

Dịch từ phải sang trái

```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000) // tần số thạch anh dùng trong vi điều khiển
```

```
unsigned int8 i, d;
```

```
//-----CHUONG TRÌNH CHÍNH-----
```

```
Void main()
```

```
{
```

```
    set_tris_d(0x00); // khai port d xuất
```

```
    d=0x00; // gán giá trị 0000 0000 vào biến d
```

```
    output_d(d); // gán cho port d giá trị của biến d
```

```
    delay_ms(500);
```

```
    while(true)
```

```
    {
```

```
        for(i=0;i<8;i++)
```

```
        {
```

```
            d = (d<<1) + 0x00;
```

```
        output_d(d);
        delay_ms(500);
    }

    for(i=0;i<8;i++)
    {
        d = d<<1,
        output_d(d);
        delay_ms(500);
    }
}
```

Dịch từ trái sang phải

```
#include <16F877A.h>

#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP

#use delay(clock=8000000) // tan so thach anh dung trong vi dieu khien

unsigned int8 i, d;

//-----CHUONG TRÌNH CHÍNH-----

Void main()
{
    set_tris_d(0x00); // khai port d xuất
    d=0x00;          // gan gia tri 0000 0000 vao bien d
    output_d(d);     // gan cho port d gia tri cua bien d
    delay_ms(500);
    while(true)
    {
        for(i=0;i<8;i++)
        {
            d = (d>>1) + 0x80; //1000 0000
```

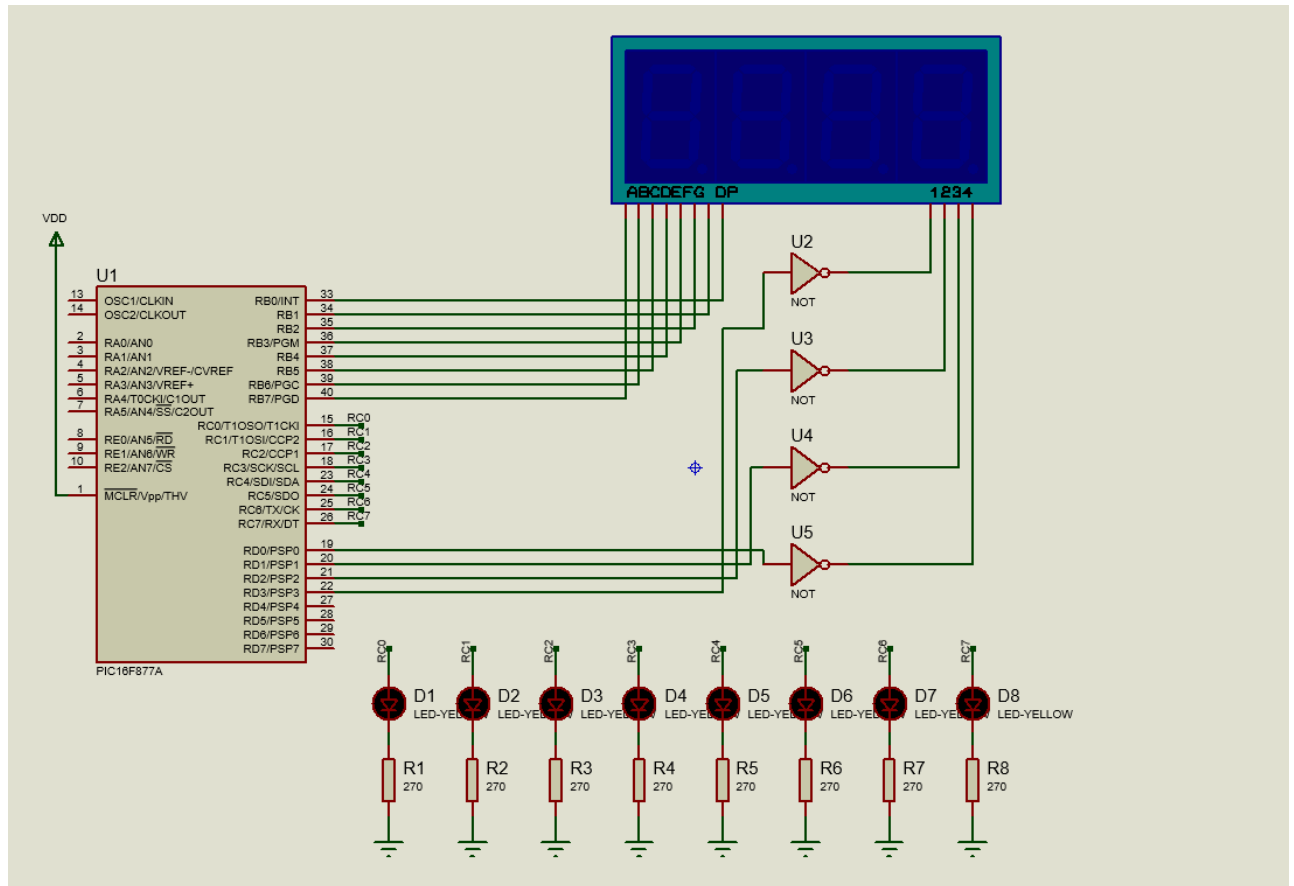
```
    output_d(d);  
    delay_ms(500);  
}
```

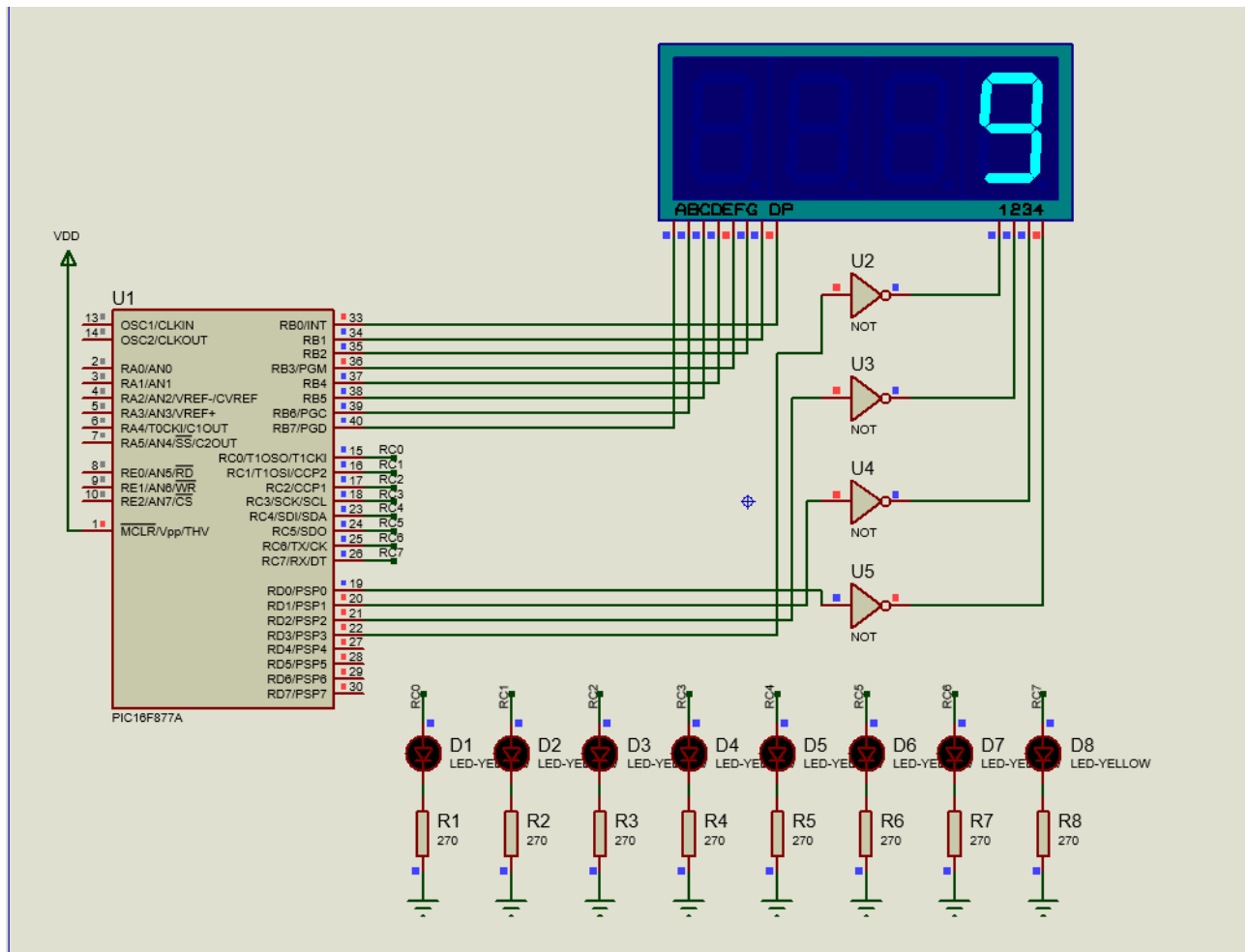
```
for(i=0;i<8;i++)  
{  
    d = (d>>1);  
    output_d(d);  
    delay_ms(500);  
}  
}
```

```
}
```

CHÚ Ý LED 7 ĐOẠN

Hiển thị 1 số trên led 7 đoạn





+tùr 0-9:

abcdefgdp

- 0 00000011 0x03
- 1 10011111 0x9f
- 2 00100101 0x25
- 3 00001101 0x0d
- 4 10011001 0x99
- 5 01001001 0x49
- 6 01000001 0x41
- 7 00011111 0x1f
- 8 00000001 0x01
- 9 00001001 0x09

```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000) // tần số thạch anh thường dùng trong hệ thống
```

```
//-----CHƯƠNG TRÌNH CHÍNH-----
```

```
Void main()
```

```
{
```

```
Output_d(0xff); // tắt các led
```

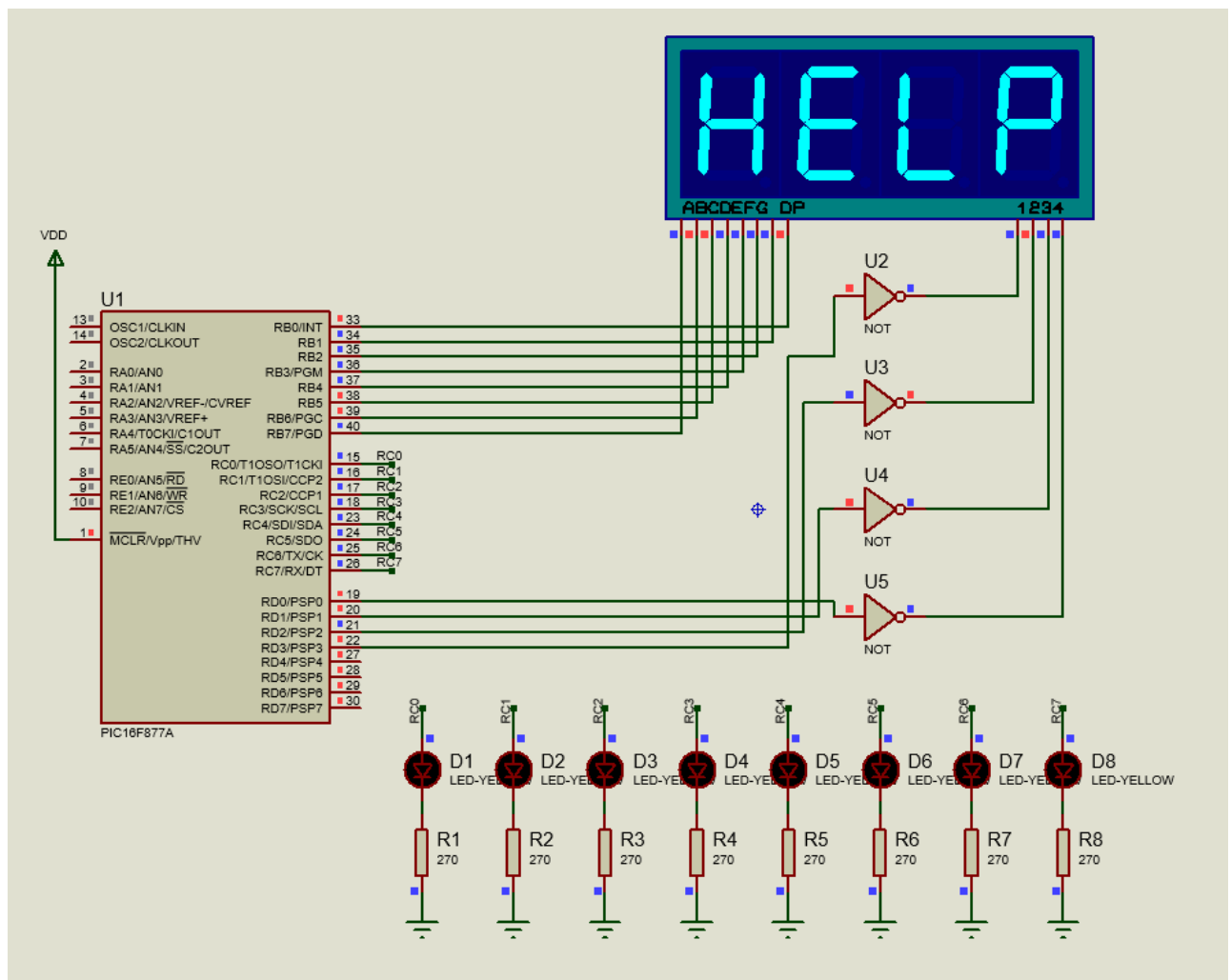
```
Output_b(0x09);
```

```
Output_low(PIN_D0);
```

```
While(1); // STOP
```

```
}
```

2. Viết chương trình hiển thị chữ “HELP” trên 4 led, sử dụng delay 5 ms để quét led.



```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG,  
NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000) // T?n s? th?ch anh dùng  
trong h? th?ng
```

```
//-----CHUONG TRÌNH CHÍNH-----
```

```
CONST unsigned char LED7S[4] = {0x91, 0x61, 0xE3,  
0x31};
```

```
CONST unsigned char dk[4] = {0xf7, 0xfb, 0xfd, 0xfe}; //  
ma dieu khien led
```

```
//    rd7 rd6 rd5 rd4 rd3 rd2 rd1
```

```
// f7 =  1111 0111
```

```
// fb =  1111 1011
```

```
// fd =  1111 1101
```

```
// fe =  1111 1110
```

```
Void quet_LED()
```

```
{
```

```
    for (int8 i = 0; i < 4; i++)
```

```
    {
```

```
        output_d(0xff);
```

```
        output_b(LED7S[i]); // ma led cua HELP
```

```
    output_d(dk[i]);  
    delay_ms(3);  
}  
}
```

//-----CHUONG TRÌNH CHÍNH-----

```
Void main()  
{  
    Output_d(0xff);// t?t các led  
    While (TRUE)  
    {  
        quet_LED(); // goi chuong trinh con quet led  
    }  
}
```

```
//output_d(0xFF);output_b(0x91);output_low(pin_d3);de  
lay_ms(3);
```

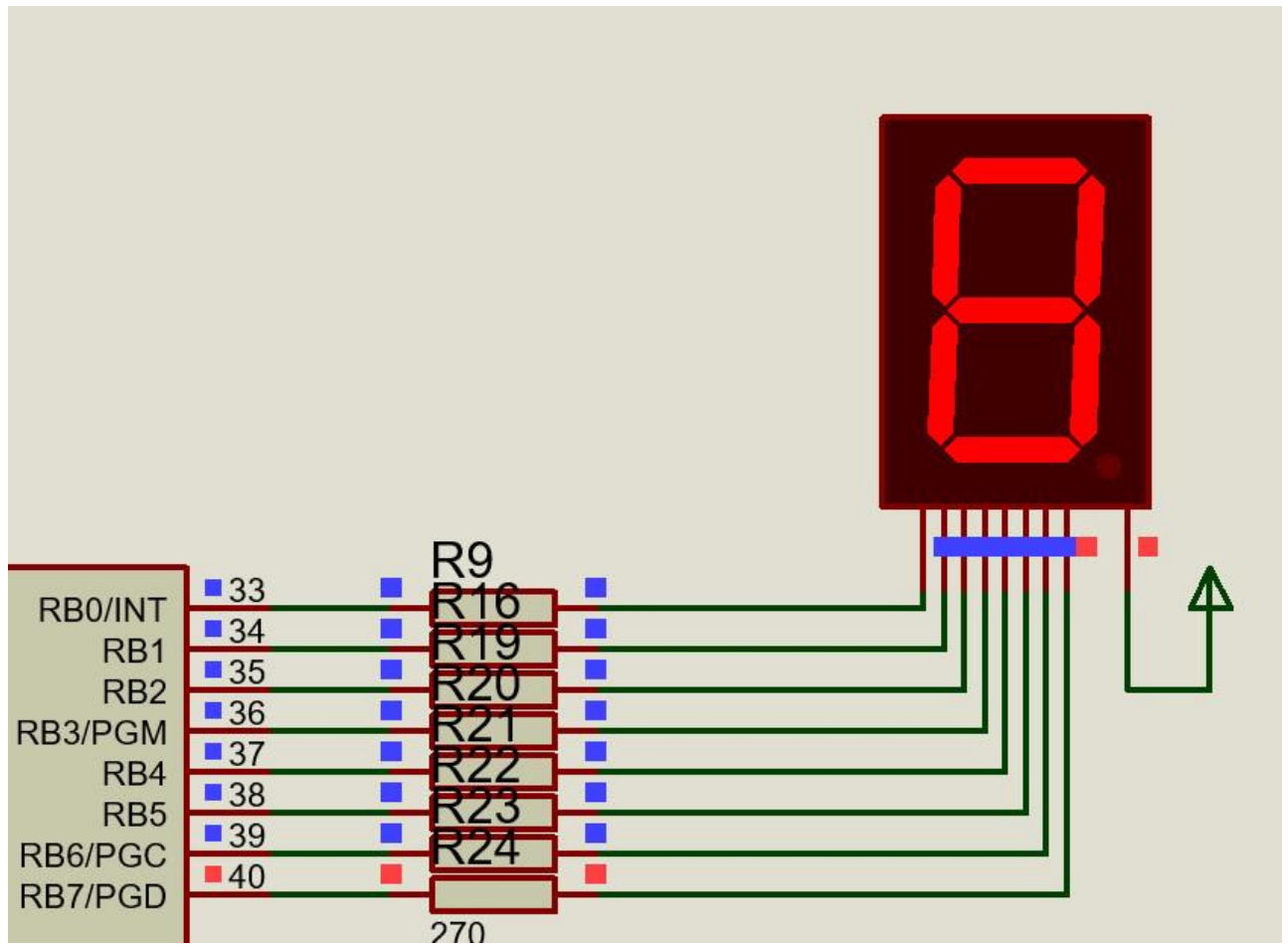
```
//output_d(0xFF);output_b(0x61);output_low(pin_d2);de  
lay_ms(3);
```



```
//output_d(0xFF);output_b(0xE3);output_low(pin_d1);de  
lay_ms(3);
```

```
//output_d(0xFF);output_b(0x31);output_low(pin_d0);de  
lay_ms(3);
```

Đếm từ 0 – 9



```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000) // tần số thạch anh dùng trong vi điều khiển
```

```
const unsigned char ma7doan [10] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90};
```

```
signed int dem;
```

```
unsigned int ma_dem;
```

```
void main()
```

```
{
```

```
    set_tris_b(0x00);
```

```
    while(true)
```

```
    {
```

```
    for(dem=0; dem<10; dem++)
    {
        ma_dem = ma7doan[dem];
        output_b(ma_dem);
        delay_ms(1000);
    }
}
```

Có thể viết gọn hơn

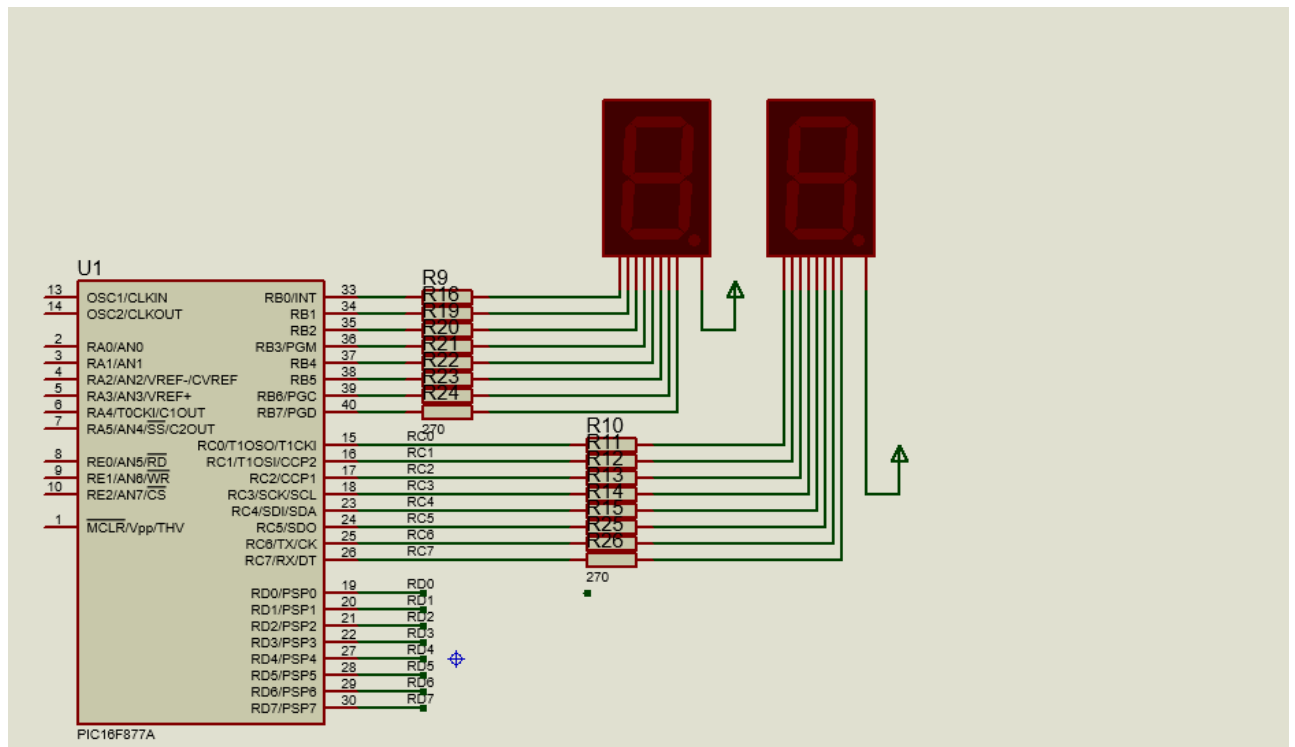
```
#include <16F877A.h>

#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP

#use delay(clock=8000000) // tần số thạch anh dùng trong vi điều khiển

const unsigned char ma7doan [10] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90};
signed int dem;
//unsigned int ma_dem;

void main()
{
    set_tris_b(0x00);
    while(true)
    {
        for(dem=0; dem<10; dem++)
        {
            //ma_dem = ma7doan[dem];
            output_b(ma7doan[dem]);
            delay_ms(1000);
        }
    }
}
```



Dem tu 0 den 99 (đang bị lỗi có số 0 vô nghĩa nhảy sang hàng chục)

```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000) // tan so thach anh dung trong vi diu khien
```

```
const unsigned char ma7doan [10] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90};
```

```
signed int dem;
```

```
//unsigned int ma_dem;
```

```
void main()
```

```
{
```

```
    set_tris_b(0x00);
```

```
    set_tris_c(0x00);
```

```
    while(true)
```

```
    {
```

```
        for(dem=0; dem<100; dem++)
```

```
        {
```

```
            //ma_dem = ma7doan[dem];
```

```
            output_c(ma7doan[dem%10]); // hang don vi % lay phan du
```

```
    delay_ms(500);  
    output_b(ma7doan[dem/10]); // hang chuc / lay phan nguyen  
    delay_ms(500);  
}  
}  
}
```

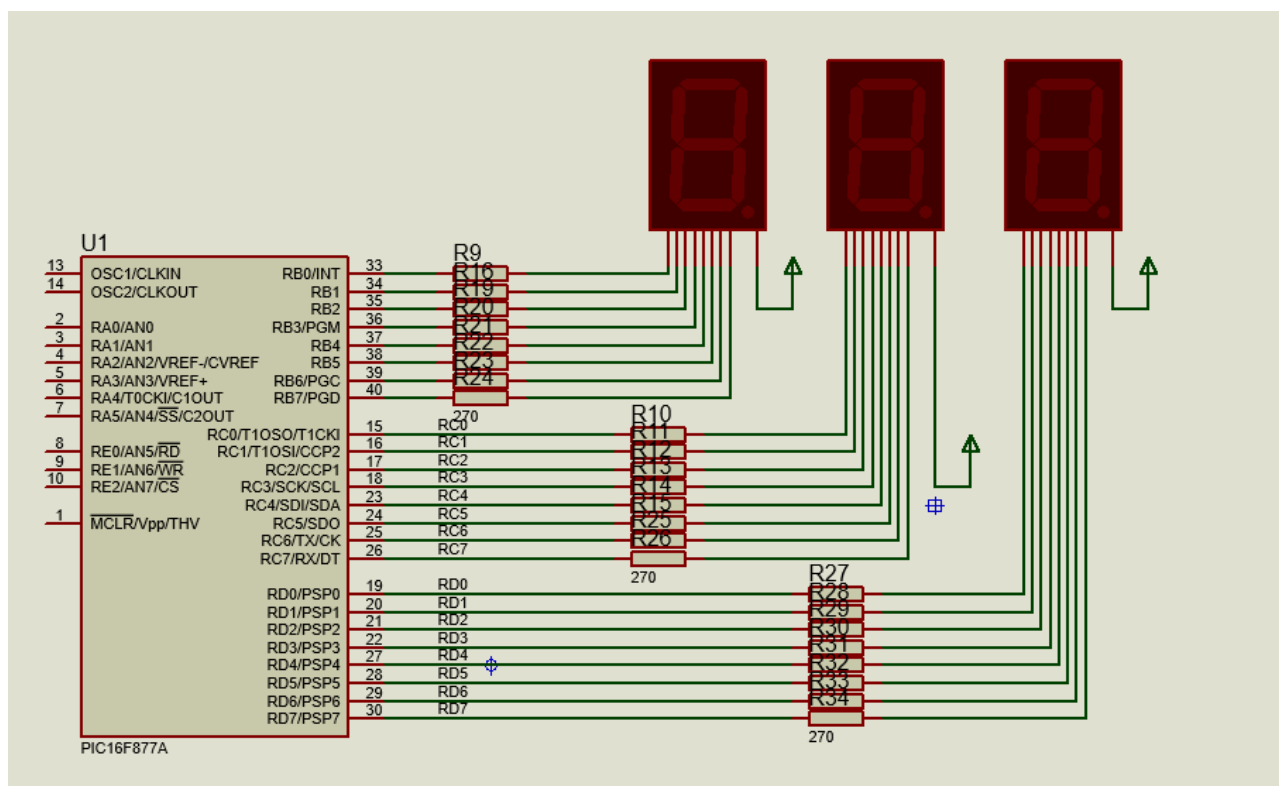
Dem tu 0 den 99 (xóa số 0 vô nghĩa)

```
#include <16F877A.h>  
  
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP  
  
#use delay(clock=8000000) // tan so thach anh dung trong vi dieu khien  
  
const unsigned char ma7doan [10] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90};  
signed int dem;  
//unsigned int ma_dem;  
  
void main()  
{  
    set_tris_b(0x00);  
    set_tris_c(0x00);  
    while(true)  
    {  
        for(dem=0; dem<100; dem++)  
        {  
            //ma_dem = ma7doan[dem];  
            output_c(ma7doan[dem%10]); // hang don vi % lay phan du  
            if (dem/10!=0) // xoa so 0 vo nghia  
                output_b(ma7doan[dem/10]);  
            else output_b(0xff);  
            delay_ms(500);  
            // output_b(ma7doan[dem/10]); // hang chuc / lay phan nguyen  
            // delay_ms(500);  
        }  
    }  
}
```

```

}
}

```



Đếm từ 0 – 999

```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000) // tần số thạch anh dùng trong vi điều khiển
```

```
const unsigned char ma7doan [10] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90};
```

```
signed int16 dem;
```

```
unsigned int16 ma_tr, ma_ch, ma_dv;
```

```
// chương trình con
```

```
void giaima_3so_xvn(signed int16 x)
```

```
{
```

```
    ma_dv = ma7doan[x%10]; // hàng đơn vị 123%10 = 3
```

```
    ma_ch = ma7doan[x/10%10]; // hàng chục 123/10 = 12, 12%10 = 2
```

```
ma_tr = ma7doan[x/100%10]; // hang tram 123/100 = 1, 1 %10 = 1
```

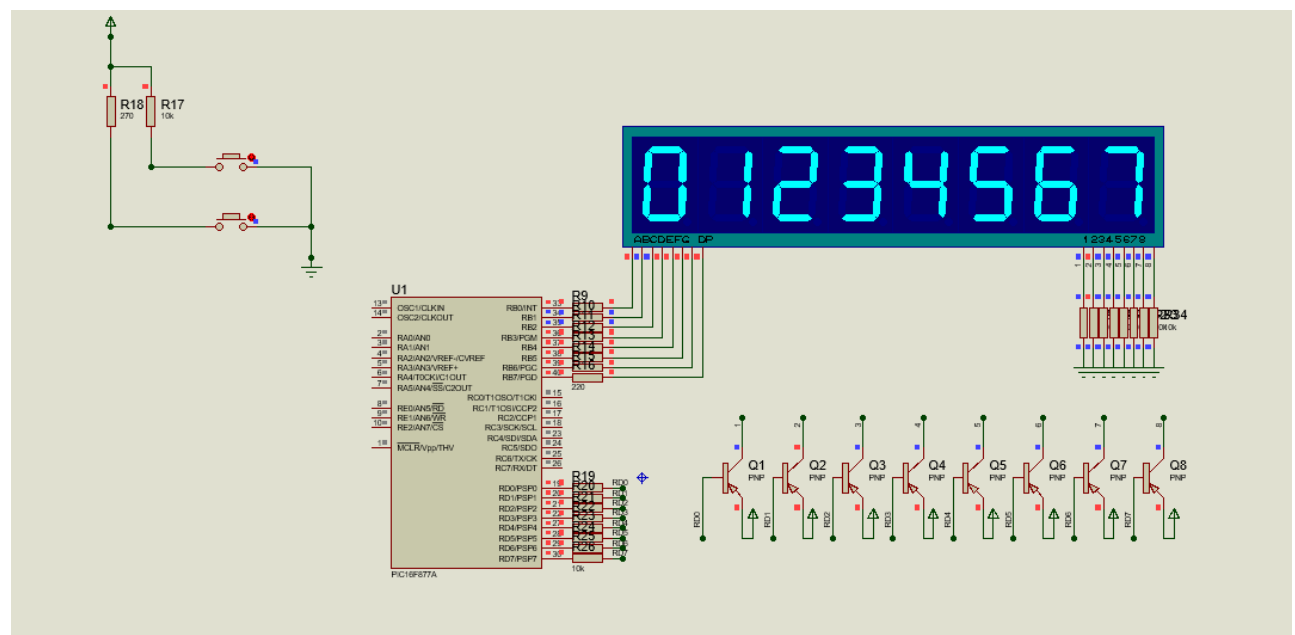
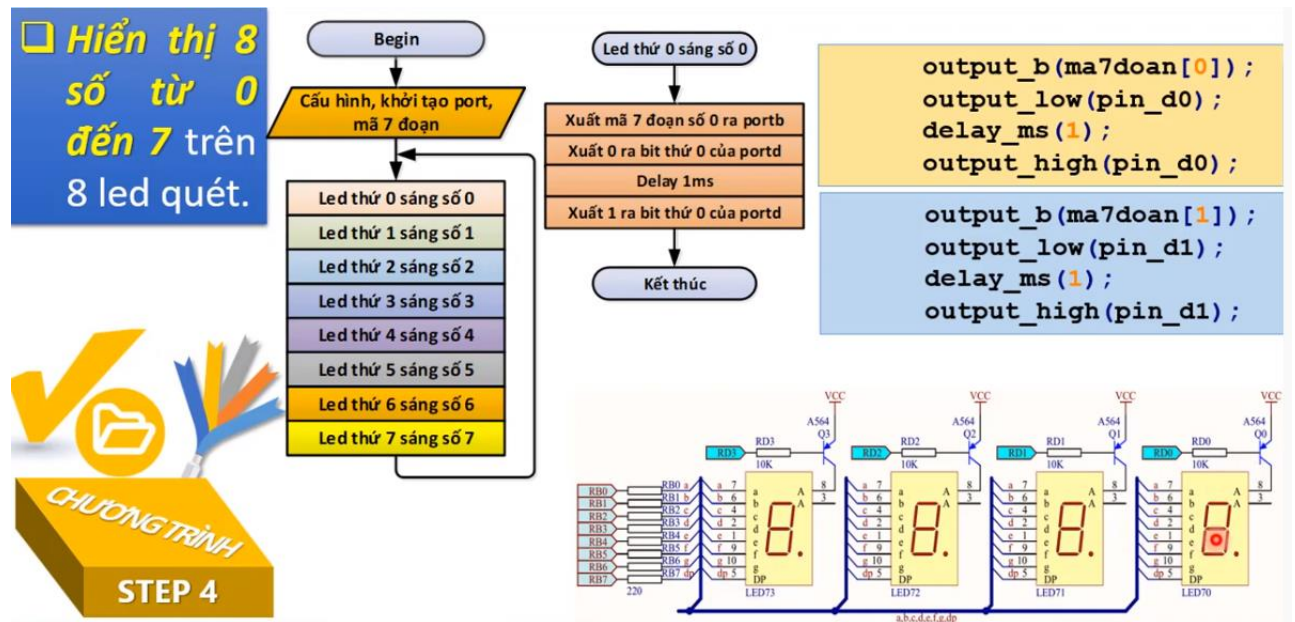
```
if(ma_tr==0xc0) // 0xc0 = 0
{
    ma_tr=0xff;
    if(ma_ch==0xc0) ma_ch=0xff;
}
}
```

```
void hienthi_3so_3port()
{
    output_d(ma_dv);
    output_c(ma_ch);
    output_b(ma_tr);
}
```

```
void main()
{
    set_tris_d(0x00);
    set_tris_c(0x00);
    set_tris_b(0x00);
    while(true)
    {
        for(dem=0;dem<1000;dem++)
        {
            giamai_3so_xvn(dem);
            hienthi_3so_3port();
            delay_ms(100);
        }
    }
}
```

LED 7 ĐOẠN Dùng phương pháp quét

Bài 1: Hiển thị số 0 đến 7 tương ứng với 8 led 7 đoạn



```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=20M) // tần số thạch anh dùng trong vi điều khiển
```

```
//const unsigned char ma7doan[16]= {0XC0,0XF9,0xA4,0XB0,0X99,0X92,0X82,0XF8,
```

```
// 0X80,0X90,0x88, 0x83, 0xC6, 0xA1, 0x86, 0x8E};
```

```
const unsigned char ma7doan[10]= {0XC0,0XF9,0xA4,0XB0,0X99,0X92,0X82,0XF8,
```

```
0X80,0X90};
```



```
//signed int8 j;
```

```
void main()
```

```
{
```

```
    set_tris_b(0x00);
```

```
    set_tris_d(0x00);
```

```
    while(true)
```

```
    {
```

```
        output_b(ma7doan[0]); // tuong ung ma 0xC0 la so 0
```

```
        output_low(pin_d0); // lay muc '0' dieu khien con led thu 0 , led thu 0 noi voi port D0
```

```
        delay_ms(1);
```

```
        output_high(pin_d0); // tat portD0 sau khi quet
```

```
        output_b(ma7doan[1]); // tuong ung ma 0XF9 la so 1
```

```
        output_low(pin_d1); // lay muc '0' dieu khien con led thu 1, led thu 1 noi voi port D1
```

```
        delay_ms(1);
```

```
        output_high(pin_d1);
```

```
        output_b(ma7doan[2]);
```

```
        output_low(pin_d2);
```

```
        delay_ms(1);
```

```
        output_high(pin_d2);
```

```
        output_b(ma7doan[3]);
```

```
        output_low(pin_d3);
```

```
        delay_ms(1);
```

```
        output_high(pin_d3);
```

```
        output_b(ma7doan[4]);
```

```
        output_low(pin_d4);
```

```
        delay_ms(1);
```

```
        output_high(pin_d4);
```

```

output_b(ma7doan[5]);
output_low(pin_d5);
delay_ms(1);
output_high(pin_d5);

```

```

output_b(ma7doan[6]);
output_low(pin_d6);
delay_ms(1);
output_high(pin_d6);

```

```

output_b(ma7doan[7]);
output_low(pin_d7);
delay_ms(1);
output_high(pin_d7);

```

```

}

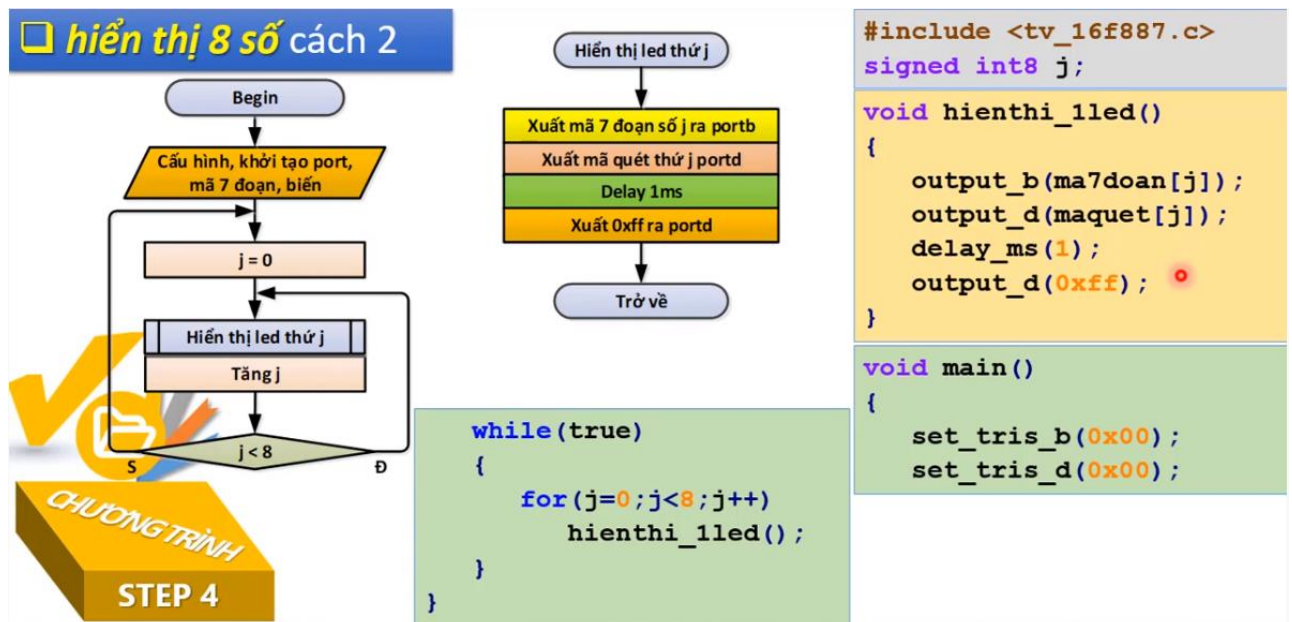
```

```

}

```

Cách 2 : đi thi nên áp dụng cách 2



Lưu ý đến mã quét

J = 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7

```

const unsigned char ma7doan [10] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82,
0xf8, 0x80, 0x90};

const unsigned char maquet [8] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf, 0xbf, 0x7f};

void quetled()
{
    output_b(ma7doan[j]);
    output_d(maquet[j]);
    delay_ms(1);
    output_d(0xff);
}

```

PORT

1	1	1	1	1	1	1	1
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0

J= 0 thì lấy ma7doan = 0xc0 là số 0

J= 0 thì lấy maquet = 0xfe là gì ? 1111 1110 xuất bit 0 vào pin D0 để điều khiển led thứ 0 tương đương với lệnh output_low(pin_d0) của cách 1

J = 1 thì lấy ma7doan = 0xf9 là số 1

J = 1 thì lấy maquet = 0xfd là gì ? 1111 1101 xuất bit 1 vào pin D1 để điều khiển led thứ 1 tương đương với lệnh output_low(pin_d1) của cách 1

J = 2 thì lấy ma7doan = 0xa4 là số 2

J = 2 thì maquet = 0xfb ? 1111 1011 xuất bit 1 vào pin D2 để điều khiển led thứ 2 tương đương với lệnh output_low(pin_d2) của cách 1

Tương tự J=3,4,5,6,7.

```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=20000000) // tan so thach anh dung trong vi dieu khien
```

```

//const unsigned char ma7doan [16] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82,
// 0xf8, 0x80, 0x90, 0x88, 0x83, 0xc6, 0xa1, 0x86, 0x8e};

```

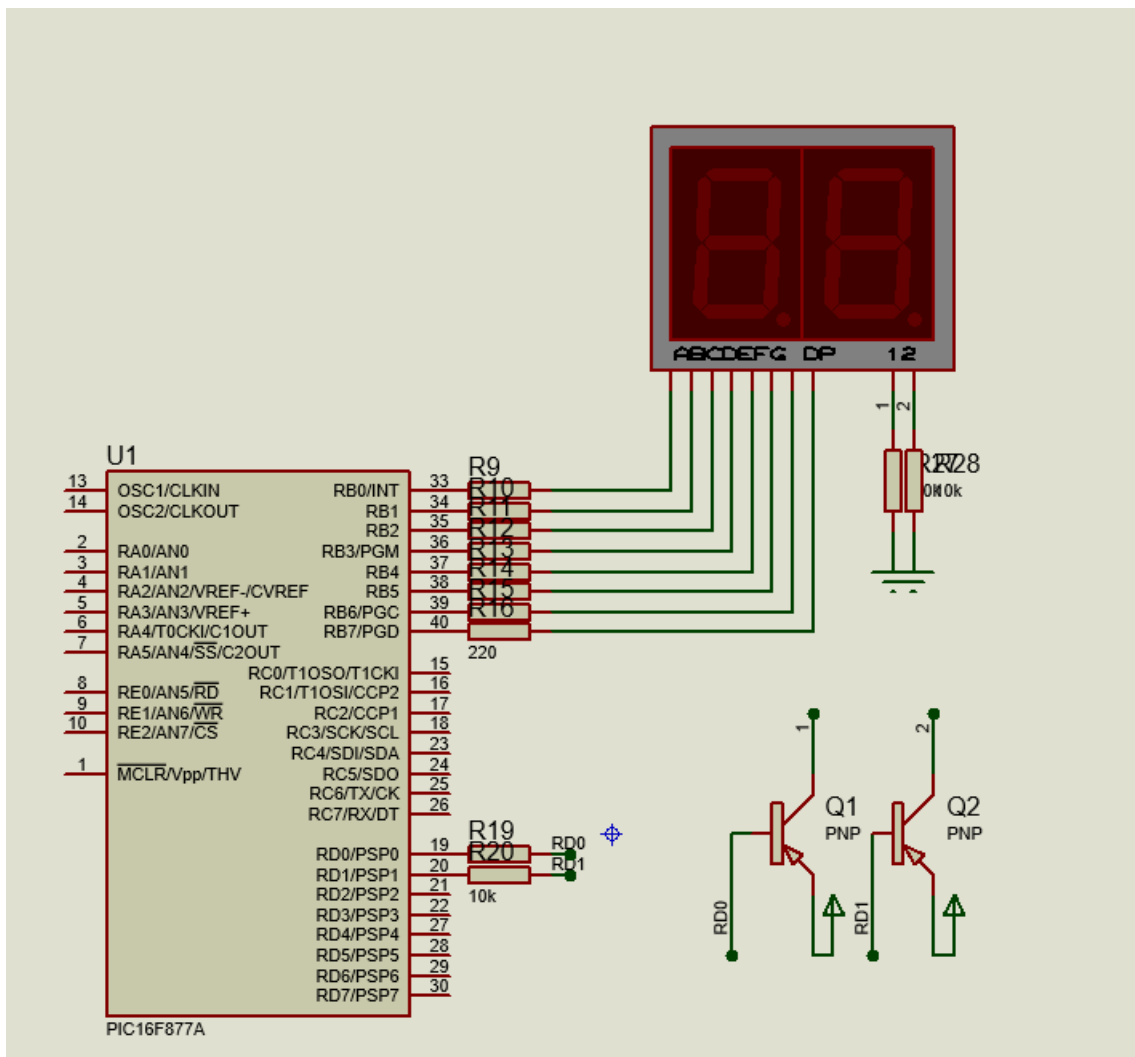
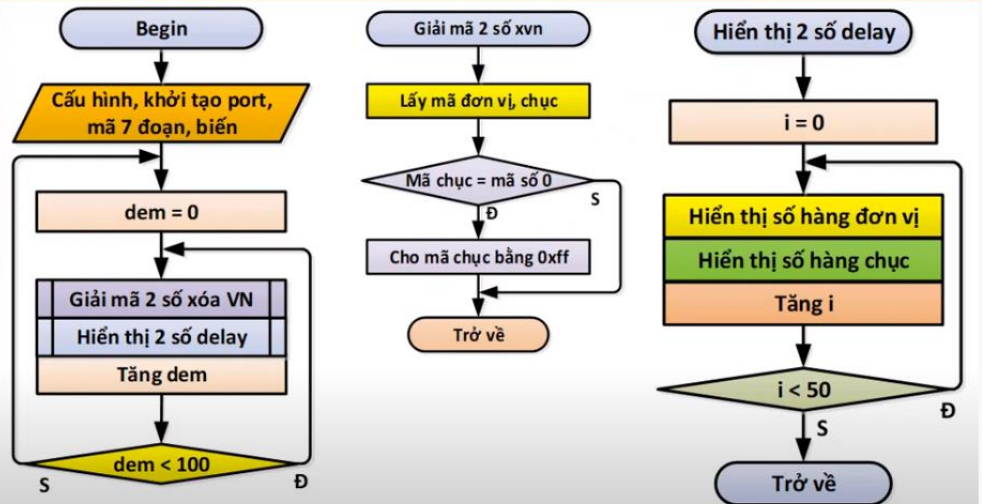
```

const unsigned char ma7doan [10] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82,
0xf8, 0x80, 0x90};

```

```
const unsigned char maquet [8] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf, 0xbf, 0x7f};
```


□ Hãy vẽ lưu đồ **đếm từ 0 đến 99** với thời gian trễ tùy chọn.



```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=20000000) // tần số thạch anh dùng trong vi điều khiển
```

```
//const unsigned char ma7doan [16] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82,
```

```
//0xf8, 0x80, 0x90, 0x88, 0x83, 0xc6, 0xa1, 0x86, 0x8e};
```

```
const unsigned char ma7doan [10] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82,  
0xf8, 0x80, 0x90};
```

```
const unsigned char maquet[8] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf, 0xbf, 0x7f};
```

```
unsigned int8 dem, i, ma_ch, ma_dv;
```

```
void giama_2so_xvn(unsigned int x)  
{  
    ma_dv = ma7doan[x%10]; // 19 % 10 = 9 so 9 la hang don vi  
    ma_ch = ma7doan[x/10%10]; // 19/ 10 = 1 , 1 % 10 = 1 la hang chuc  
    if(ma_ch==0xc0) ma_ch =0xff; // xoa so 0 vo nghia  
}
```

```
void hienthi_2so_delay()  
{  
    for(i=0;i<50;i++)  
    {  
        output_b(ma_dv);  
        output_low(pin_d1); // xem so do phan cung  
        delay_ms(1);  
        output_high(pin_d1);  
  
        output_b(ma_ch);  
        output_low(pin_d0); // // xem so do phan cung  
        delay_ms(1);  
        output_high(pin_d0);  
  
    }  
}
```

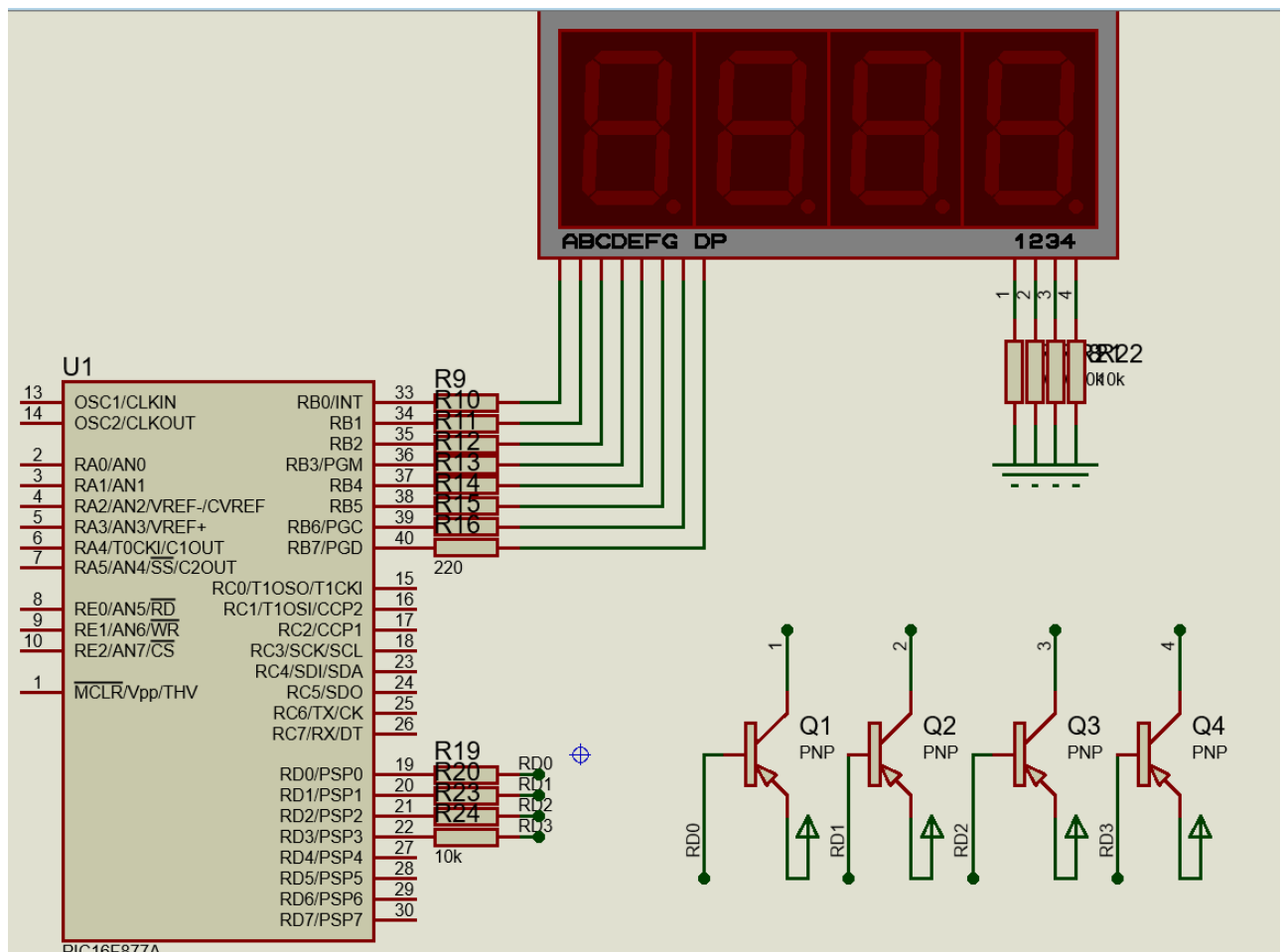
```
void main()  
{
```

```

set_tris_b(0x00);
set_tris_d(0x00);
while(true)
{
    for(dem=0;dem<100;dem++)
    {
        giamma_2so_xvn(dem);
        hienthi_2so_delay();
    }
}

```

Bài số 3: đếm từ 0 đến 999 dùng phương quét



```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=20000000) // tần số thạch anh dùng trong vi điều khiển
```

```
const unsigned char ma7doan [16] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82,  
0xf8, 0x80, 0x90, 0x88, 0x83, 0xc6, 0xa1, 0x86, 0x8e};  
const unsigned char maquet[8] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf, 0xbf, 0x7f};
```

```
unsigned int16 dem, i, ma_tr, ma_ch, ma_dv;
```

```
void giama_3so_xvn(unsigned int x)
```

```
{  
    ma_dv = ma7doan[x%10];  
    ma_ch = ma7doan[x/10%10];  
    ma_tr = ma7doan[x/100%10];  
    if(ma_tr==0xc0)  
    {  
        ma_tr = 0xff;  
        if (ma_ch==0xc0) ma_ch=0xff;  
    }  
}
```

```
void hienthi_3so_delay()
```

```
{  
    for(i=0;i<20;i++)  
    {  
        output_b(ma_dv);  
        output_low(pin_d2);  
        delay_ms(1);  
        output_high(pin_d2);  
  
        output_b(ma_ch);  
        output_low(pin_d1);  
        delay_ms(1);  
        output_high(pin_d1);  
  
        output_b(ma_tr);  
        output_low(pin_d0);
```



```

    delay_ms(1);
    output_high(pin_d0);

}

}

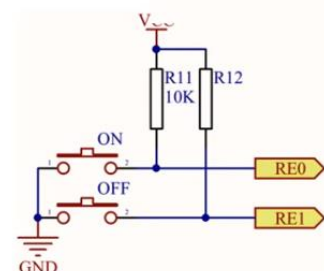
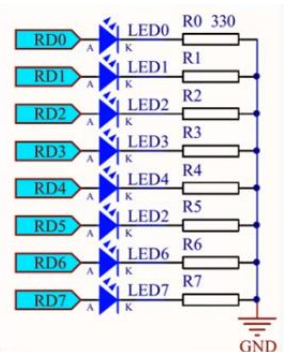
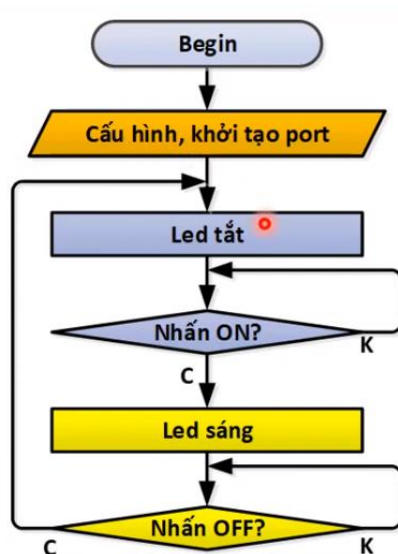
void main()
{
    set_tris_b(0x00);
    set_tris_d(0x00);
    while(true)
    {
        for(dem=0;dem<1000;dem++)
        {
            giama_3so_xvn(dem);
            hienthi_3so_delay();
        }
    }
}

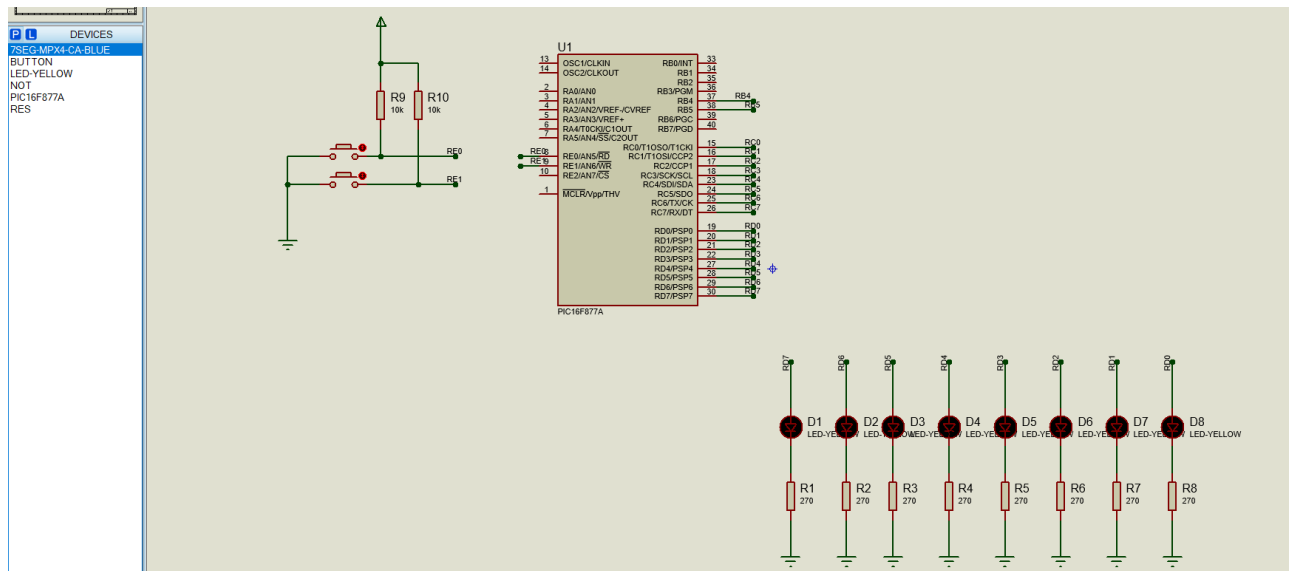
```

III – Nút nhấn

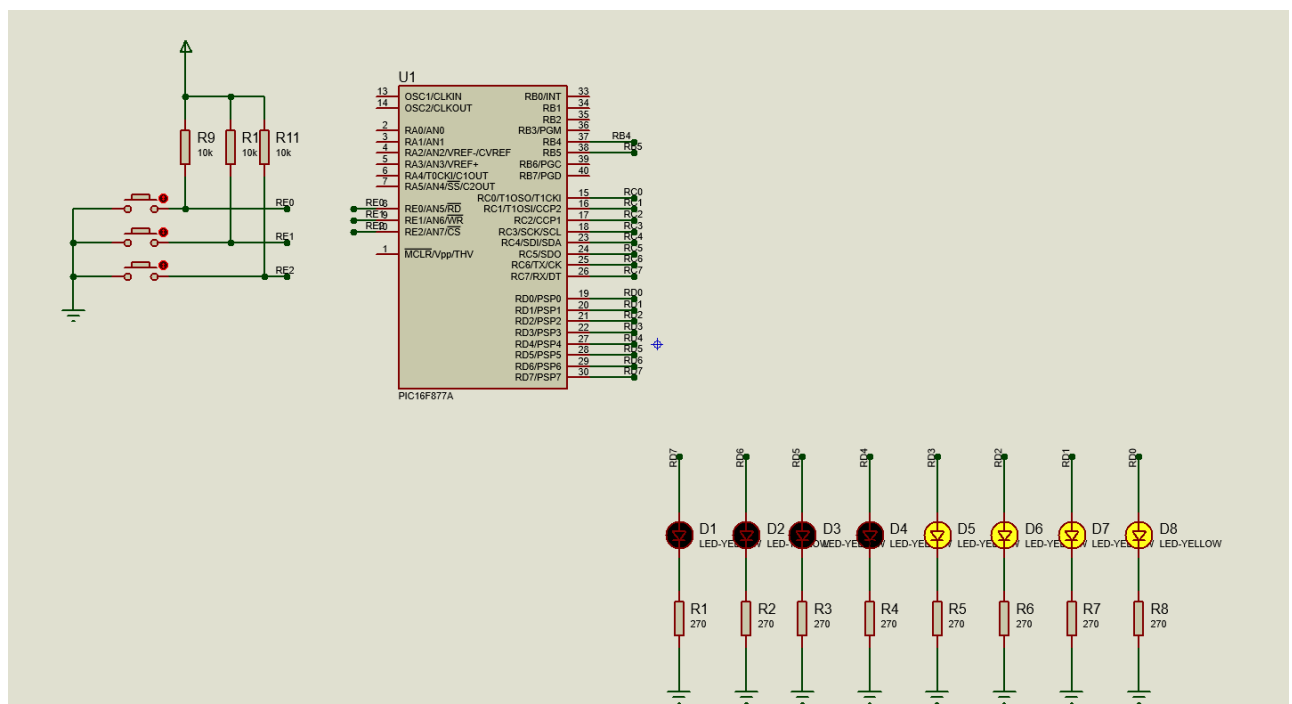
Khi nhấn On 8 led sáng, khi nhấn OFF thì 8 led tắt

Khi nhấn ON thì 8 led sáng, khi nhấn OFF thì 8 led tắt.





Bài 02: 3 nút nhấn on, off, inv (không có chống dội)



```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=20000000) // tần số thạch anh dùng trong vi điều khiển
```

```
#define on pin_e0
```

```
#define off pin_e1
```

```
#define inv pin_e2
```

```
unsigned int8 d=0;
```

```
void main()
```

```
{
```

```
    set_tris_e(0x07);
```

```
    set_tris_d(0x00);
```

```
    while(true)
```

```
    {
```

```
        d=0; output_d(d);
```

```
        while(input(on));
```

```
        d=0x0f; output_d(d);
```

```
        do{
```

```
            if(!input(inv))
```

```
            {
```

```
                d=~d;
```

```
                output_d(d);
```

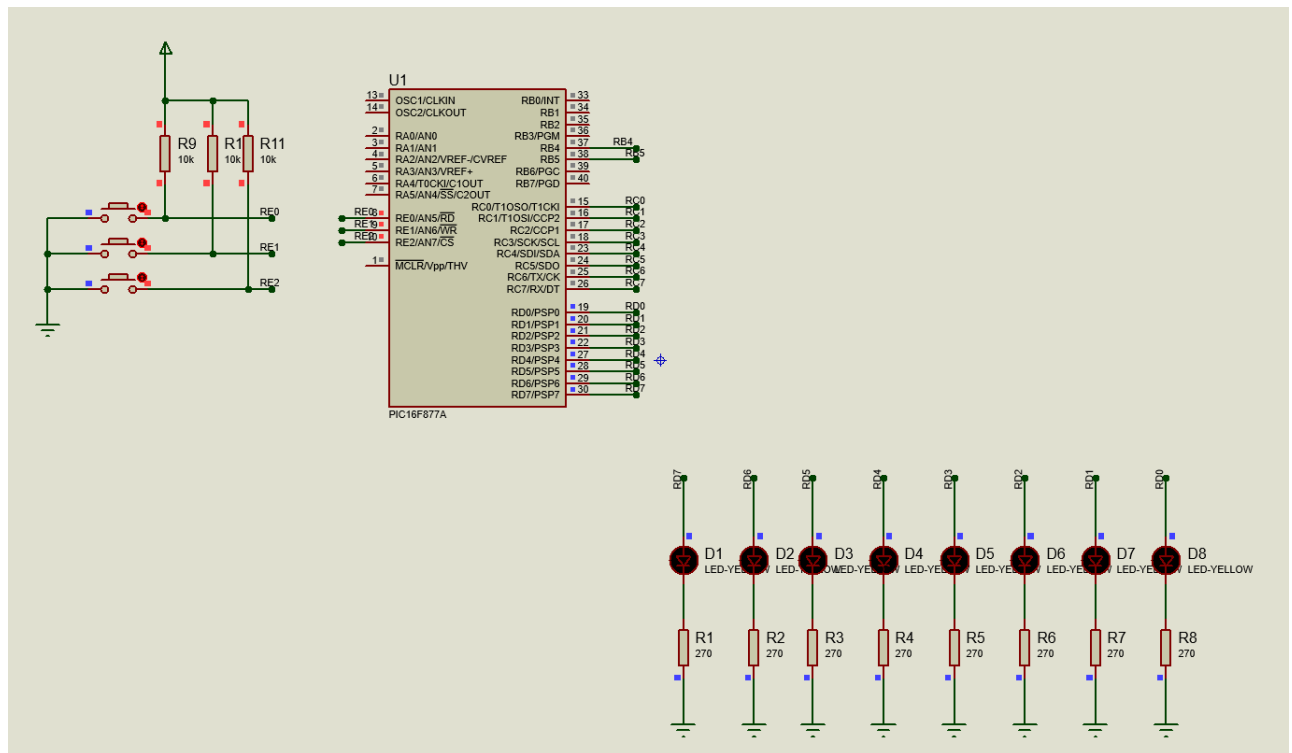
```
            }
```

```
        }while(input(off));
```

```
    }
```

```
}
```

Bài 3: 3 nút on, off, inv có thêm **hàm con chống dội phím**



```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=20000000) // tan so thach anh dung trong vi dieu khien
```

```
#define on pin_e0
```

```
#define off pin_e1
```

```
#define inv pin_e2
```

```
unsigned int8 d=0;
```

```
void chongdoi_inv()
```

```
{
```

```
    if(!input(inv))
```

```
    {
```

```
        delay_ms(20);
```

```
        if(!input(inv))
```

```
        {
```

```
            d=~d;
```

```
            output_d(d);
```

```
            while(!input(inv));
```

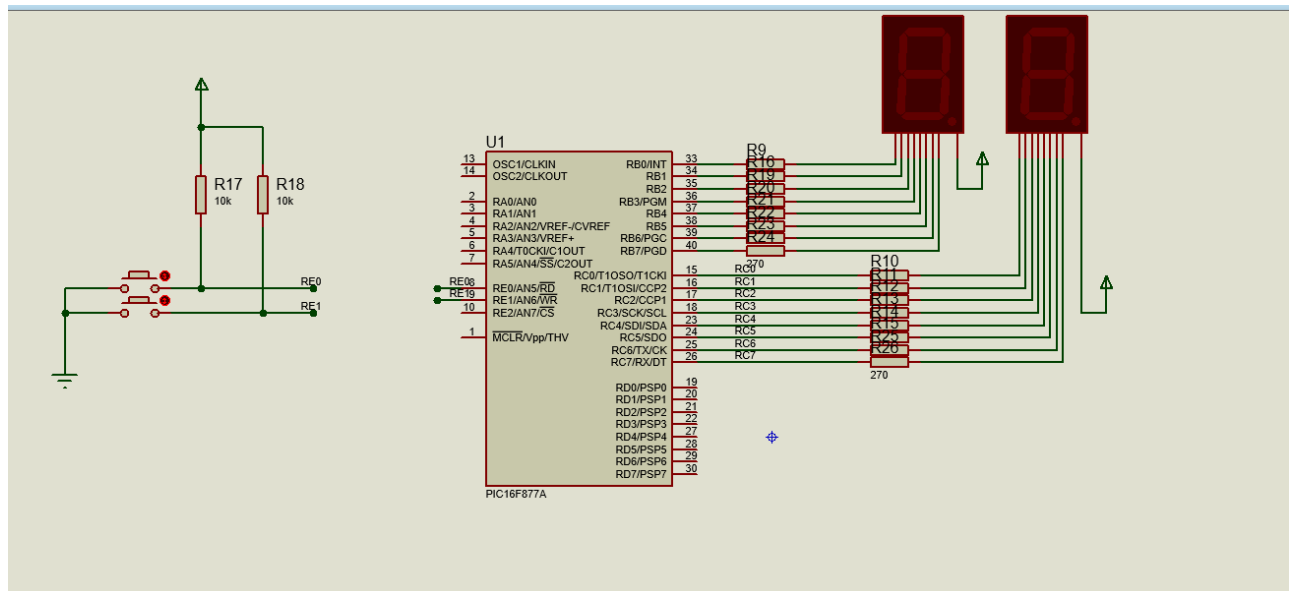
```
        }
```

```
}
```

```
    }  
}  
void main()  
{  
    set_tris_e(0x07);  
    set_tris_d(0x00);  
    while(true)  
    {  
        d=0; output_d(d);  
        while(input(on));  
        d=0x0f; output_d(d);  
        do{  
            //  if(!input(inv))  
            //  {  
            //      d=~d;  
            //      output_d(d);  
            //  }  
            chongdoi_inv();  
        }while(input(off));  
    }  
}
```

Bài 4: Viết chương trình 2 phím tăng giảm từ 00 đến 99 trên 2 led 7 đoạn dùng phương pháp nối trực tiếp.

Có hàm chống dội phím bằng phương tăng giảm



```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=20000000) // tan so thach anh dung trong vi dieu khien
```

```
#define up pin_e0;
```

```
#define dw pin_e1;
```

```
const unsigned char ma7doan [16] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82,  
0xf8, 0x80, 0x90, 0x88, 0x83, 0xc6, 0xa1, 0x86, 0x8e};
```

```
// const unsigned char maquet [8] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf, 0xbf, 0x7f};
```

```
unsigned int8 gh =0;
```

```
void giaima_hienthi_2so();
```

```
void phim_up();
```

```
void phim_dw();
```

```
void main()
```

```
{
```

```
    set_tris_b(0x00);
```

```
    set_tris_c(0x00);
```

```
    set_tris_e(0x03);
```

```
    gh=0;
```

```
giaima_hienthi_2so();  
while(true)  
{  
    phim_up();  
    phim_dw();  
}  
}
```

```
void giaima_hienthi_2so()  
{  
    output_c(ma7doan[gh%10]);  
    if(gh/10!=0)  
        output_b(ma7doan[gh/10]);  
    else output_b(0xff);  
}
```

```
void phim_up()  
{  
    if(!input(pin_e0))  
    {  
        delay_ms(20); // chong doi phim  
        if(!input(pin_e0))  
        {  
            if(gh<99) gh++;  
            giaima_hienthi_2so();  
            delay_ms(200);  
        }  
    }  
}
```

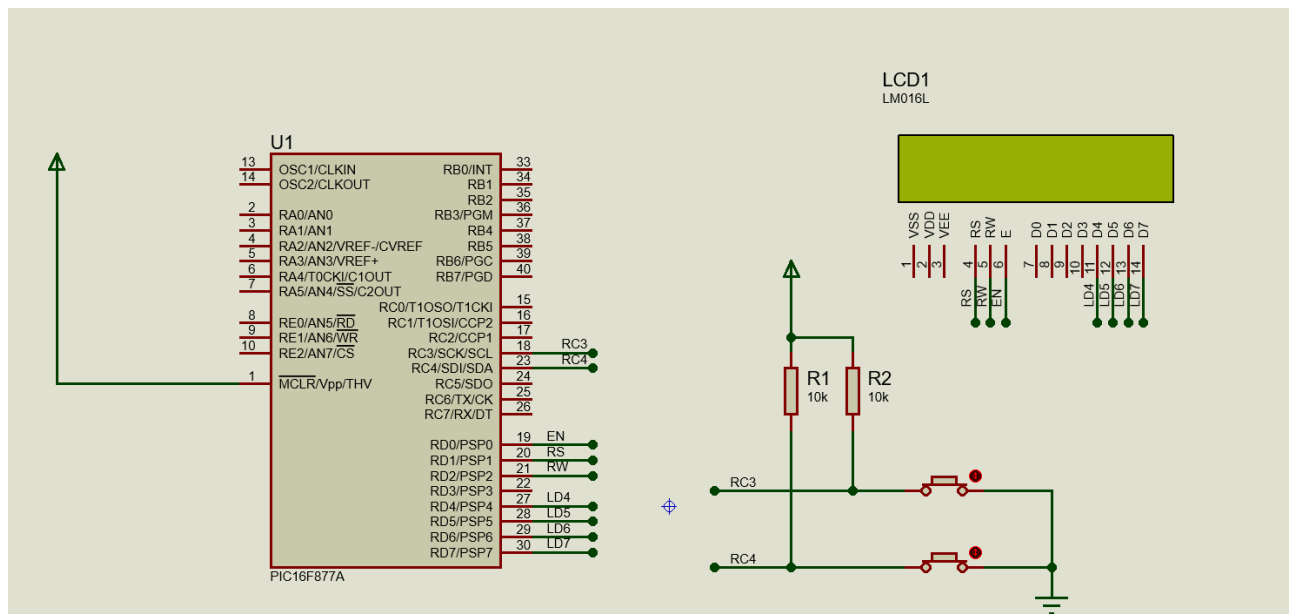
```
void phim_dw()  
{
```

```

if(!input(pin_e1))
{
    delay_ms(20); // chong doi phim
    if(!input(pin_e1))
    {
        if(gh>0) gh--;
        giamai_hienthi_2so();
        delay_ms(200);
    }
}
}
}

```

LCD



1. Viết chương trình hiển thị chữ: “Welcome to” và “***HUTECH*****” trên dòng 01 và 02 của LCD**


```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT,  
NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000)
```

```
#include <lcd.c> // khai bao thu vien lcd
```

```
//=====
```

```
void main()
```

```
{
```

```
    lcd_init(); // khoi tao lcd
```

```
    set_tris_d(0x00); //set port D xuat
```

```
    output_d(0x00); // khai bao port D
```

```
    while(TRUE)
```

```
    {
```

```
        lcd_gotoxy(1,1); // Di chuy?n d?n v? trí d?u  
        dòng 01 LCD
```

```
        //printf(lcd_putc,"Welcome to");
```

```
        lcd_putc("DUY LINH");
```

```
        delay_ms(3);
```

```
lcd_gotoxy(1,2); //Di chuy?n d?n d?u dòng
02 LCD
```

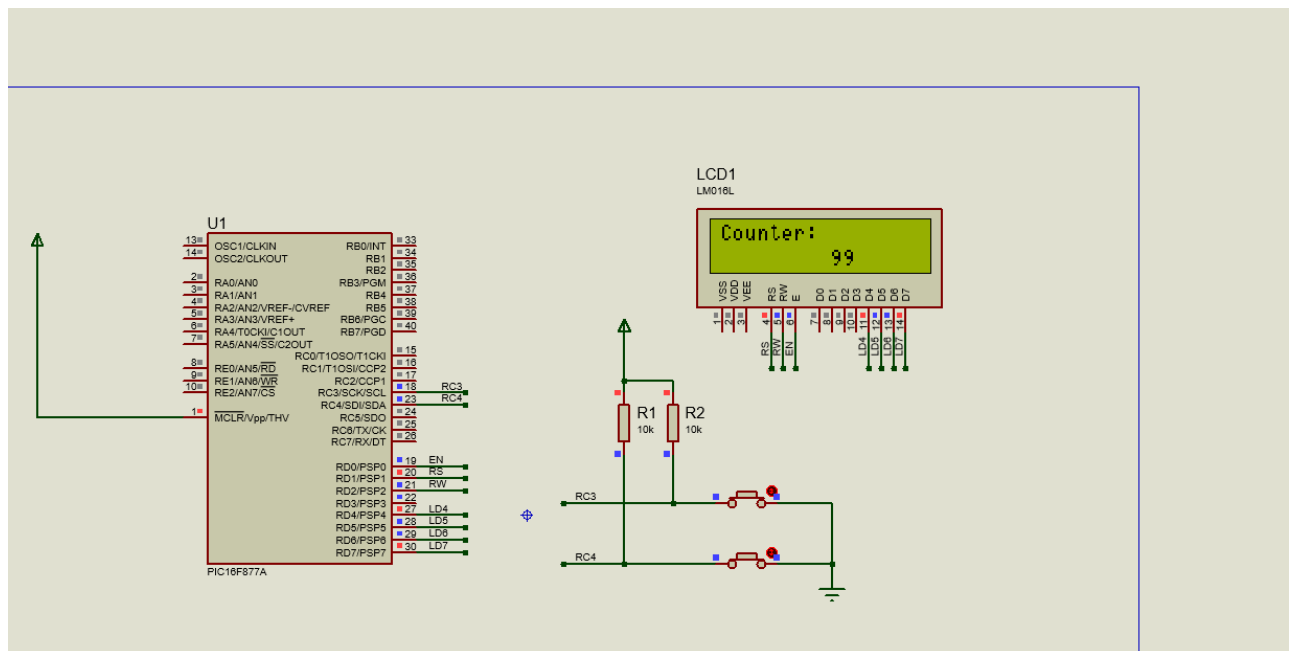
```
printf(lcd_putc,"HUTECH*****");
```

```
delay_ms(3);
```

```
}
```

```
}//end main
```

2. Viết chương trình hiển thị chữ: “Counter:” trên dòng 1 và số đếm từ 00 – 99 trên dòng 2 từ tọa độ (x,y) = (10,2); thời gian delay để tăng số đếm là 300 ms.



Format:

The format takes the generic form %nt. n is optional and may be 1-9 to specify how many characters are to be outputted, or 01-09 to indicate leading zeros, or 1.1 to 9.9 for floating point and %w output. t is the type and may be one of the following:

c	Character
s	String or character
u	Unsigned int
d	Signed int
Lu	Long unsigned int
Ld	Long signed int
x	Hex int (lower case)
X	Hex int (upper case)
Lx	Hex long int (lower case)
LX	Hex long int (upper case)
f	Float with truncated decimal
g	Float with rounded decimal
e	Float in exponential format
w	Unsigned int with decimal place inserted. Specify two numbers for n. The first is a total field width. The second is the desired number of decimal places.

Example formats:

Specifier	Value=0x12	Value=0xfe
%03u	018	254
%u	18	254
%2u	18	*
%5	18	254
%d	18	-2
%x	12	fe
%X	12	FE
%4X	0012	00FE
%3.1w	1.8	25.4

* Result is undefined - Assume garbage.

```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT,  
NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000)
```

```
#include <lcd.c>
```

```
//=====
```

```
=====
```

```
void main()
```

```
{
```

```
// khai bao bien sodem  
int8 sodem;  
sodem = 0;  
  
lcd_init(); // KHOI TAO LCD  
set_tris_d(0x00);  
output_d(0x00);  
  
lcd_gotoxy(1,1);  
//printf(lcd_putc,"Counter:");// hien thi chu  
Counter tren LCD  
lcd_putc("Counter:");  
//-----  
while(TRUE)  
{  
    lcd_gotoxy(10,2); // cot 10 dong 2  
    printf(lcd_putc,"%02u",sodem);// PRINTF  
hien thi bien so  
    delay_ms(100);
```

```
sodem++;  
if(sodem == 100) break;  
}  
} //end main
```

7. Viết chương trình thực hiện:

- Nhấn SW1: hiển thị chữ “Xin chao cac ban” trên dòng 01 LCD tại $(x,y) = (1,1)$
- Nhấn SW2: hiển thị chữ “HUTECH” trên dòng 02 tại $(x,y) = (10,2)$

```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT,  
NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000)
```

```
#include <lcd.c> // khai bao thu vien lcd
```

```
#define sw1 pin_c3
```

```
#define sw2 pin_c4
```

// chương trình chính

void main()

{

lcd_init(); // khoi tao lcd

while(true) // or viet while(1)

{

if(!input(sw1)) // khi nhan phim 1

{

delay_ms(20); // chong doi phim sw1

if(!input(sw1))

{

lcd_gotoxy(1,1);

printf(lcd_putc, "Xin chao cac ban");

}

}

**if(!input(sw2)) // khi nhan phim 2 khong co
choi doi phim**

{

lcd_gotoxy(10,2);

```

        printf(lcd_putc, "Hutech");
    }

}

}

```

8. Viết chương trình thực hiện: hiển thị chữ “Set value:” trên dòng 1 tại (x,y) = (5,1) và hiển thị biến set_val (khởi động = 0) trên dòng 2 tại (x,y) = (8,2); và nếu:

- Nhấn SW1: nếu set_val < 200: set_val = set_val + 5
- Nhấn SW2: Nếu set_val > 0: set_val = set_val -1

Nội dung thay đổi của set_val được cập nhật tức thời trên dòng 2 LCD tại vị trí (x,y) = (8,2).

```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT,
NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000)
```

```
#include <lcd.c>
```

```
#define sw1 pin_c3
```

```
#define sw2 pin_c4
```

```
// chuong trinh chinh
```

```
void main()
```

```
{
```

```
    int8 set_val =0;
```

```
    lcd_init(); // khoi tao lcd
```

```
    lcd_gotoxy(5,1);
```

```
    printf(lcd_putc,"set value: ");
```

```
    while(1)
```

```
    {
```

```
        if((!input(sw1)) && (set_val<200)) // khi nhan  
sw1 va bien set_Val <200
```

```
    {
```

```
        set_val = set_val + 5;
```

```
        delay_ms(100);
```

```
    }
```



```

    if((!input(sw2)) && (set_val>0)) // khi nhan
sw2 va bien set_Val >0
    {
        set_val = set_val - 1;
        delay_ms(100);
    }
    lcd_gotoxy(8,2);
    printf(lcd_putc,"%03u",set_val);
}

}

```

9. Viết chương trình đồng hồ đếm giây: dòng 1 tại (1,1) hiển thị “DONG HO DEM GIAY”, dòng 2 tại (4,2) hiển thị “00:00”, và khi:

- Nhấn SW1: bắt đầu đếm “giây : phần trăm giây”
- Nhấn SW2: dừng đếm
- Nhấn giữ SW2 hơn 5 giây sẽ xóa số đếm về 0

```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT,  
NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000)
```

```
#include <lcd.c>
```

```
#define sw1 pin_c3
```

```
#define sw2 pin_c4
```

```
void main()
```

```
{
```

```
    int8 sw = 0; // khai bao gia tri cho sw
```

```
    int8 giay = 0;
```

```
    int8 pt_giay = 0;
```

```
    int8 dem = 0;
```

```
    lcd_init(); // khoi tao lcd
```

```
    lcd_gotoxy(1,1);
```

```
    printf(lcd_putc,"DONG HO DEM GIAY");
```

```
while(1)
{
    lcd_gotoxy(4,2);
    printf(lcd_putc,"%02u:%02u",giay, pt_giay);

    if(!input(sw1)) sw = 1; // khi nhan nut nhan 1
    thi sw co gia tri la 1
    while(!input(sw2))
    {
        sw = 2; // khi nhan nut nhan 2 thi sw co gia
        tri la 2
        delay_ms(100); // 50 * 100 = 5000ms = 5
        giay
        dem++;
        if(dem>50) //khi 5s
        {
            pt_giay = 0; giay = 0;
            lcd_gotoxy(4,2);
```

```
        printf(lcd_putc,"%02u:%02u",giay,  
pt_giay);  
    }  
}
```

```
switch(sw)  
{  
    case 1:  
    {  
        delay_ms(10);  
        pt_giay++;  
        if(pt_giay==100)  
        {  
            pt_giay=0; giay++;  
        }  
        lcd_gotoxy(4,2);  
        printf(lcd_putc,"%02u:%02u",giay,  
pt_giay);
```

```

    }

    case 2:

    {

        dem = 0;

        break;

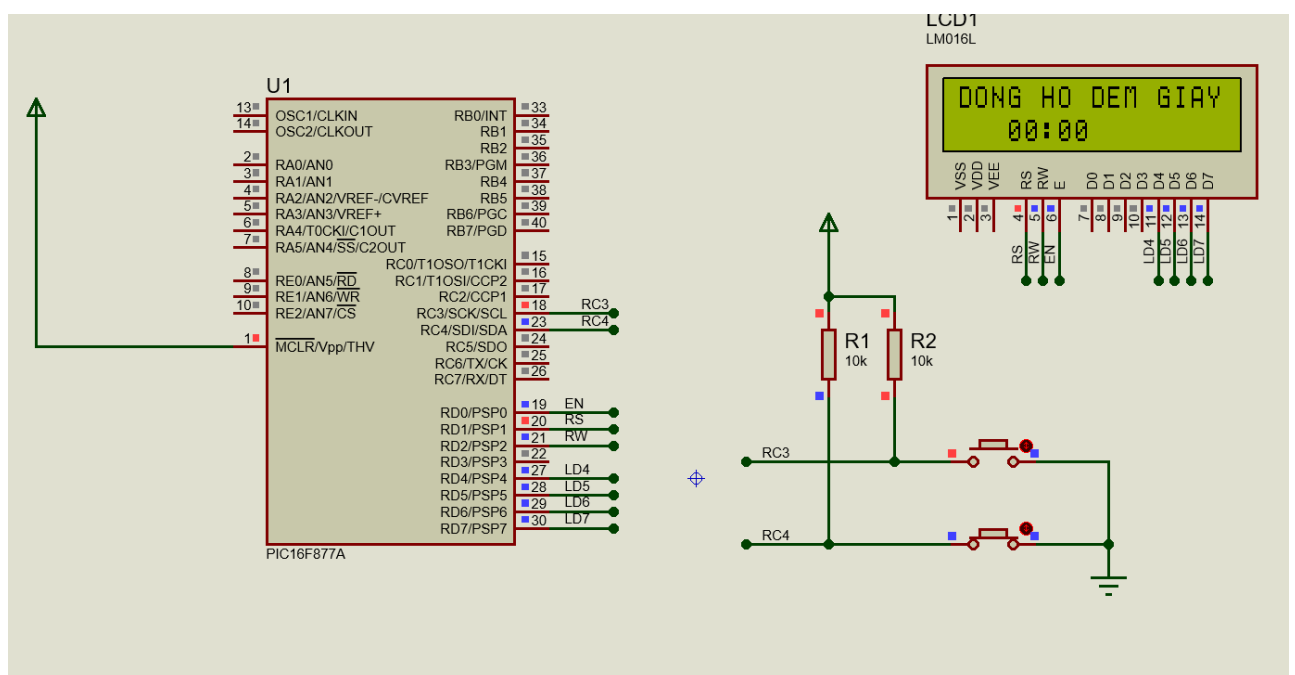
    }

}

}

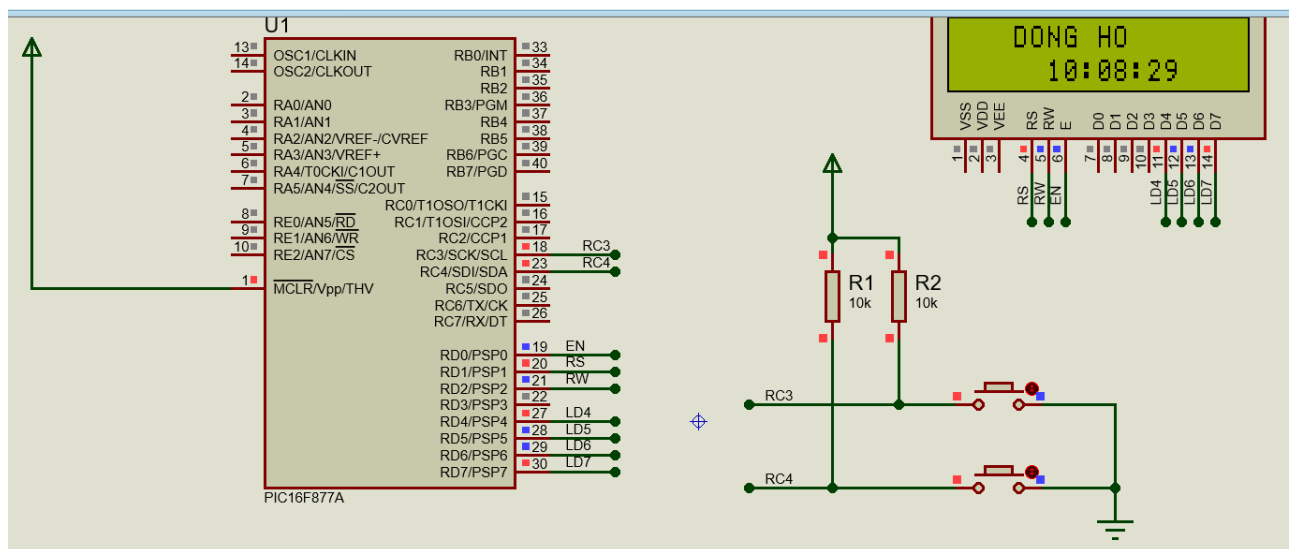
}

```



10. Viết chương trình thực hiện: hiển thị chữ “DONG HO” trên dòng 1 tại (4,1) và hiển thị “giờ:phút:giây” tại (6,2); thời gian bắt đầu là “00:00:00” và nếu:

- Nhấn SW1: cho phép chỉnh giờ (0-23)
- Nhấn SW2: cho phép chỉnh phút (0-59)



```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT,  
NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000)
```

```
#include <lcd.c>
```

```
#define sw1 pin_c3
```

```
#define sw2 pin_c4
```

```
void main()
{
    int8 gio,phut,giay;
    gio = 0; phut = 0; giay = 0;

    lcd_init();

    lcd_gotoxy(4,1);
    printf(lcd_putc,"DONG HO");

    lcd_gotoxy(6,2);

    printf(lcd_putc,"%02u:%02u:%02u",gio,phut,giay
);

    while(1)
    {
```

```
    delay_ms(100); giay ++;
    if (giay == 60)
    {
        giay = 0;
        phut++;
    }
    if (phut == 60)
    {
        phut = 0;
        gio++;
    }
    if (gio ==24)
    {
        gio = 0;
    }
    lcd_gotoxy(6,2);

    printf(lcd_putc,"%02u:%02u:%02u",gio,phut,giay
);
```



```
if(!input(sw1))
{
    gio++;
    gio = gio % 24;
}
if(!input(sw2))
{
    phut++;
    phut = phut % 60;
}
}
```

Ngày 27/03

Ví dụ 02: Viết chương trình delay 400ms dùng timer 1 với thạch anh 8 MHz.

Sử dụng:

Setup_timer1(T1_INTERNAL|T1_DIV_BY_2);

Bước 1: 8Mhz(Clock- tần số thạch anh đề cho) phải chia cho 4 - > $8/4 = 2\text{Mhz}$

Bước 2: Sau đó nhìn vô câu lệnh

Setup_timer1(T1_INTERNAL|T1_DIV_BY_2);

Timer 1 và có **bộ chia 2** cho trước (đề cho)

$$f_{\text{timer}} = 2\text{Mhz} / 2 = 1\text{ Mhz}$$

Bước 3:

Thời gian cho 1 xung (chu kỳ) $T_{\text{timer}} = 1/f_{\text{timer}} = 1/1\text{Mhz} = 1\mu\text{s}$

Bước 4: Viết chương trình delay 400 ms dùng timer 1

Thời gian delay là 400ms (đề bài cho)

Xác định số xung cần đếm = thời gian cần tạo (400ms) / T_{timer}

$$400\text{ms}/1\mu\text{s} = 400\,000 \text{ xung (} 40.000 \times 10 \text{)}$$

Timer 1 đếm tối đa ≤ 65.535 xung (16 bit) $0 \rightarrow 65.535$

Timer đếm nhiều lần (**40.000 xung x 10 lần**)

(Nháp $400\,000 / 65536 = 6,1$ lần)

Bước 5:

X.định giá trị bắt đầu của Timer = số tràn – số xung cần đếm

(Số tràn = số đếm lớn nhất + 1 = $65.535 + 1 = 65.536$)

$$\text{Timer bắt đầu đếm} = 65.536 - 40.000 = 25.536$$

```
void delay400ms()
```

```
{
```

```
    Setup_timer_1(T1_INTERNAL|T1_DIV_BY_2);
```

```
// de cho
```

```
    For (int8 i = 0;i < 8;i++) // 50.000 xung x 8 lan
```

```
{
```

```
    Set_timer1(15536); // bat dau dem tu xung
```

```
    While (get_timer1()>15536) ;
```

```
}
```

```
}
```

Ví dụ 02: Viết chương trình delay 1 s dùng timer 1 với thạch anh 4 MHz.

Sử dụng:

```
Setup_timer_1(T1_INTERNAL|T1_DIV_BY_4)
```

Bước 1: 4Mhz(Clock- tần số thạch anh đề cho) **phải chia cho 4 (luôn luôn là 4) - > $4/4 = 1\text{Mhz}$**

Bước 2: Sau đó nhìn vào câu lệnh

```
Setup_timer_1(T1_INTERNAL|T1_DIV_BY_4);
```

Timer 1 và có **bộ chia 4** cho trước (đề cho)

$f_{\text{timer}} = 1\text{Mhz} / 4 = 0.25 \text{ Mhz}$

Bước 3:

Thời gian cho 1 xung (chu kỳ) $T_{\text{timer}} = 1/f_{\text{timer}} = 1/0.25\text{Mhz} = 4\mu\text{s}$

Bước 4: Viết chương trình delay 1s dùng timer 1

Thời gian delay là 1s (đề bài cho)

Xác định số xung cần đếm = thời gian cần tạo (1s) / T_{timer}

$1\text{s}/1\mu\text{s} = 1\,000\,000$ xung (**62500 x 16**)

Timer 1 đếm tối đa ≤ 65.535 xung (16 bit) $0 \rightarrow 65.535$

Timer đếm nhiều lần (**62500xung x 16 lần**)

(Nháp $1000\,000 / 65536 = 15.26$ lần)

Bước 5:

X.định giá trị bắt đầu của Timer = số tràn – số xung cần đếm

(Số tràn = số đếm lớn nhất + 1 = $65.535 + 1 = 65.536$)

Timer bắt đầu đếm = $65.536 - 62.500 = 3036$

```
void delay1s()
```

```
{
```

```
    Setup_timer_1(T1_INTERNAL|T1_DIV_BY_4); //
    de cho
```

```
    For (int8 i = 0; i < 16; i++) // 62500 xung x 16 lan
```

```
{
```

```
    Set_timer1(3036); // bat dau dem tu xung
```

```
    While (get_timer1() > 3036) ;
```

```
}
}
```

Ví dụ 03: Viết chương trình delay x ms dùng timer 0 với thạch anh 8 MHz.

Sử dụng: `Setup_timer_0(RTCC_DIV_8 | RTCC_INTERNAL);`

Bước 1: 8Mhz(Clock- tần số thạch anh đề cho) **phải chia cho 4 (luôn luôn là 4) - > $8/4 = 2\text{Mhz}$**

Bước 2: Sau đó nhìn vào câu lệnh

`Setup_timer_0(RTCC_DIV_8 | RTCC_INTERNAL);`

Timer 0 và có **bộ chia 8 cho trước (đề cho)**

$$f_{\text{timer}} = 2\text{Mhz} / 8 = 0.25 \text{ Mhz}$$

Bước 3:

Thời gian cho 1 xung (chu kỳ) $T_{\text{timer}} = 1/f_{\text{timer}} = 1/0.25\text{Mhz} = 4\mu\text{s}$

Bước 4: Viết chương trình delay x ms dùng timer 0

Thời gian delay là x ms (đề bài cho)

Xác định số xung cần đếm = thời gian cần tạo (x ms) / T_{timer}

$x \text{ ms} / 1\mu\text{s} = 1000x \text{ xung (} 250x \times 4 \text{)}$

Timer 0 đếm tối đa ≤ 256 xung (8 bit) 0 -> 255

Timer đếm nhiều lần (**$250x \text{ xung} \times 4 \text{ lần}$**)

(Nháp $1000x / 256 = 3.9x \text{ lần}$)

Bước 5:

X.định giá trị bắt đầu của Timer = số tràn – số xung cần đếm
(Số tràn = số đếm lớn nhất + 1 = 255 + 1 = 256)

Timer bắt đầu đếm = 256 – 250x = x

```
#include <16F877A.h>
```

```
#fuses NOWDT, HS, NOPUT, NOPROTECT,  
NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=4M) // tần số thạch anh là 4  
Mhz
```

```
// chương trình con delay 400ms
```

```
void delay_xms()
```

```
{
```

```
    Setup_timer_0(RTCC_DIV_8 | RTCC_INTERNAL);  
    // để cho
```

```
    For (int8 i = 0; i < 4; i++) // 250x xung x 4 lần
```

```
{
```

```
    Set_timer1(x); // bắt đầu đếm từ xung
```

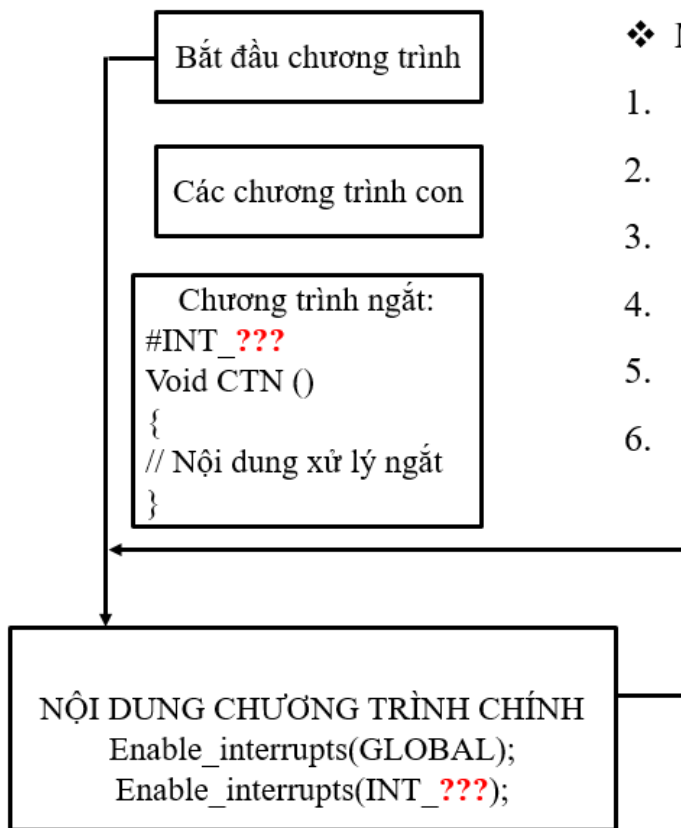
```
    While (get_timer1() > x);
```

```
}
```

```
}
```

```
void main()
{
    While (1)
    {
        Output_c(0x01);
        delay_xms(); // chuong trinh delay tu viet
        Output_c(0x00);
        delay_xms();
    }
}
```

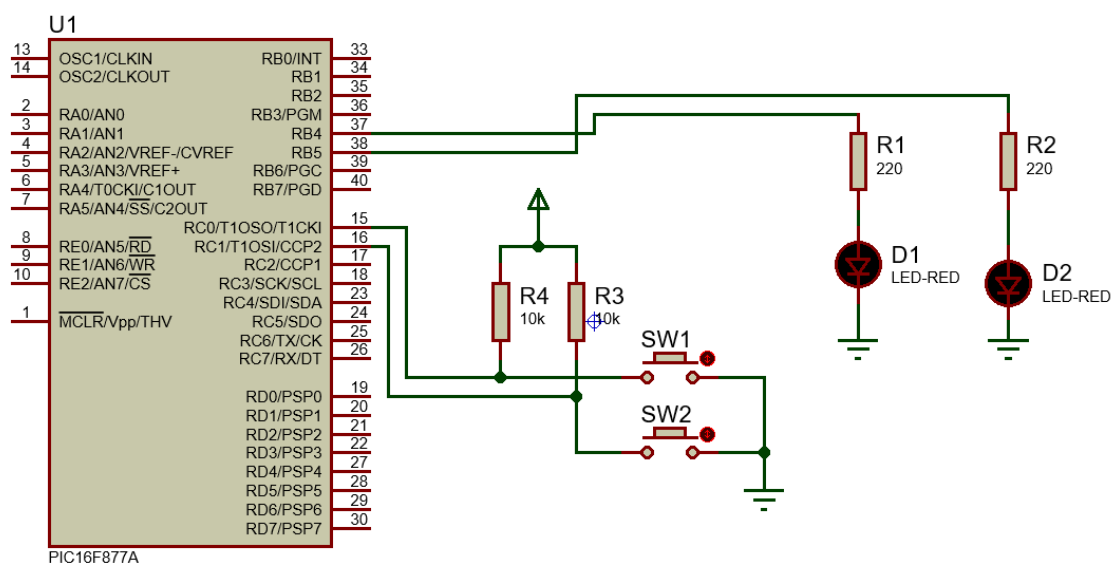
❖ Hoạt động của một chương trình ứng dụng có sử dụng sự kiện ngắt



❖ Một số ngắt sử dụng:

1. #INT_TIMER0 (tràn timer 0)
2. #INT_RTCC (tràn timer 0)
3. #INT_TIMER1 (tràn time 1)
4. #INT_TIMER2 (tràn timer 2)
5. #INT_EXT (mức thấp tại RB0)
6. #INT_RDA (nhận một byte)

134



PIC16F877A

Dùng ngắt timer 0 kiểm tra 2 phím nhấn, sử dụng:

```
Setup_timer_0(RTCC_DIV_4| RTCC_INTERNAL)
```

1. Tính chu kì quét phím của timer 0
2. Nếu phím 1 nhấn: sáng D1; phím 2 nhấn: tắt D1

Giải:

Tần số thạch anh 8Mhz (đề cho)

Bước 1: 8Mhz(Clock- tần số thạch anh đề cho) **phải chia cho 4 (luôn luôn là 4) - > $8/4 = 2\text{Mhz}$**

Bước 2: Sau đó nhìn vào câu lệnh

```
Setup_timer_0(RTCC_DIV_4| RTCC_INTERNAL);
```

Timer 0 và có **bộ chia 4 cho trước (đề cho)**

$$f_{\text{timer}} = 2\text{Mhz} / 4 = 0.5 \text{ Mhz}$$

Bước 3:

Thời gian cho 1 xung (chu kỳ) $T_{\text{timer}} = 1/f_{\text{timer}} = 1/0.5\text{Mhz} = 2\mu\text{s}$

Chu kỳ Timer 0 là $2\mu\text{s}$

(Số tràn của Timer 0 = số đếm lớn nhất + 1 = $255 + 1 = 256$)

chu kì quét phím của timer 0: $256 * 2 = 512 \mu\text{s}$

```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT,  
NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000)
```

```
#define D1  PIN_B4
```

```
#define SW1  PIN_C0
```

```
#define SW2  PIN_C1
```

```
// Chương trình ngat
```

```
#INT_RTCC
```

```
void ktra_sw()
```

```
{
```

```
    if (!input(SW1)) output_high(D1);
```

```
    if (!input(SW2)) output_low(D1);
```

```
}
```

```
//-----CHUONG TRÌNH CHÍNH-----
```

```
Void main()
```

```
{
```

```
enable_interrupts(GLOBAL);// cho phép ngắt
toan cuc
```

```
enable_interrupts(INT_RTCC);// cho phép ngắt
timer 0
```

```
setup_timer_0(RTCC_DIV_4|RTCC_INTERNAL);//
cau hình timer 0
```

```
while(TRUE);
}
```

Bài 2: led 7 đoạn

Dùng ngắt timer 1 để quét LED, thời gian delay mỗi led 3 ms. Sử dụng:

```
Setup_timer_1(T1_INTERNAL| T1_DIV_BY_1)
```

1. Tính chu kì quét LED của timer 1
2. Hiển thị chữ “HELP” trên 4 LED

Giải

Tần số thạch anh 8Mhz (đề cho)

Bước 1: 8Mhz(Clock- tần số thạch anh đề cho)
phải chia cho 4 (luôn luôn là 4) - > $8/4 = 2\text{Mhz}$

Bước 2: Sau đó nhìn vào câu lệnh

Setup_timer_1(T1_INTERNAL| T1_DIV_BY_1);

Timer 0 và có bộ chia 1 cho trước (để cho)

$$f_{\text{timer}} = 2\text{Mhz} / 1 = 2 \text{ Mhz}$$

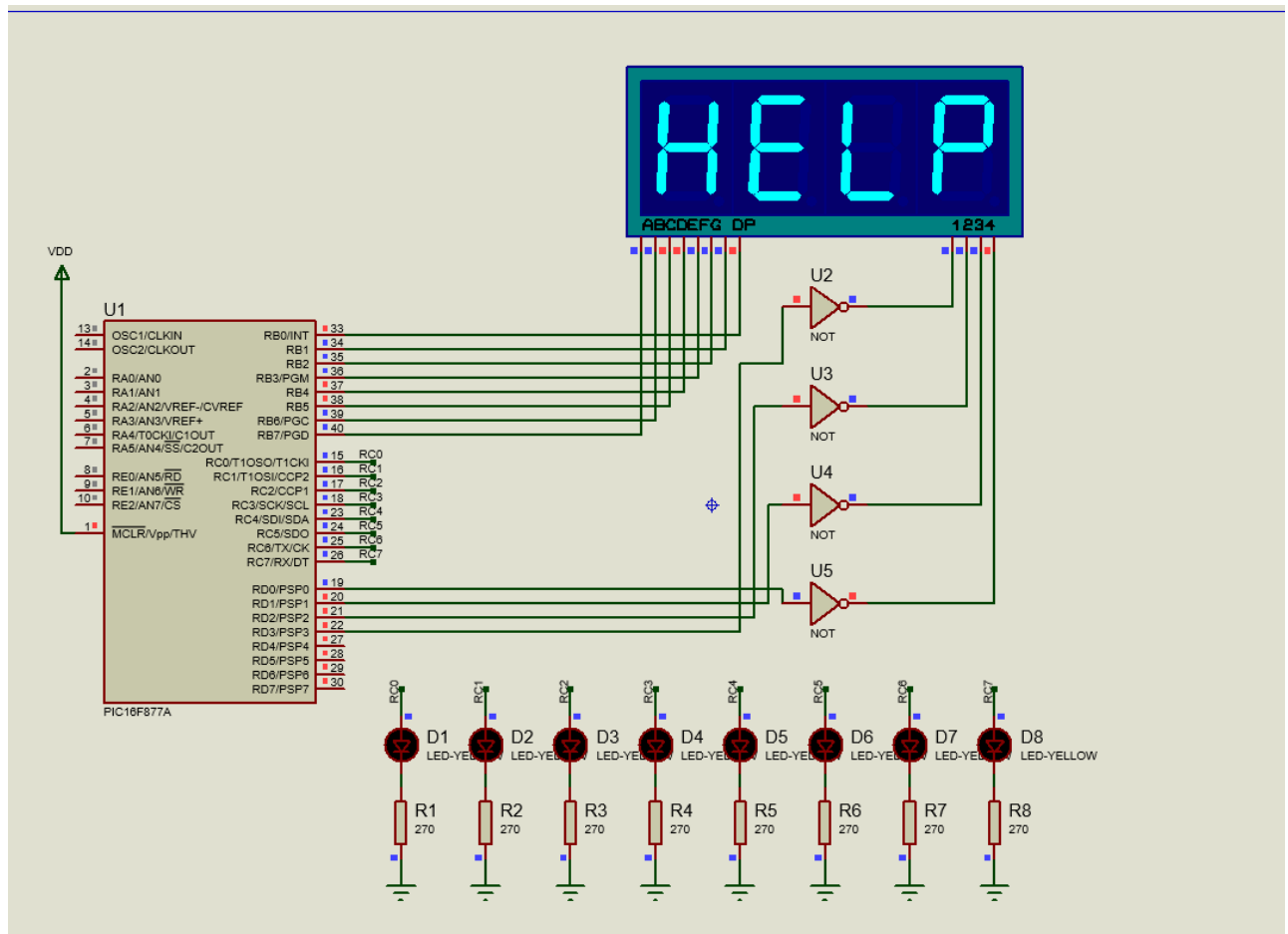
Bước 3:

Thời gian cho 1 xung (chu kỳ) $T_{\text{timer}} = 1/f_{\text{timer}} = 1/2\text{Mhz} = 0.5\mu\text{s}$

Chu kỳ Timer 1 là 0.5us

(Số tràn của Timer 1 = số đếm lớn nhất + 1 = 65535 + 1 = 65536)

**Chu kì quét 4 LED là: Số xung tràn * T = ~33 mS
: 65536 * 0.5= 32768 μS = 32,768 mS \approx 33ms**



// Chu kì quét 4 LED là: ~33 mS

```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT,  
NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000)
```

```
CONST int8 LED[4] = {0x91,0x61,0xe3,0x31};  
// bang ma cua chu HELP
```

```
CONST int8 DK[4] =  
{0x07,0x0b,0x0d,0x0e};//ma dieu khien 4 led 7  
doan anode chung
```

```
Int8 i=0;
```

```
//           L1 L2 L3 L4
```

```
//      RD7 RD6 RD5 RD4 RD3 RD2 RD1 RD0
```

```
// 0x07 = 0  0  0  0  0  1  1  1
```

```
// 0x0b = 0  0  0  0  1  0  1  1
```

```
// 0x0d = 0  0  0  0  1  1  0  1
```

```
// 0x0e = 0  0  0  0  1  1  1  0
```

```
#INT_TIMER1
```

```
void qled()
```

```
{
```

```
    Output_d(0xff);// tat các led
```

```
    Output_b(LED[i]); // ma chu HELP
```

```
    output_d(DK[i]);
```

```
    i++; i = i%4;
```

```
}
```

```
//-----CHUONG TRÌNH CHÍNH-----
```

Void main()

{

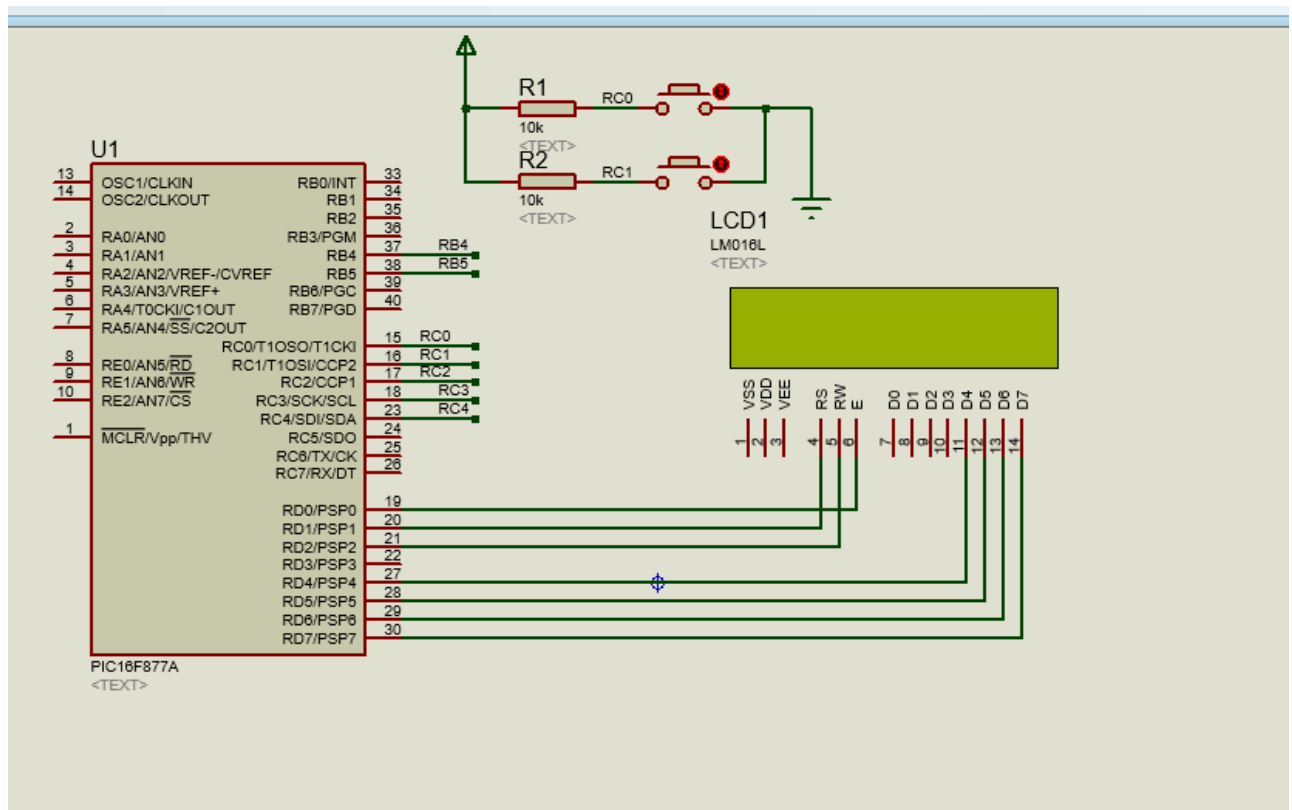
enable_interrupts(GLOBAL);

enable_interrupts(INT_TIMER1);

setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);

while(TRUE);

}



1. Dùng ngắt timer 1 để kiểm tra phím. Sử dụng:

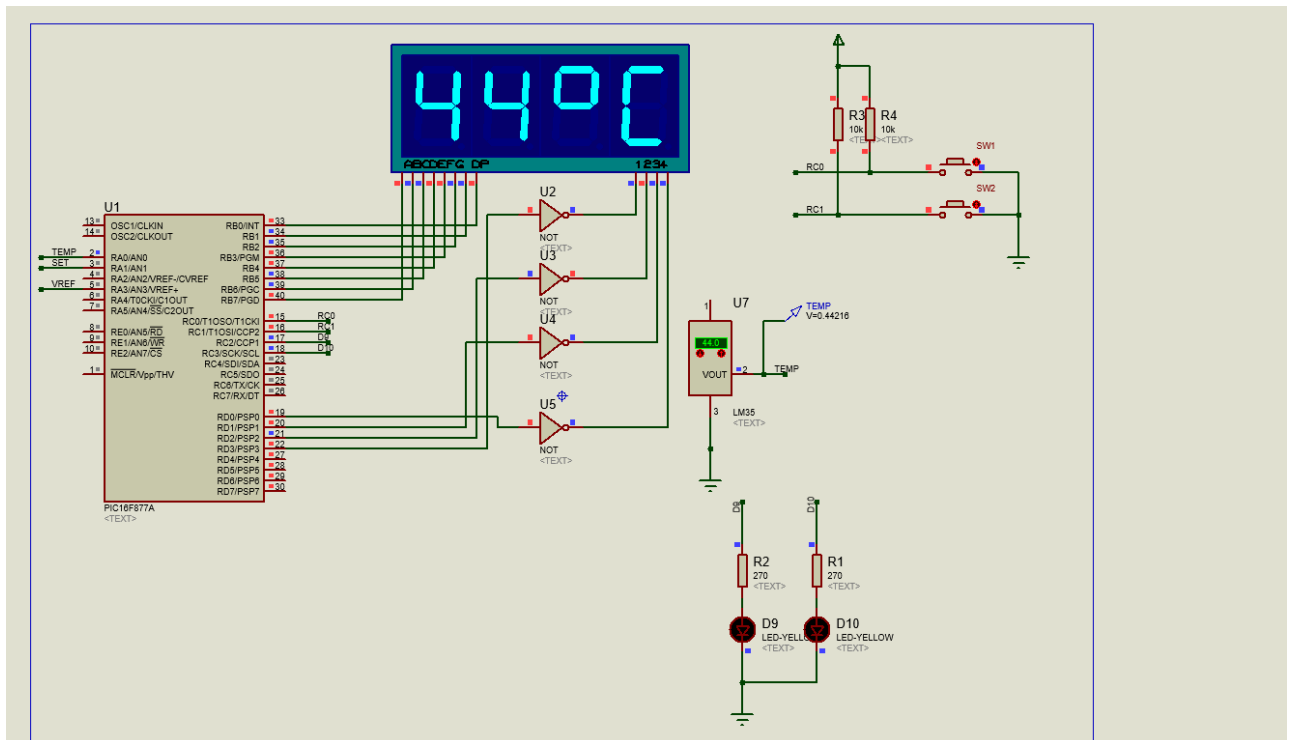
**Setup_timer_1(T1_INTERNAL|
T1_DIV_BY_1)**

Viết chương trình thực hiện: hiển thị chữ “Set value:” trên dòng 1 tại (x,y) = (5,1) và hiển thị biến set_val (khởi động = 0) trên dòng 2 tại (x,y) = (8,2); và nếu:

- Nhấn SW1: nếu set_val < 200: set_val = set_val + 5**
- Nhấn SW2: Nếu set_val > 0: set_val = set_val -1**

Nội dung thay đổi của set_val được cập nhật tức thời trên dòng 2 LCD tại vị trí (x,y) = (8,2). Tính chu kì ngắt?

ADC



Theo sơ đồ: Led 7 đoạn anode chung, 0 sáng , 1 tắt

A -> RB7

B -> RB6

C -> RB5

D -> RB4

E -> RB3

F -> RB2

G -> RB1

DP ->RB0

Output_b(maled)

RB7 RB6 RB5 RB4 RB3 RB2 RB1 RB0

A B C D E F G DP

Chân điều khiển

RD3 -> LED 1

RD2 -> LED 2

RD1 -> LED 3

RD0 -> LED 4

Output_d(maleddk)

RD7 RD6 RD5 RD4 RD3 RD2 RD1 RD0

1	1	1	1		0	1	1	1	led1 0xf7
1	1	1	1		1	0	1	1	led2 0xfb
1	1	1	1		1	1	0	1	led3 0xfd
1	1	1	1		1	1	1	0	led4 0xfe

```

#include <16F877A.h>

#FUSES NOWDT, HS, NOPUT, PROTECT, NODEBUG, NOBROWNOUT, NOLVP

#use delay(clock=8000000) // Tần số thạch anh dùng trong hệ thống

CONST unsigned char LED7S[4] = {"H"7s, "E"7s, "L"7s, "P"7s };
CONST unsigned char dk[4] = {0xf7, 0xfb, 0xfd, 0xfe};

Void quet_LED()
{ for (int8 i = 0; i < 4; i++)
    {output_d(0xff); output_b(LED7S[i]); output_d(dk[i]); delay_ms(5);}
}

//-----CHƯƠNG TRÌNH CHÍNH-----

Void main()
{
    Output_d(0xff); // tắt các led
    While (TRUE){ quet_LED();}
}

```

Chương trình quét led 7 đoạn không dùng ngắt

YÊU CẦU (sử dụng ADC 8 bit):

Ví dụ 1:

Đọc giá trị nhiệt độ từ bộ ADC và hiển thị trên led 7 đoạn dạng xx°C, sử dụng điện áp chuẩn là VDD (5V)

$$1 \text{ biến ADC} = \frac{V_{ref}=V_{dd}}{2^n-1} = \frac{5V}{2^8-1} = 20mV$$

Mà 10mV/ độ C

N = read_adc();

Temp=N*2;

temp = read_adc()*2;

#include <16F877A.h>

#device ADC=8 // yeu cau de bai la ADC 8 BIT

#FUSES NOWDT, HS, NOPUT, PROTECT,
NODEBUG, NOBROWNOUT, NOLVP

#use delay(clock=8M) // TAN SO THACH ANH

#define D1 PIN_C2

#define D2 PIN_C3

// LED ANDDE CHUNG

```
CONST int8 a[10] = {0x03,  
0x9f,0x25,0x0d,0x99,0x49,0x41,0x1f,0x01,0x09};  
int8 chuc,dvi;  
int16 temp;  
// Chuong trinh delay ms  
void dl_ms(int16 gtri)  
{  
    for (int16 i = 0; i < gtri; i++) delay_us(1000);  
}  
// Chuong trình ngat timer 0  
#INT_TIMER0  
void qled()  
{  
    Output_d(0xff);// t?t các led  
    Output_b(a[chuc]); output_low(PIN_D3);  
    delay_ms(3);// "chuc: nhiet do  
    // output_low(PIN_D3) = output_d(0xf7)  
    Output_d(0xff);// t?t các led
```

```

    Output_b(a[dvi]); output_low(PIN_D2);
delay_ms(3);// "Don vi: nhiet do

    // output_low(PIN_D2) = output_d(0xfb)
    Output_d(0xff);// t?t các led

    Output_b(0x39); output_low(PIN_D1);
delay_ms(3);// "dau: o

    // output_low(PIN_D1) = output_d(0xfd)
    Output_d(0xff);// t?t các led

    Output_b(0x63); output_low(PIN_D0);
delay_ms(3);// "C

    // output_low(PIN_D0) = output_d(0xfe)
}

//-----CHUONG TRÌNH CHÍNH-----

Void main()
{
    // cho phép ngat timer 0
    enable_interrupts(GLOBAL);
    enable_interrupts(INT_TIMER0);

```

// không có gọi chương trình con qled() khi sử dụng ngắt

```
Setup_timer_0(RTCC_DIV_4|RTCC_INTERNAL);
```

```
// Cấu hình ADC
```

```
setup_adc(ADC_CLOCK_INTERNAL);
```

```
setup_adc_ports(AN0_AN1_AN3); // Vref =  
VDD = 5V
```

```
// nếu mà bí setup_adc_ports(ALL_ANALOG);
```

```
set_adc_channel(0);
```

```
while(1)
```

```
{
```

```
temp = read_adc()*2;
```

```
//-----
```

```
chuc = temp/10;
```

```
dvi = temp %10;
```

```
// 43 / 10 = 4 hàng chuc
```

```
// 43 % 10 = 3 hàng đơn vị
```

```
}
```



```
while(true);///  
}
// STOP
```

Đọc giá trị nhiệt độ từ bộ ADC (8bit) và hiển thị trên led 7 đoạn dạng xx°C, sử dụng điện áp chuẩn là VDD (2.5V)

$$\text{1 biến ADC} = \frac{V_{ref}=V_{dd}}{2^n-1} = \frac{2.5V}{2^8-1} = 10mV$$

Mà 10mV/ độ C (LM35)

N = read_adc();

Temp=N;

- **Đọc giá trị nhiệt độ từ bộ ADC (8 bit) và hiển thị trên led 7 đoạn dạng xx°C, sử dụng điện áp chuẩn là VDD (5V)**
- **Nhiệt độ > 20 độ C: D9 sáng, D10 tắt**
- **Nhiệt độ > 30 độ C: D9 tắt, D10 sáng**
- **Nhiệt độ > 35 độ C: D9 và D10 chớp tắt với delay 100ms**
- **1 biến ADC = $\frac{V_{ref}=V_{dd}}{2^n-1} = \frac{5V}{2^8-1} = 20mV$**
- **Mà 10mV/ độ C**

- **N = read_adc();**
- **Temp=N*2;**

```
#include <16F877A.h>
```

```
#device ADC=8
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT,  
NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8M)
```

```
#define D9 PIN_C2
```

```
#define D10 PIN_C3
```

```
CONST int8 a[10] = {0x03,  
0x9f,0x25,0x0d,0x99,0x49,0x41,0x1f,0x01,0x  
09 };
```

```
int8 chuc,dvi;
```

```
int16 temp;
```

```
// Chuong trinh delay ms
```

```
void dl_ms( int16 gtri)
```

```
{
```

```
    for (int16 i = 0; i < gtri; i++) delay_us(1000);
```

```
}
```

```

// Chương trình ng?t timer 0

#include <TIMER0>

void qled()
{
    Output_d(0xff); // t?t các led

    Output_b(a[chuc]); output_low(PIN_D3);
    delay_ms(3); // "chuc: nhiet do

    Output_d(0xff); // t?t các led

    Output_b(a[dvi]); output_low(PIN_D2);
    delay_ms(3); // "Don vi: nhiet do

    Output_d(0xff); // t?t các led

    Output_b(0x39); output_low(PIN_D1);
    delay_ms(3); // "dau: o

    Output_d(0xff); // t?t các led

    Output_b(0x63); output_low(PIN_D0);
    delay_ms(3); // "C
}

//-----CHUONG TRÌNH CHÍNH-----

Void main()

```

```
{  
    // cho phep ngat timer 0  
    enable_interrupts(GLOBAL);  
    enable_interrupts(INT_TIMER0);  
  
    Setup_timer_0(RTCC_DIV_4|RTCC_INTERNAL);  
  
    // Cau hinh ADC  
    setup_adc(ADC_CLOCK_INTERNAL);  
    setup_adc_ports(AN0_AN1_AN3);  
    set_adc_channel(0);  
    while(1)  
    {  
        temp = read_adc()*2;//  
        //-----  
        chuc = temp/10;  
        dvi = temp %10;  
        if(temp<=20)  
        {
```

```
    output_low(d9);
    output_low(d10);
}
if((temp>20) && (temp<=30))
{
    output_high(d9);
    output_low(d10);
}
if((temp>30) && (temp<=35))
{
    output_low(d9);
    output_high(d10);
}
if(temp>35)
{
    output_toggle(d9);
    output_toggle(d10);
    dl_ms(100);
}
```

```
}
```

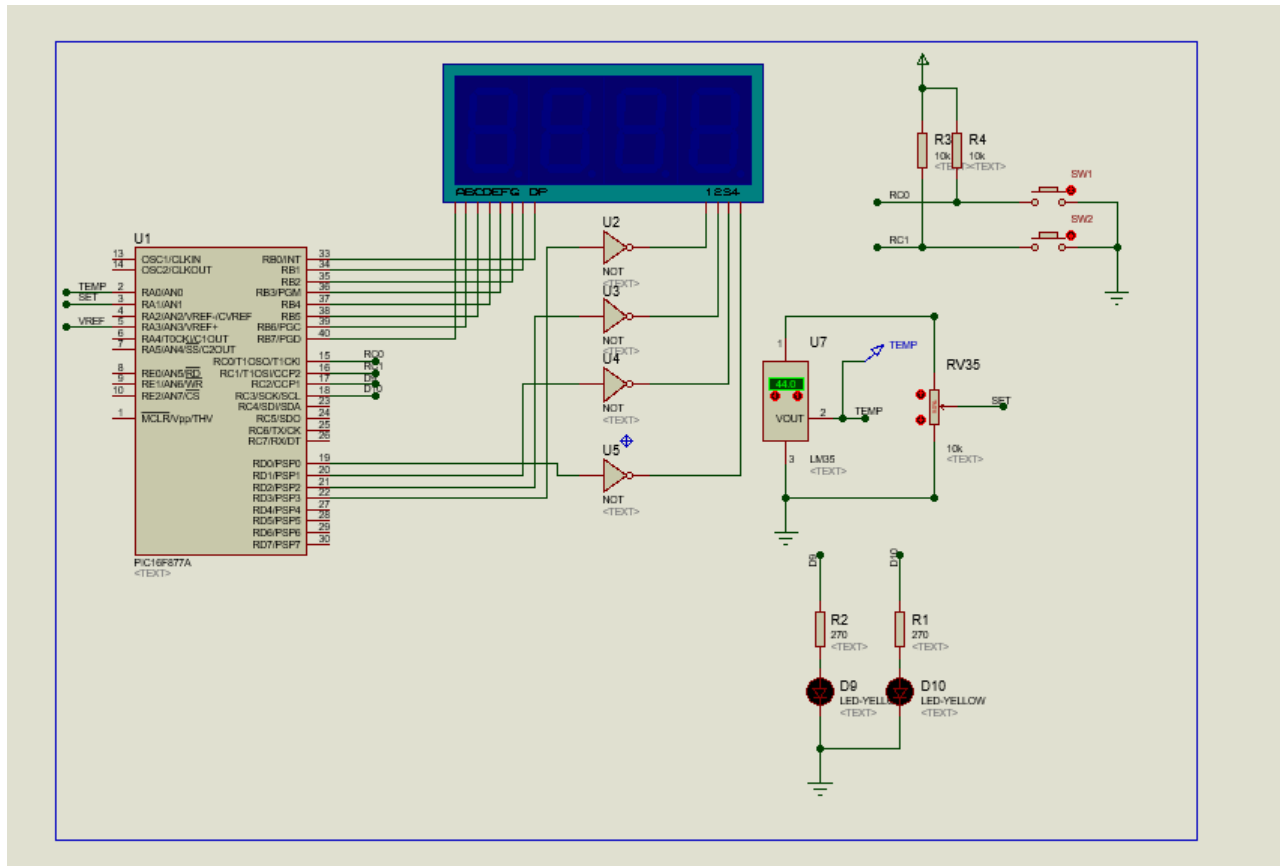
```
while(true);/// STOP
```

```
}
```

YÊU CẦU (sử dụng ADC 8 bit):

Sử dụng biến trở R35 để đặt nhiệt độ đặt (set_temp, mặc định là 20°C) và hiển thị giá trị trên 4 led theo dạng “St.XX”, sử dụng điện áp chuẩn là VDD. Nếu:

- Nhiệt độ đo từ LM 35 < nhiệt độ đặt: D9 sáng, D10 tắt
- Nhiệt độ đo từ LM 35 > nhiệt độ đặt: D10 sáng, D9 tắt



$$1 \text{ biến ADC} = \frac{V_{ref}=V_{dd}}{2^n-1} = \frac{5V}{2^8-1} = 20mV$$

Mà 10mV/ độ C

$N = \text{read_adc}();$

$\text{Temp} = N * 2;$

$\text{temp} = \text{read_adc}() * 2;$

`#include <16F877A.h>`

```
#device ADC=8// ADC = 10

#FUSES NOWDT, HS, NOPUT, PROTECT,
NODEBUG, NOBROWNOUT, NOLVP

#use delay(clock=8000000)

#define D9 PIN_C2

#define D10 PIN_C3

CONST int8 a[10] = {0x03,
0x9f,0x25,0x0d,0x99,0x49,0x41,0x1f,0x01,0x
09 };

int8 chuc,dvi;

int16 temp;

// Chuong trinh delay ms

//void dl_ms( int16 gtri)

//{

//  for (int16 i = 0; i < gtri; i++)
//    delay_us(1000);

//}

// Chuong trình ng?t timer 1

#INT_TIMER0
```



```

void qlcd()
{
    Output_d(0xff);// t?t các led
    Output_b(0x49); output_low(PIN_D3);
    delay_ms(3);// "chuc: nhiet do
    Output_d(0xff);// t?t các led
    Output_b(0xe0); output_low(PIN_D2);
    delay_ms(3);// "Don vi: nhiet do
    Output_d(0xff);// t?t các led
    Output_b(a[chuc]); output_low(PIN_D1);
    delay_ms(3);// "dau: o
    Output_d(0xff);// t?t các led
    Output_b(a[dvi]); output_low(PIN_D0);
    delay_ms(3);// "C
}

//-----CHUONG TRÌNH CHÍNH-----

Void main()
{
    // cho phép ngat timer 0

```

```
enable_interrupts(GLOBAL);  
enable_interrupts(INT_TIMER0);
```

```
Setup_timer_0(RTCC_DIV_4|RTCC_INTERNAL);
```

```
// Cau hinh ADC
```

```
setup_adc(ADC_CLOCK_INTERNAL);
```

```
setup_adc_ports(AN0_AN1_AN3);
```

```
set_adc_channel(0);
```

```
while(1)
```

```
{
```

```
temp = read_adc()*2;
```

```
//-----
```

```
chuc = temp/10;
```

```
dvi = temp %10;
```

```
if(temp<=20)
```

```
{
```

```
output_high(D9);
```

```
output_low(D10);
```

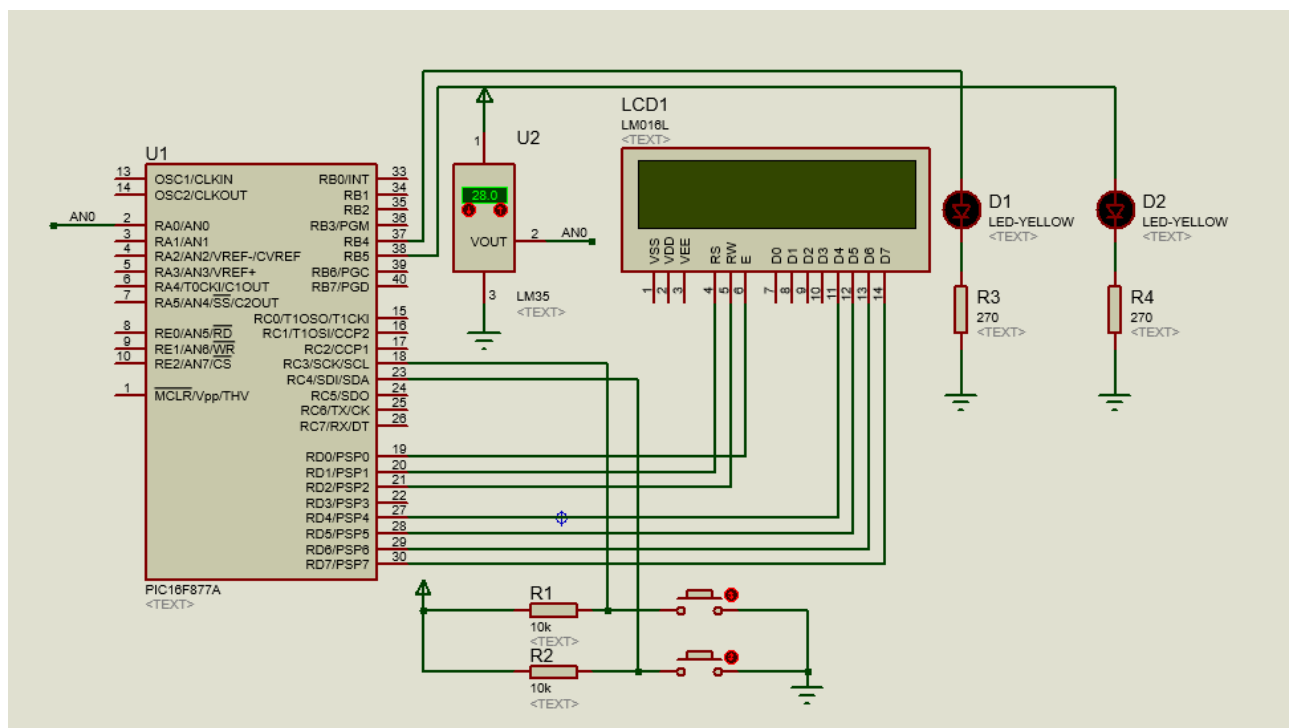
```
    }  
    if(temp>20)  
    {  
        output_high(D10);  
        output_low(D9);  
    }  
    if(temp==20)  
    {  
        output_low(D10);  
        output_low(D9);  
    }  
  
    }  
  
while(true);/// STOP  
}
```

Cho mạch text LCD. Đọc giá trị nhiệt độ từ bộ ADC và hiển thị trên LCD, dòng 1: “Nhiệt do:”, dòng 2: xx°C sử dụng điện áp chuẩn là VDD (5V).

Nhiệt độ >20 độ C: D1 sáng, D2 tắt

Nhiệt độ >30 độ C: D1 tắt, D2 sáng

Nhiệt độ >35 độ C: D1 và D2 chớp tắt với delay 100ms



```
#include <16F877A.h>
```

```
#device adc=8
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT,  
NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000)
```

```
#include <lcd.c>

#define LED_D1 pin_b4
#define LED_D2 pin_b5

//=====

=====

void main()
{
    int8 temp;

    //=====

    =====

    Setup_adc(ADC_CLOCK_INTERNAL);
    Setup_adc_ports(AN0_AN1_AN3);
    Set_adc_channel(0);
    //-----

    lcd_init(); // khoi tao LCD
    temp = 0;

    lcd_gotoxy(4,1); // hien thi dong 1 cot 4
    printf(lcd_putc,"NHIET DO:");
```

```
//-----  
while(TRUE)  
{  
    temp = read_adc()*2;  
    lcd_gotoxy(8,2); // dong 2 cot 8  
    printf(lcd_putc,"%02u",temp); // do bien  
temp la int8  
    lcd_putc(223); // hien thi o  
    lcd_putc("C");  
  
    if(temp>=35)  
    {  
        // output_toggle(D1)  
        // delay_ms(100);  
        // output_toggle(D2)  
        // delay_ms(100);  
        output_high(LED_D1);  
        output_high(LED_D2);  
        delay_ms(100);
```

```
    output_low(LED_D1);  
    output_low(LED_D2);  
    delay_ms(100);  
  
}  
  
else if(temp>=30)  
{  
    output_high(LED_D2);  
    output_low(LED_D1);  
}  
  
else if(temp>=20)  
{  
    output_low(LED_D2);  
    output_high(LED_D1);  
}  
  
else
```

```

{
    output_low(LED_D2);
}

}

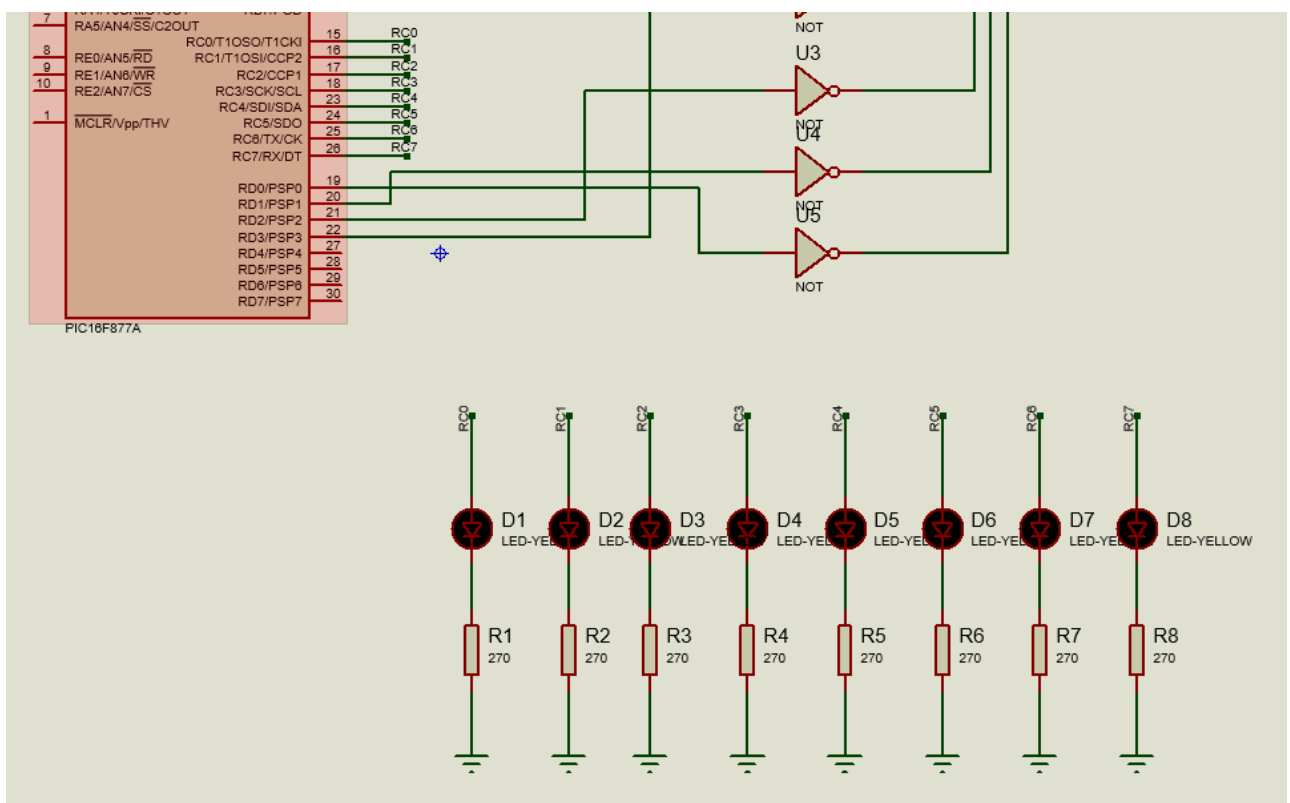
```

```

} //end main

```

Bài 1.7 (không dùng ngắt) Sửa chương trình để Led sáng dần từ D8 – D1, thời gian delay 300ms, lặp lại 30 lần, sau đó tắt dần từ D8 – D1, lặp lại 257 lần. Lặp lại toàn bộ quá trình 40 lần.



output_c()		SÁNG DẪN										
Theo đề bài		D8	D7	D6	D5	D4	D3	D2	D1			
for	stt	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0			
i	0	0	0	0	0	0	0	0	0	0X00		
i	1	1	0	0	0	0	0	0	0	0X80		
i	2	1	1	0	0	0	0	0	0	0XC0		
i	3	1	1	1	0	0	0	0	0	0XE0		
i	4	1	1	1	1	0	0	0	0	0XF0		
i	5	1	1	1	1	1	0	0	0	0XF8		
i	6	1	1	1	1	1	1	0	0	0XFC		
i	7	1	1	1	1	1	1	1	0	0XFE		
i	8	1	1	1	1	1	1	1	1	0XFF		

		TẮT DẪN										
		D8	D7	D6	D5	D4	D3	D2	D1			
stt		RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0			
		1	1	1	1	1	1	1	1	0XFF		
		0	1	1	1	1	1	1	1	0X7F		
		0	0	1	1	1	1	1	1	0X3F		
		0	0	0	1	1	1	1	1	0X1F		
		0	0	0	0	1	1	1	1	0X0F		
		0	0	0	0	0	1	1	1	0X07		
		0	0	0	0	0	0	1	1	0X03		
		0	0	0	0	0	0	0	1	0X01		

```
#include <16F877A.h>
```

```
#FUSES NOWDT, HS, NOPUT, PROTECT,  
NODEBUG, NOBROWNOUT, NOLVP
```

```
#use delay(clock=8000000)
```

```
CONST unsigned char ledsangdan[9] = {0X00,  
0X80, 0XC0, 0XE0, 0XF0, 0XF8, 0XFC, 0XFE,  
0XFF};
```

```
CONST unsigned char ledtatdan[9] =  
{0XFF,0X7F,0X3F,0X1F,0X0F,0X07, 0X03,  
0X01};
```

```
void main()
```

```
{
```

```
    // chay 40 lan toan bo qua trinh
```

```
    for(int8 y=0;y<40;y++)
```

```
    {
```

```
        // chay 30 lan
```

```
        for(int8 h=0;h<30;h++)
```

```
        {
```

```
            // quet ma led
```

```
            for(int8 i=0;i<9;i++)
```

```
            {
```

```
                output_c(ledsangdan[i]);
```

```
                delay_ms(300);
```

```
            }
```

```
        }
```

```

// chạy 257 lần
for(int16 k=0;k<257;k++)
{
    // quét ma led
    for(int8 i=0;i<9;i++)
    {
        output_c(ledtatdan[i]);
        delay_ms(300);
    }
}
}
}

```

Viết chương trình 1 led đơn sáng tắt với thời gian delay là 400ms dùng timer 1 với cấu hình như sau:

```

Setup_timer_1(T1_INTERNAL|T1_DIV_BY_2)

```

```

#include <16F877A.h>

```

```

#fuses NOWDT, HS, NOPUT, NOPROTECT, NODEBUG, NOBROWNOUT, NOLVP

```

```
#use delay(clock=8M)
```

```
void delay400ms()
```

```
{
```

```
Setup_timer_1(T1_INTERNAL|T1_DIV_BY_2); //1us/xung
```

```
For (int8 i = 0;i < 10;i++) // c?n 400 000 xung  
= 10*40 000
```

```
{
```

```
Set_timer1(25536);
```

```
While (get_timer1()>25536) ;
```

```
}
```

```
}
```

```
// chuong trinh chinh
```

```
void main (void)
```

```
{
```

```
While (1)
```

```
{
```

```
Output_c(0x01);
```

```

    delay400ms();
    Output_c(0x00);
    delay400ms();
}
}

```

Viết chương trình hiển thị chữ HELP trên led 7 đoạn không dùng ngắt

```

#include <16F877A.h>

#FUSES NOWDT, HS, NOPUT, PROTECT,
NODEBUG, NOBROWNOUT, NOLVP

#use delay(clock=8000000)

//-----
---

Void main()
{
While (1){
    Output_d(0xff);// t?t các led
    Output_b(0x91); output_low(PIN_D3);
    delay_ms(3);// "H"
}
}

```

```

    Output_d(0xff);// t?t các led
    Output_b(0x61); output_low(PIN_D2);
    delay_ms(3); // "E"

    Output_d(0xff);// t?t các led
    Output_b(0xe3); output_low(PIN_D1);
    delay_ms(3);// "L"

    Output_d(0xff);// t?t các led
    Output_b(0x31); output_low(PIN_D0);
    delay_ms(3);// "P"
}
}

```

Viết chương trình hiển thị chữ HELP trên led 7 đoạn dùng ngắt TIMER1 với cấu hình
setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);

Led 7 đoạn thì viết chương trình con quét led
#include <16F877A.h>
#FUSES NOWDT, HS, NOPUT, PROTECT,
NODEBUG, NOBROWNOUT, NOLVP

```

#use delay(clock=8000000)
// Chương trình ng?t timer 1
#INT_TIMER1
void qled()
{
    Output_d(0xff);// t?t các led
    Output_b(0x91); output_low(PIN_D3);
delay_ms(3);// "H"
    Output_d(0xff);// t?t các led
    Output_b(0x61); output_low(PIN_D2);
delay_ms(3); // "E"
    Output_d(0xff);// t?t các led
    Output_b(0xe3); output_low(PIN_D1);
delay_ms(3);// "L"
    Output_d(0xff);// t?t các led
    Output_b(0x31); output_low(PIN_D0);
delay_ms(3);// "P"
}
//-----CHƯƠNG TRÌNH CHÍNH-----
Void main()

```

```

{
    enable_interrupts(GLOBAL);
    enable_interrupts(INT_TIMER1);

    setup_timer_1(T1_INTERNAL/T1_DIV_BY_
1);
    while(TRUE);
}

```

Timer là bộ định thời: **tạo thời gian chính xác.**

→ Vào: $f \rightarrow T$ (s)

→ Ra: Số xung đếm được $\times T$ = thời gian cụ thể.

→ $f_{\text{TIMER}} = f_{\text{TA}} / 4$.

*** TIMER:**

1. Xác định chu kỳ T của ngõ vào xung của Timer.

- $T = 1 / ((f_{\text{TA}}/4) / \text{Tỷ lệ chia trước})$
- /4: theo cấu trúc của Timer
- Tỷ lệ chia trước: 1,2,4,8 (Timer 1)
- VD: $f_{\text{TA}} = 20 \text{ MHz}$, tỷ lệ chia trước = 8 → $T = 1.6 \text{ uS}$

2. Xác định số xung cần đếm = thời gian cần tạo / T

VD: Tạo 1s → $1\text{s} / 1.6\text{uS} = 1.000.000 / 1.6 = 625\,000$ xung

Ta có: Timer 1 đếm tối đa ≤ 65.535 xung (16 bit) → Timer đếm nhiều lần ($62.500 \text{ xung} \times 10 \text{ lần}$)

3. X.định giá trị bắt đầu của Timer = số tràn – số xung cần đếm

- Số tràn = số đếm lớn nhất + 1 = $65535 + 1 = 65.536$

- Timer bắt đầu = $65.536 - 62.500 = 3036$

Thực hành

1. Viết chương trình delay 400 ms dùng timer1, thạch anh 8 MHz

```
Setup_timer_1(T1_INTERNAL|T1_DIV_BY_2); //1us/xung
```

Giải

1. Xác định chu kỳ T của ngõ vào xung của Timer.

$f_{TA} = 8 \text{ MHz}$ chia 4 = 2 Mhz

Tỷ lệ chia trước đề bài cho = 2 $\rightarrow f_{timer} = 2\text{Mhz} / 2 = 1\text{Mhz}$

$T_{timer} = 1 / f_{timer} = 1 / 1\text{Mhz} = 1 \text{ uS}$

2. Xác định số xung cần đếm = thời gian cần tạo / T

Tạo hàm delay 400ms $\rightarrow 400\text{ms} / 1\text{uS} = 400\,000 / 1 = 400\,000\text{xung}$

Ta có: Timer 1 đếm tối đa $\leq 65\,535$ xung (16 bit) \rightarrow Timer đếm nhiều lần (50 000 xung x 8 lần = 400 000xung)

3. X.định giá trị bắt đầu của Timer = số tràn – số xung cần đếm

- Số tràn = số đếm lớn nhất + 1 = 65 535 + 1 = 65 536

- Timer bắt đầu = 65 536 – 50 000 = 15 536

```
void delay400ms()
{
    Setup_timer_1(T1_INTERNAL|T1_DIV_BY_2); //1us/xung
    For (int8 i = 0; i < 8; i++) // cần 400 000 xung = 8*50 000
    {
        Set_timer1(15536);

        While (get_timer1() > 15536) ;
    }
}
```

Ghi chú:

1. SET_TIMERX():

SET_TIMER0(0);

\rightarrow Thiết lập giá trị cho Counter/Timer (ban đầu counter = 0 vì chưa có sản phẩm nào)

\rightarrow Tùy theo Timer 0,2 (giá trị thiết lập là 8 bit ≤ 255), Timer 1 (16 bit ≤ 65535)

2. GET_TIMERX():

\rightarrow Đọc giá trị của Counter/Timer.

Các lệnh liên quan timer 0:

Setup_timer_0(RTCC_DIV_4|RTCC_INTERNAL)

RTCC_INTERNAL: xung clock $f_{TA} / 4$

(Chu kỳ lệnh = 4 * chu kỳ máy; Chu kỳ máy = 1/ tần số của vi điều khiển)

RTCC_DIV_4: tần số xung clock chia 4 trước khi đưa vào bộ đếm

Set_timer0(byte)

Đặt trước giá trị ban đầu vào bộ đếm timer 0 (0-255)

Ví dụ:

```
Set_timer0(56); // tràn sau 200 xung clock
```

256 = giá trị đặt trước + (giá trị mong muốn/tỷ lệ chia * chu kỳ lệnh)) us

Get_timer0()

- Đọc giá trị hiện tại trong timer 0

- Ví dụ:

a = get_timer0() // đọc giá trị hiện tại của timer0 và gán vào biến a

Ví dụ:

Sử dụng thạch anh 4Mhz, timer 0 (8bit), bộ chia tần số 4, đếm (delay) 200us thì giá trị nạp trước tính theo công thức sau:

$$256 - (200/4.1) = 206$$

Giải thích các số trên như sau:

256 vì đây là Timer 8 bit

200: đếm 200 us

4: bộ chia tần số 4

1: chu kỳ lệnh = $4 * (1 / f_{TA})$

Như vậy chúng ta phải cài đặt giá trị cho Timer 0 như sau

```
Setup_timer_0(RTCC_INTERNAL|RTCC_DIV_4);
```

```
Set_timer0(206);
```

Chương trình con delay x_ms, thạch anh 8Mhz

```
void dl_xms(int16 x)
```

```
{
```

```
    setup_timer_0(RTCC_DIV_8|RTCC_INTERNAL);
```

```
    for(int16 i = 0 ; i<x; i++)
```

```
    {
```

```
        set_timer0(6);
```

```
        while(get_timer0()<255)
```

```
    }
```

```
}
```