra của IC ở mức 1.

Khi $\bar{Q} = [0]$, transistor tắt, tụ C tiếp tục nạp vào R, điện áp trên tụ tăng. Khi nhấn công tắc lần nữa Op-amp 1 có $V- = [1]$ lớn hơn V+ nên

ngõ ra của Op-amp 1 ở mức 0, $S = [0]$. Q và \bar{Q} vẫn không đổi. Trong khi tụ điện áp C nhỏ hơn V_2 , FF vẫn giữ nguyên trạng thái đó.

Giai đoạn ngõ ra ở mức 0:

Khi tụ C nạp tiếp, Op-amp 2 có V_+ lớn hơn $V_- = 2/3 V_{cc}$, $R = [1]$ nên $Q = [0]$ và $\bar{Q} = [1]$. Ngõ ra của IC ở mức 0. Vì $\bar{Q} = [1]$, transistor mở dẫn, Op-amp2 có $V_+ = [0]$ bé hơn V_- , ngõ ra của Op-amp 2 ở mức 0. Vì vậy Q và \bar{Q} không đổi giá trị, tụ C xả điện thông qua transistor.

Kết quả cuối cùng : Ngõ ra OUT có tín hiệu dao động dạng sóng vuông, có chu kì ổn định

1. Trong mạch này, **giá trị tần số tạo ra 3 dải tần số** bằng cách thay đổi giá trị tụ điện tương ứng
2. $R_1 = 10k\Omega$; R_2 là biến trở $100k\Omega + R1k\Omega$

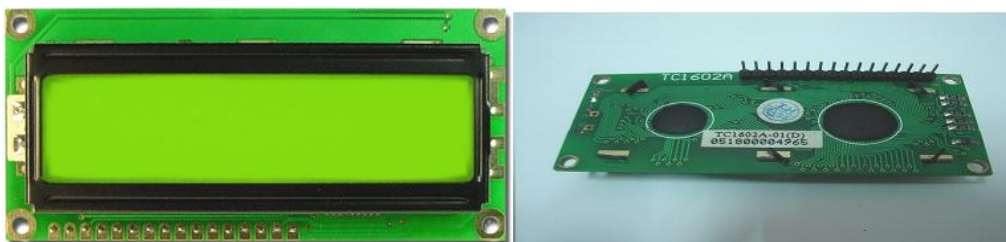
Dải 1 từ 0Hz đến 100Hz; tụ giá trị $100\mu F$

Dải 2 từ 1000Hz đến 2000Hz ; tụ giá trị $47nF$

Dải 3 từ 1 KHz; Tụ giá trị $4,7nF$

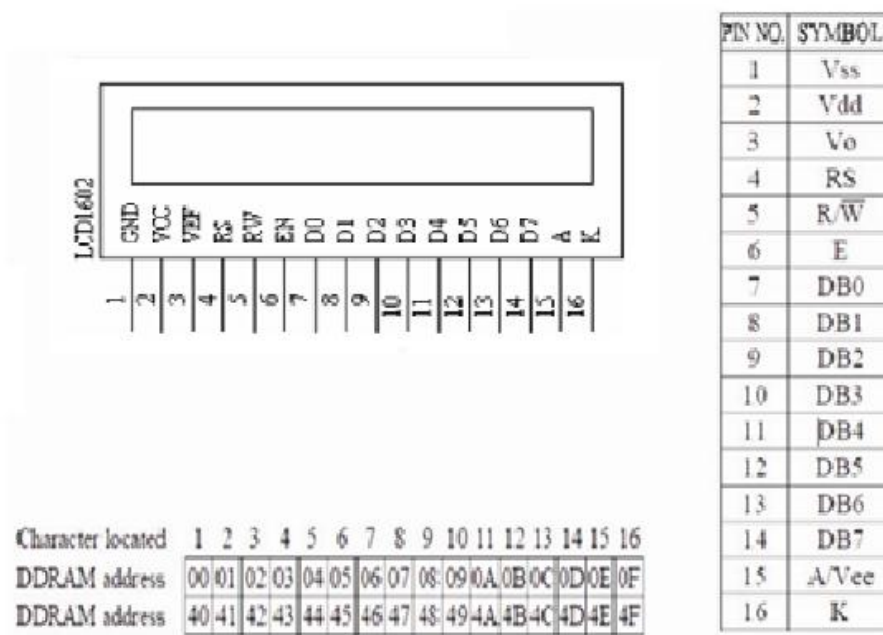
Phần 3 : Hiển thị LCD

-Hình dáng và kết cấu chân của LCD



Hình dáng LCD thực tế

- Kết cấu chân của LCD



-Cách ghép nối LCD với vi điều khiển 8051 như sau:

+Chân P2.0 đến P2.7 được nối với chân dữ liệu D0 –D7 của LCD

+Chân P2.0 được nối tới chân RS của LCD

+Chân P2.1 được nối với chân R/W của LCD

+Chân P2.2 được nối với chân E của LCD

+ Chân P2.4 được nối với chân D4 của LCD

+ Chân P2.5 được nối với chân D5 của LCD

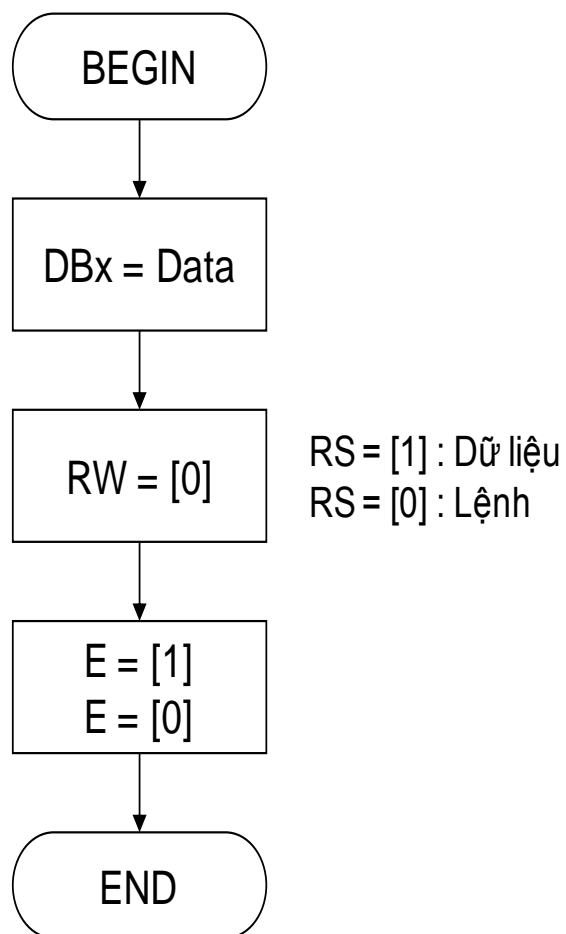
+ Chân P2.6 được nối với chân D6 của LCD

+ Chân P2.7 được nối với chân D7 của LCD

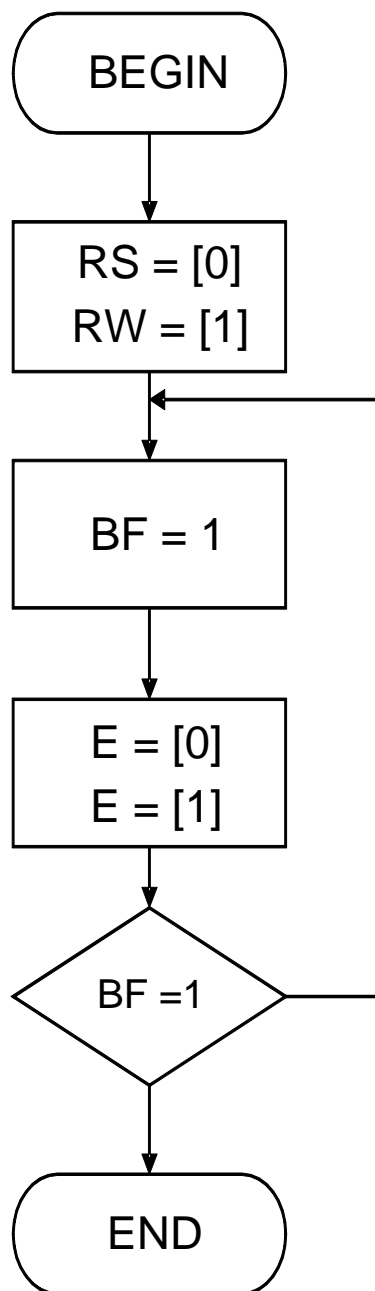
-Phương pháp gửi mã lệnh hoặc dữ liệu tới LCD

❖ LCD và 8051 hoạt động không đồng bộ với nhau, 8051 xử lý nhanh hơn LCD, do đó sau khi ra một lệnh cho LCD, 8051 phải đợi LCD thực hiện xong lệnh trước đó mới được ra lệnh tiếp theo. Để chờ LCD thực hiện xong lệnh trước đó ta có 2 phương pháp được sử dụng để gửi mã lệnh hoặc dữ liệu đến LCD:

❖ - Phương pháp 1: Gửi các lệnh và dữ liệu đến LCD với một độ trễ, tức là sau khi ra một lệnh, 8051 phải đợi một khoảng thời gian cố định, thời gian này phải dài hơn thời gian làm việc của LCD (do nhà sản xuất quy định khoảng từ 37 us đến 1,52 ms). Phương pháp này được mô tả bằng bảng lưu đồ sau



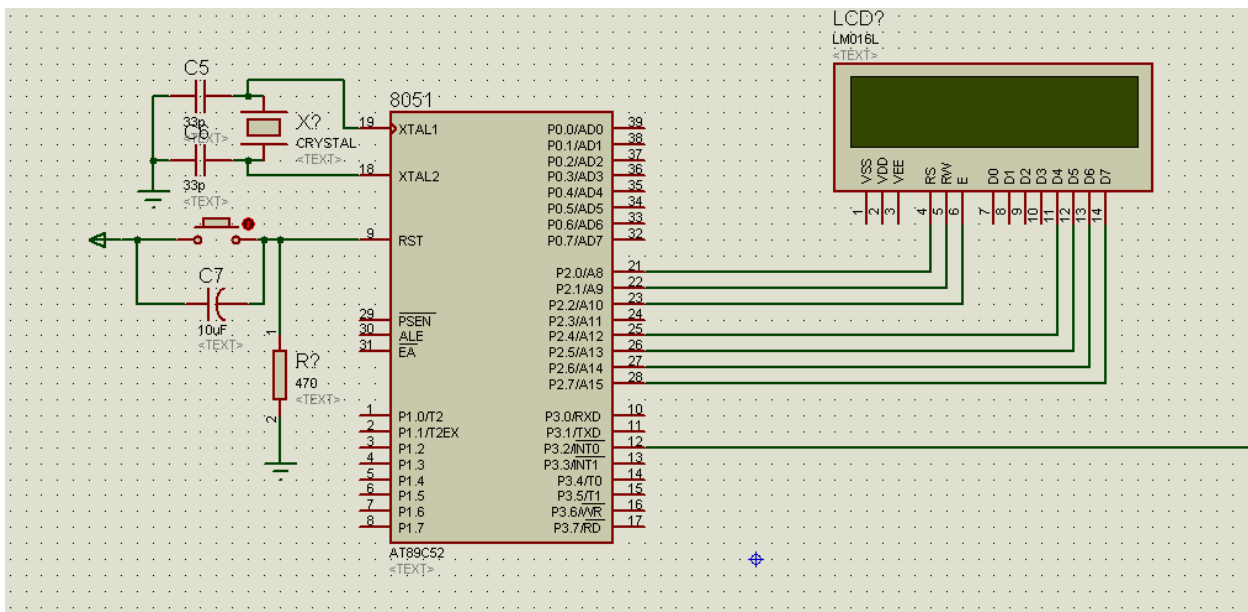
- ❖ -Phương pháp 2: - Phương pháp 2:Gửi các lệnh và dữ liệu đến LCD có kiểm tra cờ bận (ở phương pháp này ta không cần phải đặt 1 độ trễ lớn trong quá trình xuất dữ liệu hay ra lệnh cho LCD mà chỉ cần hiện thị cờ bận (đọc cờ bận BF thông qua chân DB7) từ LCD trước khi xuất một lệnh hoặc dữ liệu tới LCD). Phương pháp này được mô tả bằng lưu đồ sau



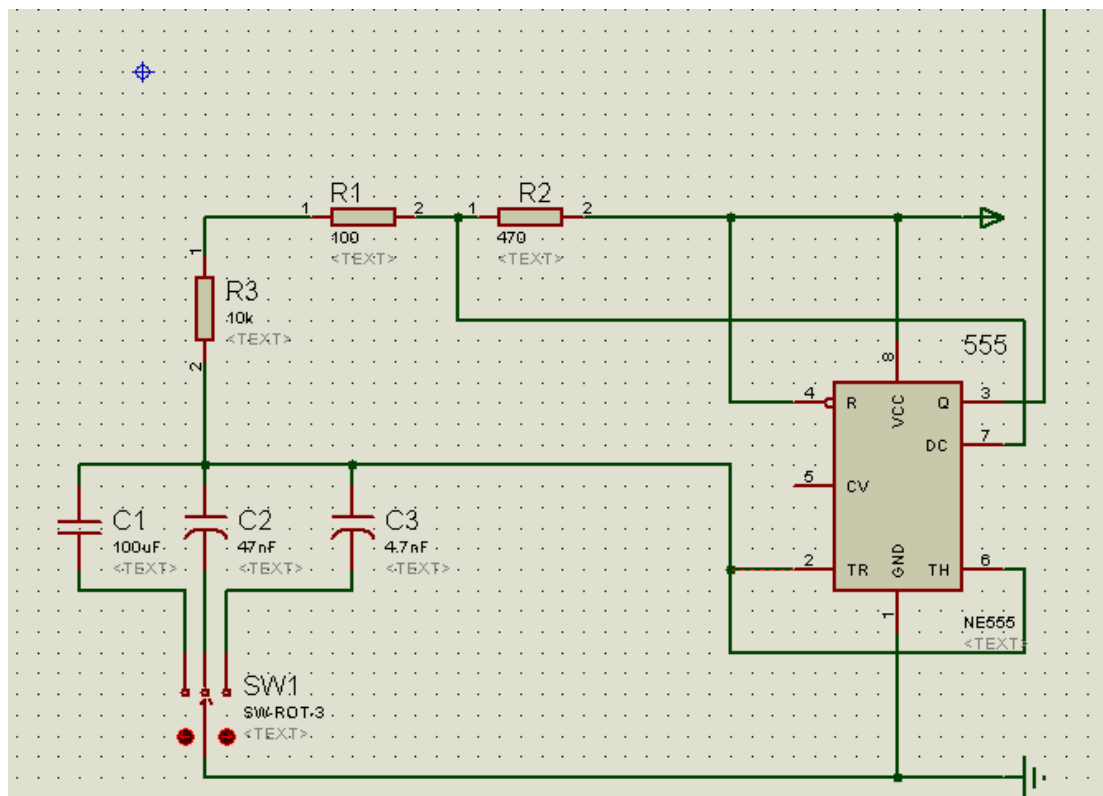
Lưu ý rằng ở bất kì phương pháp nào để gửi một lệnh bất kì tới LCD ta phải đưa chân RS về 0. Đối với dữ liệu thì bật RS = 1 sau đó gửi một sườn xung cao xuống thấp đến chân E để cho phép chốt dữ liệu trong LCD

1. Sơ đồ nguyên lý, mạch in mạch đo tần số.

- Mạch ghép nối giữa vi điều khiển và LCD



- Mạch tạo xung bằng IC NE555



** Xung từ chân số 3 của IC NE 555 được gửi tới chân số P3.2 (chân số 12) của vi điều khiển 89s52

2. Code chương trình

```
#include <at89x51.h>

#define RS  P2_0          // khai bao chan RS cua ICD
#define RW  P2_1          //khai bao chan RW cua LCD
#define E   P2_2

#define DATA P2 //dinh nghi PORTD lam chan du lieu cho LCD

void lcd_set(unsigned char lenh); //setting LCD

void lcd_command(unsigned char lenh1); //ghi lenh vao LCD

void lcd_write(unsigned char lenh2); //ghi du lieu vao LCD

void lcd_init();

void lcd_putnum(unsigned long info); //viet so vao LCD

void lcd_Gotoxy(unsigned char x,unsigned char y); //nhay den vi tri
x,y trong LCD

void lcd_clear();//ham xoa man hinh trong LCD

void lcd_putchar(unsigned char *s); //ham ghi 1 chuoai ky tu vao
LCD

unsigned int Millisec; //bien dem ms

unsigned int so_xung;

unsigned int frequency;
```

```
void delay_ms(unsigned int ms)           //ham delay
{
    unsigned int i,j;
    for(i=0;i<ms;i++)
        for(j=0;j<121;j++);
}

void delay_us(unsigned int us)
{
    unsigned int i;
    for(i=0;i<us;i++);
}

void ISR_1000ms(void) interrupt 1        // ham ngat timer 0
{
    TF0 = 0;//xoa co tran
    TH0 =0xE2;        // nap du lieu vao thanh ghi timer
    TL0 =0x0A;
    TR0=1;
    Millisec++;
    if(Millisec==0xFFFD)
        Millisec=0;
}
```

```
void EXT_INT0(void) interrupt 0
{
    so_xung++;
}
void main()
{
    IE=0x85;//cho phép ngắt ngoài INT0 và INT1
    TCON = 0x01; // chế độ 16 bit tự nạp lại
    TR0=1;    //cho phép timer hoạt động
    TH0=0xfc;
    TL0=0x18; //nạp giá trị cho timer
    TR0=1; // cho phép timer 0 hoạt động
    TF0=0;
    EA=1; //cho phép ngắt
    ET0=1; //cho phép ngắt timer 0
    IT0=1;
    lcd_init();
    while(1)
    {
        if(Millisec>=1000)
        {
            Millisec=0;
        }
    }
}
```

```
        frequency=so_xung;    //so xung trong 1 s dem duoc
tu ngat ngoai
        so_xung=0;
        lcd_Gotoxy(0,1);
        lcd_putchar("tan So: ");
        lcd_putnum(frequency);    //hien thi len LCD tan
so do duoc
        lcd_putchar(" Hz");
    }
}
}
void lcd_set(unsigned char lenh)
{
    RS=0;
    RW=0;
    DATA=lenh;
    E=1;
    delay_us(50);
    E=0;
}
void lcd_command(unsigned char lenh1)
{
```

```
// char temp;

RS=0;

RW=0;

DATA=(DATA&0x0F)|(lenh1&0xF0); //temp;

E=1;

delay_us(20);

E=0;

delay_us(20);

DATA=(DATA&0x0F)|((lenh1<<4)&0xF0);

E=1;

delay_us(20);

E=0;

delay_us(50);

}

void lcd_write(unsigned char lenh2)

{

RS=1;

RW=0;

DATA=(DATA&0x0F)|(lenh2&0xF0);

E=1;
```

```
    delay_us(10);  
    E=0;  
    delay_us(10);  
    RS=1;  
    RW=0;  
    delay_us(10);  
    DATA=(DATA&0x0F)|((lenh2<<4)&0xF0);  
    E=1;  
    delay_us(10);  
    E=0;  
    delay_us(50);  
}
```

```
void lcd_init()  
{  
    lcd_set(0x30);  
    delay_us(50);  
    lcd_set(0x30);  
    delay_us(50);  
    lcd_set(0x30);  
    delay_us(50);  
    lcd_set(0x20);
```

```
    delay_us(50);

    //delay_us(50);
    lcd_command(0x80);
    lcd_command(0x06);
    lcd_command(0x08);
    lcd_command(0x0C);
    lcd_command(0x28);
    lcd_command(0x01);
    delay_ms(20);
}

void lcd_putchar(unsigned char *s)
{
    char i;
    for(i=0;s[i]!=0;i++)
    {
        lcd_write(s[i]);
    }
}

void lcd_putnum(unsigned long info) //ham in 1 so len man hinh
LCD
```

```
{  
    unsigned int chucnghin,nghin,tram,chuc,donvi;  
    chucnghin=info/10000;  
    nghin=(info-chucnghin*10000)/1000;  
    tram=(info-chucnghin*10000-nghin*1000)/100;  
    chuc=(info-chucnghin*10000-nghin*1000-100*tram)/10;  
    donvi= (info-chucnghin*10000-nghin*1000-100*tram-chuc*10);  
  
    lcd_write(chucnghin+48);  
    lcd_write(nghin+48);  
    lcd_write(tram+48);  
    lcd_write(chuc+48);  
    lcd_write(donvi+48);  
}
```

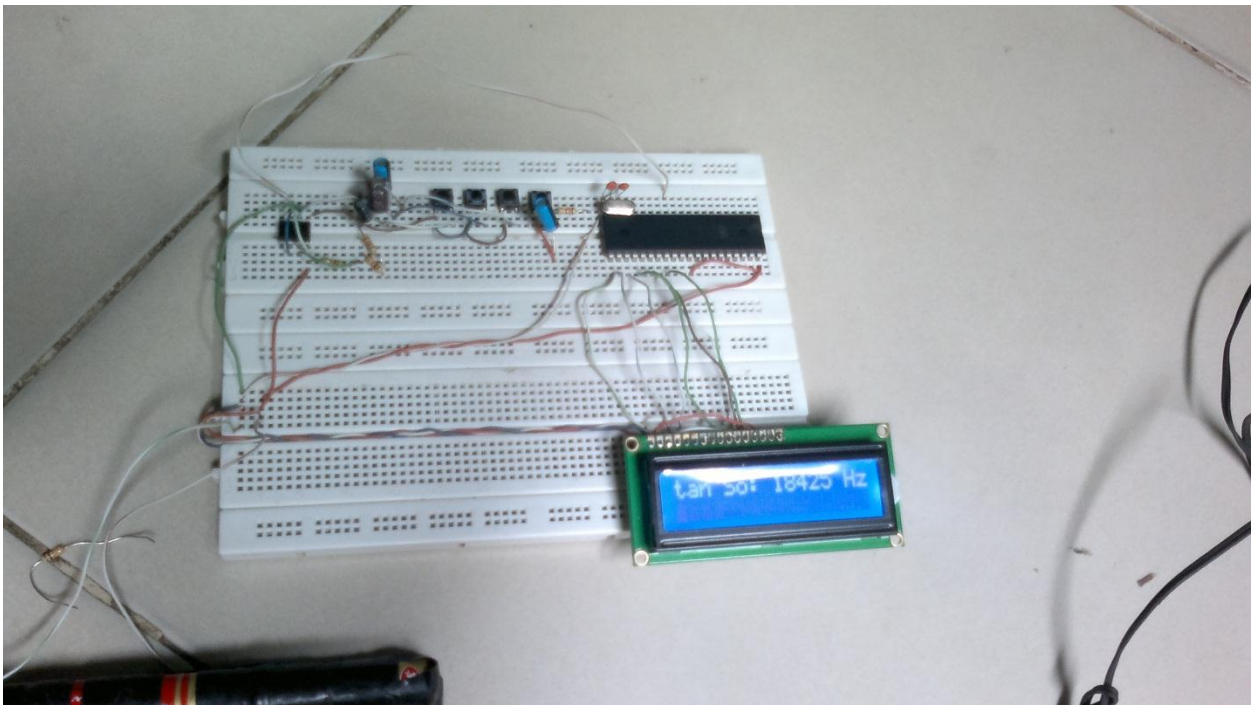
void lcd_Gotoxy(unsigned char x,unsigned char y)//ham nhay dem vi tri x,y trong man hinh lcd

```
{  
    char position;  
    if (y==1)  
    {  
        position=0x80;  
        lcd_command(position|x);  
    }  
}
```



```
    }  
else  
{  
    position=0xC0;  
    lcd_command(position|x);  
}  
}
```

1. Mạch thật



Kết luận

Chúng em xin chân thành cảm ơn sự giúp đỡ tận tình của thầy

Nguyễn Văn Tiến đã giúp chúng em hoàn thành bài tập lớn này. Với kiến thức còn nhiều hạn chế nên bài làm của chúng em vẫn còn nhiều sai sót, rất mong nhận được sự góp ý và chỉ bảo của thầy.