

PIC16F877A

CQ.ITS59
1-15-2021

Một vài lệnh cơ bản

Config chân : **TRIS# = 0** hoặc **1**; (trong đó : # là tên port A,B,C,D; **0** là OUTPUT **1** là INPUT;

Ví dụ : TRISB=0 (PortB là output)

Nhập xuất I/O: **PORTB# = 0xFF**; xuất một PORT

Hoặc **R#* = 0**; xuất theo chân (trong đó * là tên chân từ 0-7)

Ví dụ: RB1 = 0

Trước tiên là bảng mã HEX từ 0 -15 như hình 1.

8	4	2	1	MÃ HEX
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

Và đây là bảng tính mã HEX cụ thể để xuất ra các chân của Vi Điều Khiển để điều khiển LED sáng theo mong muốn. Các bạn xem hình 2.

8	4	2	1	8	4	2	1	KQ
1	0	1	0	1	0	1	0	AA
0	1	0	1	0	1	0	1	55
1	1	1	1	0	0	0	0	F0

(Tài liệu này soạn ra chỉ với mục đích tham khảo . Có thể đúng có thể sai nên các bạn hãy xem lại nhé)

Một số thư viện

Thư viện lcd4.h

<pre>#define rs RC0 #define en RC1 void lcd_init(); void cmd(unsigned char a); void dat(unsigned char b); void show(unsigned char *s); void lcd_delay(); void lcd_init() { cmd(0x02); cmd(0x28); cmd(0x0e); cmd(0x06); cmd(0x80); } void cmd(unsigned char a) { rs = 0; PORTB &= 0x0F; PORTB = (a & 0xf0); en = 1;</pre>	<pre> lcd_delay(); en = 0; lcd_delay(); PORTB &= 0x0f; PORTB = (a << 4 & 0xf0);} en = 1; lcd_delay(); en = 0; lcd_delay(); } void dat(unsigned char b) { rs = 1; PORTB &= 0x0F; PORTB = (b & 0xf0); en = 1; lcd_delay(); en = 0; lcd_delay(); PORTB &= 0x0f;</pre>	<pre> PORTB = (b << 4 & 0xf0); en = 1; lcd_delay(); en = 0; lcd_delay(); void show(unsigned char *s) { while (*s) { dat(*s++); } } void lcd_delay() { unsigned int lcd_delay; for (lcd_delay = 0; lcd_delay <= 1000; lcd_delay++); }</pre>
---	--	--

Thư viện lcd8.h

```
// lcd8.h
#define lcd_dat PORTD
#define rs RC0
#define rw RC1
#define en RC2
#define          delay
for(i=0;i<1000;i++)

int i;

void lcd_init();
void cmd(unsigned char a);
void dat(unsigned char b);
void show(unsigned char *s);

void lcd_init()
{
```

```
TRISC0=TRISC1=TRISC2=
TRISD=0;

cmd(0x38);
cmd(0x0e);
cmd(0x06);
cmd(0x80);
}

void cmd(unsigned char a)
{
    lcd_dat=a;
    rs=0;
    rw=0;
    en=1;
    delay;
    en=0;
}
```

```
void dat(unsigned char b)
{
    lcd_dat=b;
    rs=1;
    rw=0;
    en=1;
    delay;
    en=0;
}

void show(unsigned char *s)
{
    while(*s)
    {
        dat(*s++);
    }
}
```

UART

```
// serial.h
```

```
void serial_init();
```

```
void tx(unsigned char dat);
```

```
unsigned char rx();
```

```
void tx_string(unsigned char *s);
```

```
void serial_init()
```

```
{
```

```
    TRISC6=TRISC7=1;
```

```
    TXSTA=0b00100010;
```

```
    RCSTA=0b10010000;
```

```
    SPBRG=17;
```

```
    TXIF=RCIF=0;
```

```
}
```

```
void tx(unsigned char dat)
```

```
{
```

```
    TXREG=dat;
```

```
    while(!TXIF);
```

```
}
```

```
unsigned char rx()
```

```
{
```

```
    while(!RCIF);
```

```
    return RCREG;
```

```
}
```

```
void tx_string(unsigned char *s)
```

```
{
```

```
    while(*s)
```

```
    {
```

```
        tx(*s++);
```

```
    }
```

```
}
```

Lcd8_2.h

//lcd8_2.h

#define rs RC0

#define rw RC1

#define en RC2

#define

delay for(i=0;i<1000;i++)

int i;

void lcd_init();

void cmd(unsigned char a);

void dat(unsigned char b);

void show(unsigned char *s);

void lcd_init(){

cmd(0x38);

cmd(0x0c);

cmd(0x06);

cmd(0x80);

}

void cmd(unsigned char a)

{

PORTB=a;

rs=0;

rw=0;

en=1;

delay;

en=0;

}

void dat(unsigned char b)

{

PORTB=b;

rs=1;

rw=0;

en=1;

delay;

en=0;

}

void show(unsigned char
*s){

while(*s)

{ dat(*s++); }

Thư viện keypad

```
//key.h

#define R1 RB0
#define R2 RB1
#define R3 RB2
#define R4 RB3
#define C1 RB4
#define C2 RB5
#define C3 RB6
#define C4 RB7

#define delay for(int i; i<10000; i++)

unsigned char key();
void keyinit();

unsigned char keypad[4][4] = {
    {'7', '8', '9', '/'},
    {'4', '5', '6', '*'},
    {'1', '2', '3', '-'},
    {'C', '0', '=', '+'}
};

unsigned char rowloc, colloc;

void keyinit() {
    TRISB = 0XF0;
    OPTION_REG  &=  0X7F;
//ENABLE PULL UP
}

unsigned char key() {
    RC5 = 0;
    PORTB = 0X00;
    while (C1 && C2 && C3 && C4);
    while (!C1 || !C2 || !C3 || !C4) {
        R1 = 0;
        R2 = R3 = R4 = 1;
        if (!C1 || !C2 || !C3 || !C4) {
            rowloc = 0;
            break;
        }
        R2 = 0;
        R1 = 1;
        if (!C1 || !C2 || !C3 || !C4) {
            rowloc = 1;
            break;
        }
        R3 = 0;
        R2 = 1;
        if (!C1 || !C2 || !C3 || !C4) {
            rowloc = 2;
            break;
        }
        R4 = 0;
        R3 = 1;
        if (!C1 || !C2 || !C3 || !C4) {
            rowloc = 3;
            break;
        }
    }
    if (C1 == 0 && C2 != 0 && C3 != 0 && C4 != 0) {
        colloc = 0;
        RC5 = 0x00;
        delay;
        RC5 = 0xff;
    } else if (C1 != 0 && C2 == 0 && C3 != 0 && C4 != 0) {
        colloc = 1;
        RC5 = 0x00;
        delay;
        RC5 = 0xff;
    } else if (C1 != 0 && C2 != 0 && C3 == 0 && C4 != 0) {
        colloc = 2;
        RC5 = 0x00;
        delay;
        RC5 = 0xff;
    } else if (C1 != 0 && C2 != 0 && C3 != 0 && C4 == 0) {
        colloc = 3;
        RC5 = 0x00;
        delay;
        RC5 = 0xff;
    }
    while (C1 == 0 || C2 == 0 || C3 == 0 || C4 == 0);
    return
    (keypad[rowloc][colloc]);
}

}
```

Thư viện timers.h

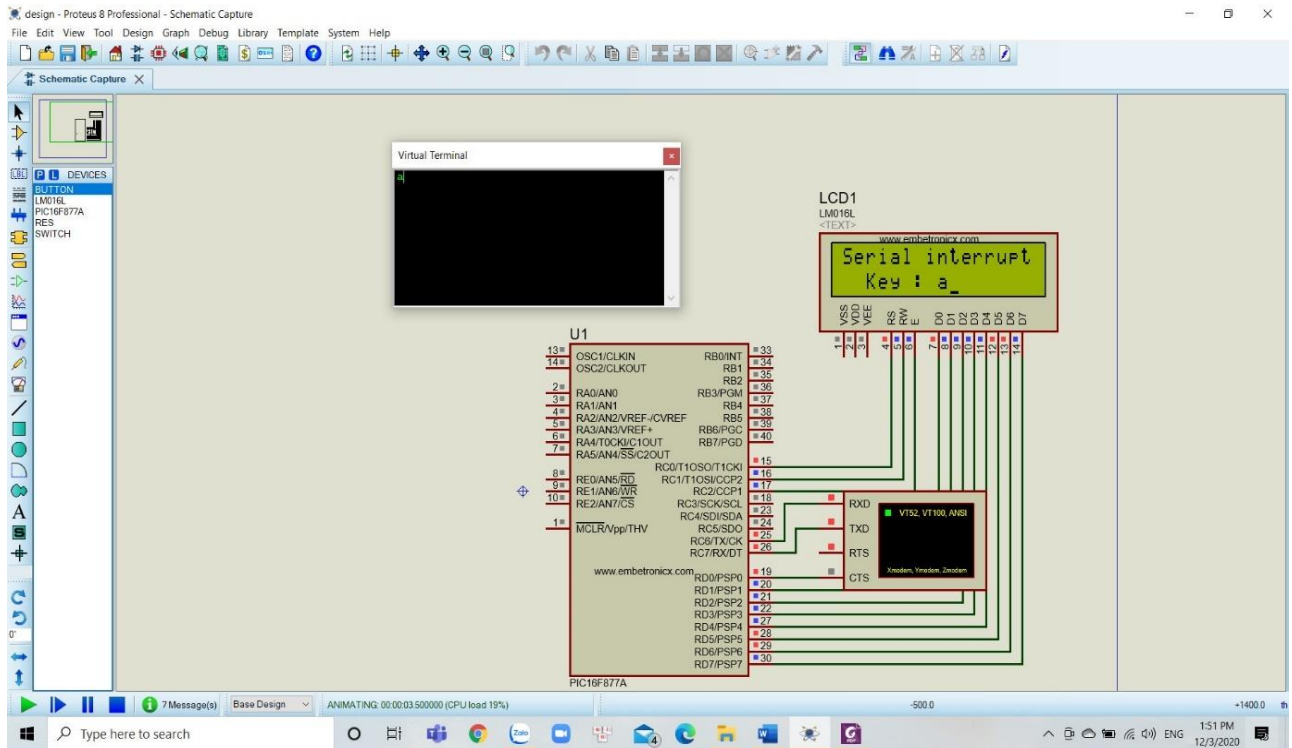
//timers.h	void t0delay(unsigned int a)	while(!TMR1IF);
void timer_int();	{	TMR1IF=0;
void t0delay(unsigned int);	unsigned int i;	}
void t1delay(unsigned int);	for(i=0;i<a;i++)	}
void t2delay(unsigned int);	{	void t2delay(unsigned int c)
void timer_int()	while(!TMR0IF);	{
{	TMR0IF=0; }	unsigned int i;
OPTION_REG=0b00000111;//int	}	T2CON =(1<<2); //timer2 on
ernal clk,rising edge,prescaler	void t1delay(unsigned int b)	for(i=0;i<c;i++){
with tim0,256	{	while(!TMR2IF);
T1CON=0b00000000;//prescale=	unsigned int i;	TMR2IF=0;
1,oscillator is off,internal clk,timer	T1CON =(1<<0); //timer1 on	}
off	for(i=0;i<b;i++){	}
T2CON=0b01111000;//postscale	TMR1H=TMR1L=0;	
=16,prescale=1,timer off		
}		

Tổng hợp các bài tập về PIC16F877A

I. Các bài tập lớn trên lớp

1. UART Terminal

Thiết lập sơ đồ phần cứng kết nối vi xử lý PIC16F877A với máy tính để truyền thông tin. Chương trình này, trước tiên hãy truyền một chuỗi ký tự (Nhóm ... test USART): Nhập ký tự tu tu ban phim may tinh:). Sau đó, nó sẽ hoạt động như một bộ vọng echo. Khi nhấn bàn phím, ký tự nhận được hiển thị tại serial terminal (thiết bị đầu cuối nối tiếp) và màn hình LCD.



Code

```
#include<pic.h>

#include"serial.h"

#include"lcd8.h"

__CONFIG( FOSC_HS &
WDTE_OFF & PWRTE_OFF &
CP_OFF & BOREN_ON & LVP_OFF
& CPD_OFF & WRT_OFF &
DEBUG_OFF);

#define delay for(z=0;z<=50000;z++)

unsigned int z;

void interrupt ser();
```

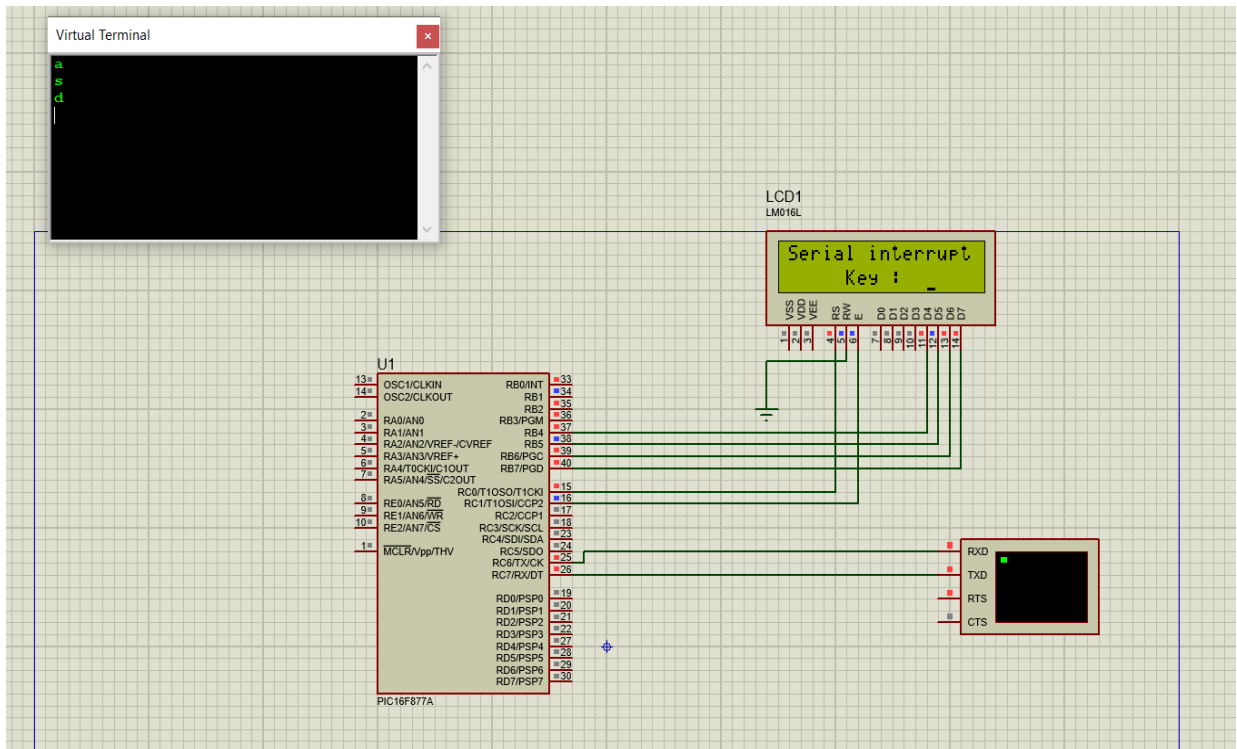
```
void main()
{
    TRISD=0;
    INTCON|=0b11000000;
    PIE1=0b00100000;
    lcd_init();
    serial_init();
    while(1) {
        cmd(0x01);
    }
}

void interrupt ser()
{
```

```
    unsigned char a = RCREG;
    tx(a);
    cmd(0x80);
    show("Serial interrupt");
    cmd(0xc0);
    show(" Key : ");
    cmd(0xc8);
    dat(a);
    delay;delay;
    TXIF=RCIF=0;
}
```

(thư viện mình viết ở trên nhớ chép vào nhé 😊)

Cũng là bài trên nhưng viết theo LCD4 bit



```
#include <pic.h>
#include "uart.h"
#include "lcd4.h"

void main(void) {
    char rx;
    TRISB = TRISC0 = TRISC1 = 0; // chan
    c0 c1 output
    TRISB0 = 0;
    lcd_init();
    cmd(0x80);
    show("Key : ");
    UART_init();
    UART_printf("Bai Tap Lon ITS59\n\r");
    while(1)
    {
        rx = UART_read();
        UART_printf(rx);
        cmd(0xc8);
        show(rx);
    }
}
```

```
}
return;}

//thu vien uart.h

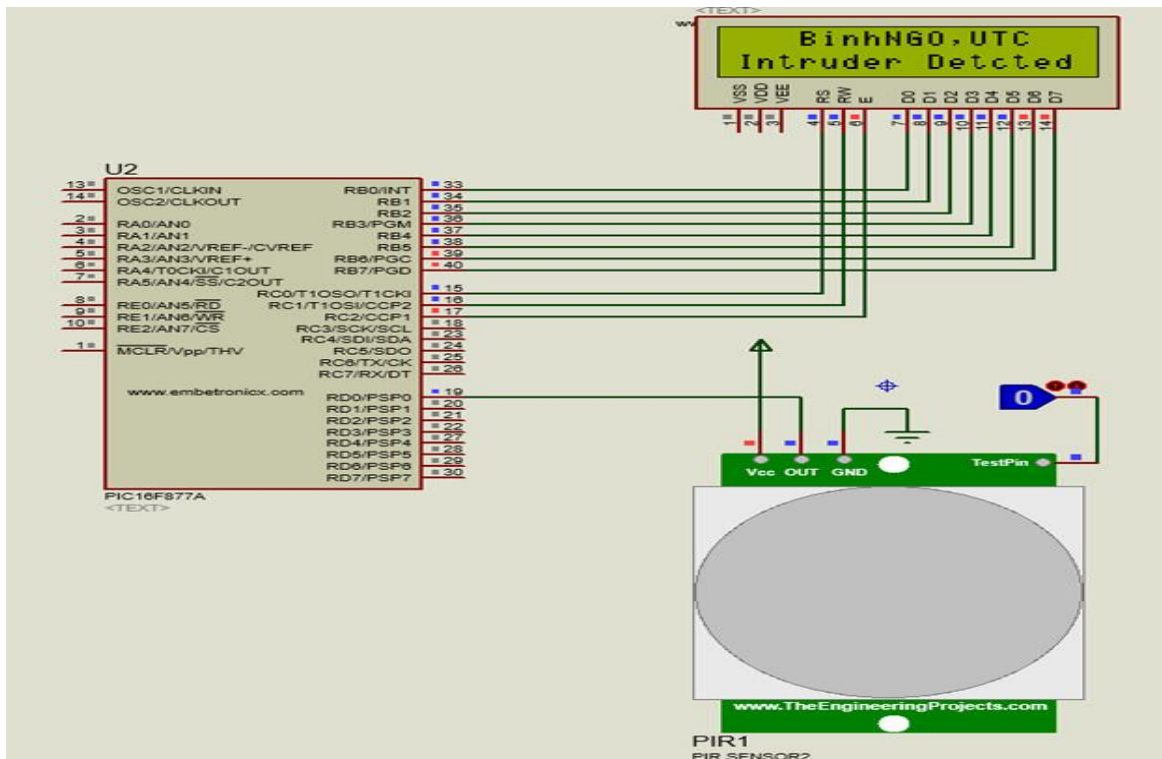
void UART_init()
{
    TRISC7 = 1; //RX input
    TRISC6 = 0; //TX output
    SYNC = 0; //UART
    TX9 = 0; //8 BITS
    BRGH = 1; //ALTA VELOCIDAD
    SPBRG = 129; //9600 a 20MHZ
    SPEN = 1; //UART on
    TXEN = 1; //TX on
    CREN = 1; //RX on
}

char UART_read(void)
{
    if (RCIF == 1)
        return RCREG;
}
```

```
else
    return 0;
}

void UART_write(char data)
{
    TXREG = data;
    while(!TXIF);
    TXIF = 0;
}

void UART_printf(char *s)
{
    while(*s != 0x00){
        UART_write(*s);
        s++;
    }
}
```



2. PIR SENSOR

Code

```
#include<htc.h>

#include"lcd8_2.h"// chú ý thư
viện ở trên

__CONFIG( FOSC_HS &
WDTE_OFF & PWRTE_OFF &
CP_OFF & BOREN_ON &
LVP_OFF & CPD_OFF &
WRT_OFF & DEBUG_OFF);

#define PIR RD0

void main()
{
TRISB=TRISC0=TRISC1=TRIS
C2=0;

TRISD=0xff;
//Port D act as Input

lcd_init();

cmd(0x80);

//show(" EmbeTronicX ");
show(" BinhNGO,UTC ");
```

```
while(1) {

if(PIR == 0) {

cmd(0xc0);

show("Intruder Detcted");

// sáng đèn, quay động cơ servo
mở cửa

delay;delay;

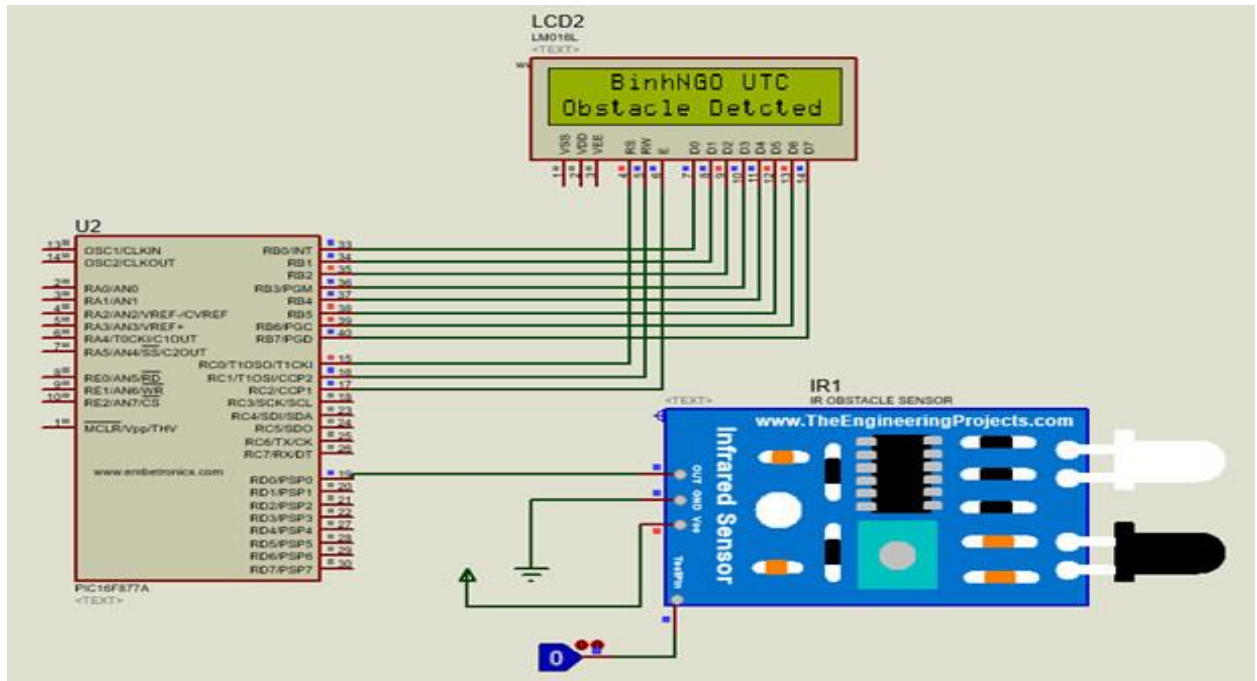
} else {

cmd(0xc0);

show(" ");

}}}
```

3. Cảm biến IR



Code

```
#include<htc.h>

#include"lcd8_2.h"

__CONFIG( FOSC_HS &
WDTE_OFF & PWRTE_OFF &
CP_OFF & BOREN_ON &
LVP_OFF & CPD_OFF &
WRT_OFF & DEBUG_OFF);

#define IR RD0

void main(){
```

```
TRISB=TRISC0=TRISC1=TRIS
C2=0;

TRISD=0xff;
//Port D act as Input

lcd_init();

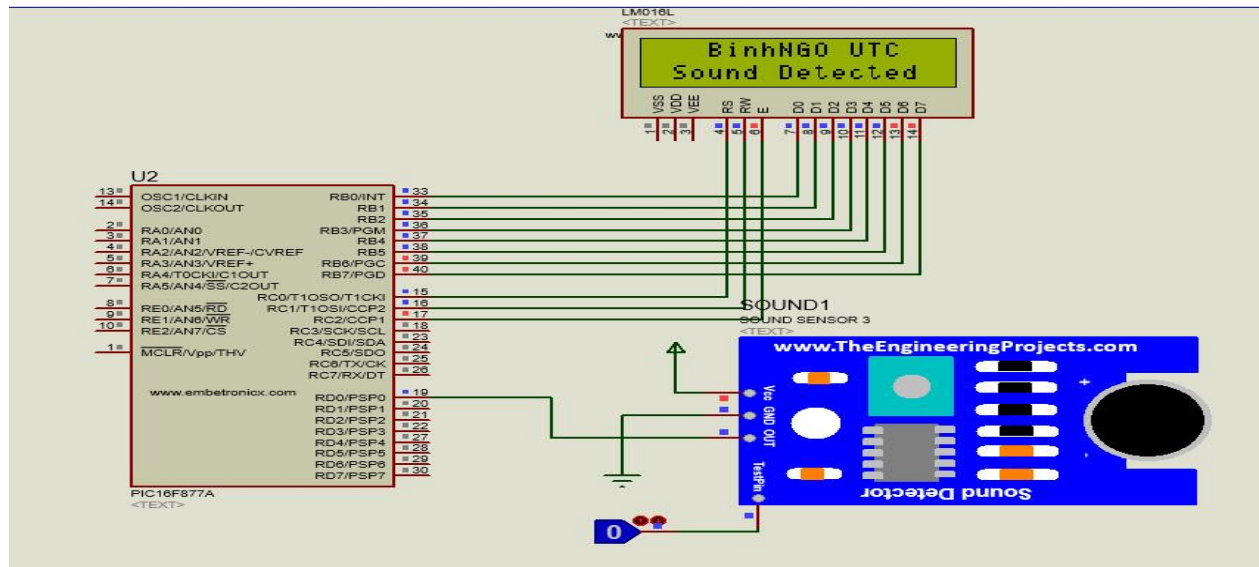
cmd(0x80);

show(" EmbeTronicX ");

while(1) {
```

```
if(IR == 0) {
    cmd(0xc0);
    show("Obstacle Detcted");
    delay;delay;
} else {
    cmd(0xc0);
    show(" ");}}}
```

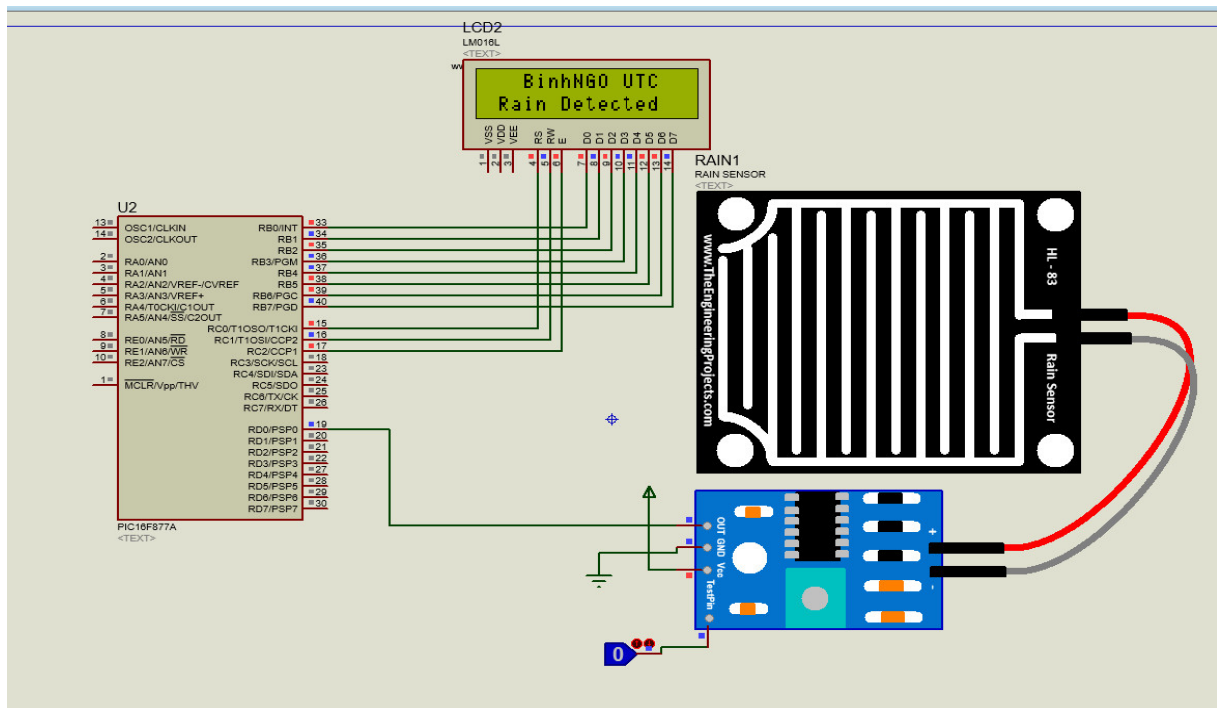
4. Cảm biến âm thanh



Code

<pre>#include<htc.h> #include"lcd8_2.h" __CONFIG(0x373A); //0b0011 0111 0011 1010 #define SOUND RD0 //Sound Sensor Output is connected at PORTD.0 void main() {</pre>	<pre>TRISB=TRISC0=TRISC1=TRIS C2=0; TRISD=0xff; //Port D act as Input lcd_init(); cmd(0x80); show(" BinhNGO UTC "); while(1) { if(SOUND == 0) {</pre>	<pre>cmd(0xc0); show(" Sound Detected"); delay;delay; } else { cmd(0xc0); show(" "); } }</pre>
---	---	---

5. Cảm biến mưa



Code

```
#include<htc.h>

#include"lcd8_2.h"

__CONFIG(0x373A); //0b0011
0111 0011 1010

#define RAIN RD0 //Rain
sensor Output

void main()
{
```

```
TRISB=TRISC0=TRISC1=TRIS
C2=0;

TRISD=0xff;
//Port D act as Input

lcd_init();

cmd(0x80);

show(" BinhNGO UTC ");

while(1) {

if(RAIN == 0) {
```

```
cmd(0xc0);

show(" Rain Detected");

delay;delay;

} else {

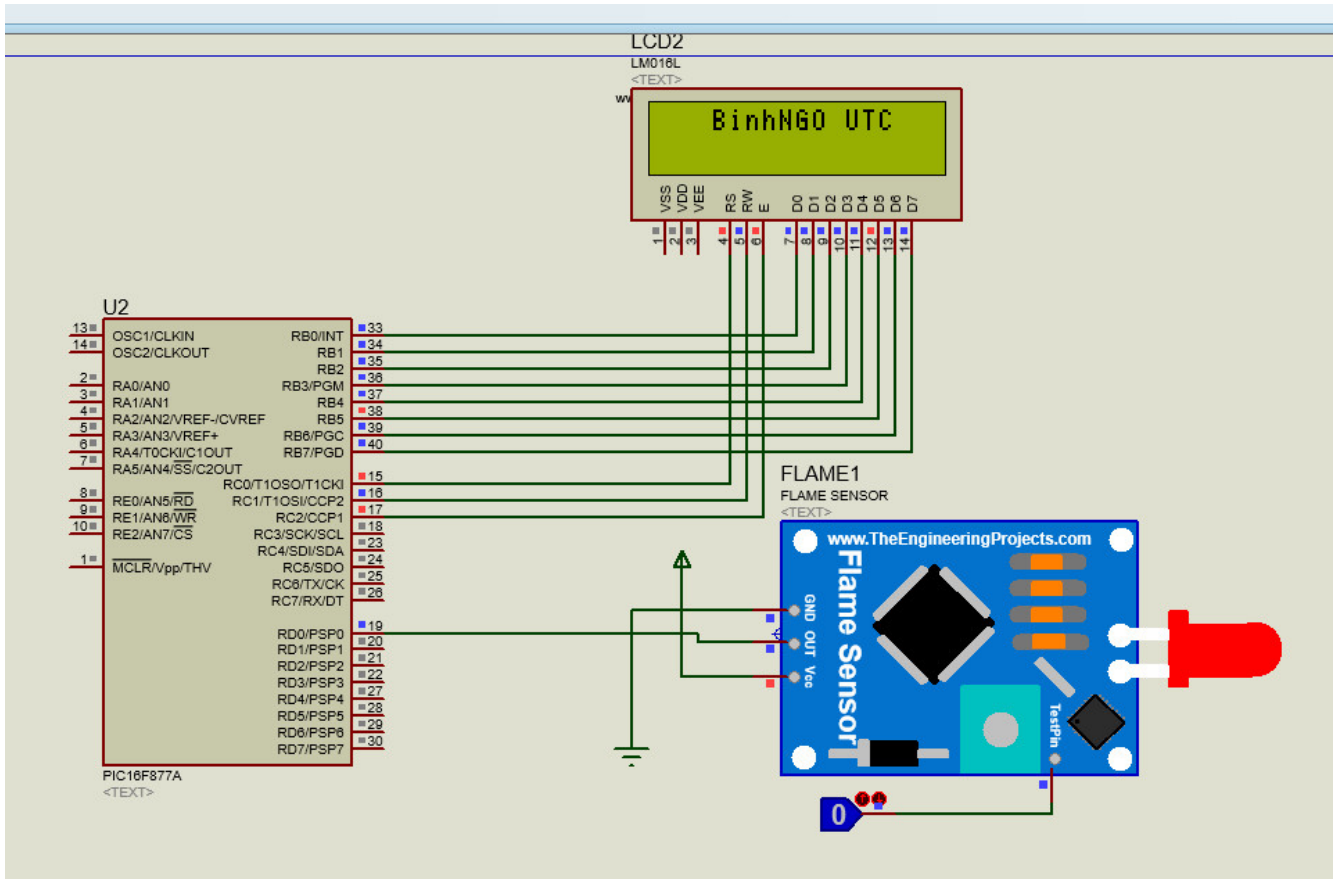
cmd(0xc0);

show(" ");

}

}
```

6. Mô-đun báo cháy Flame Sensor



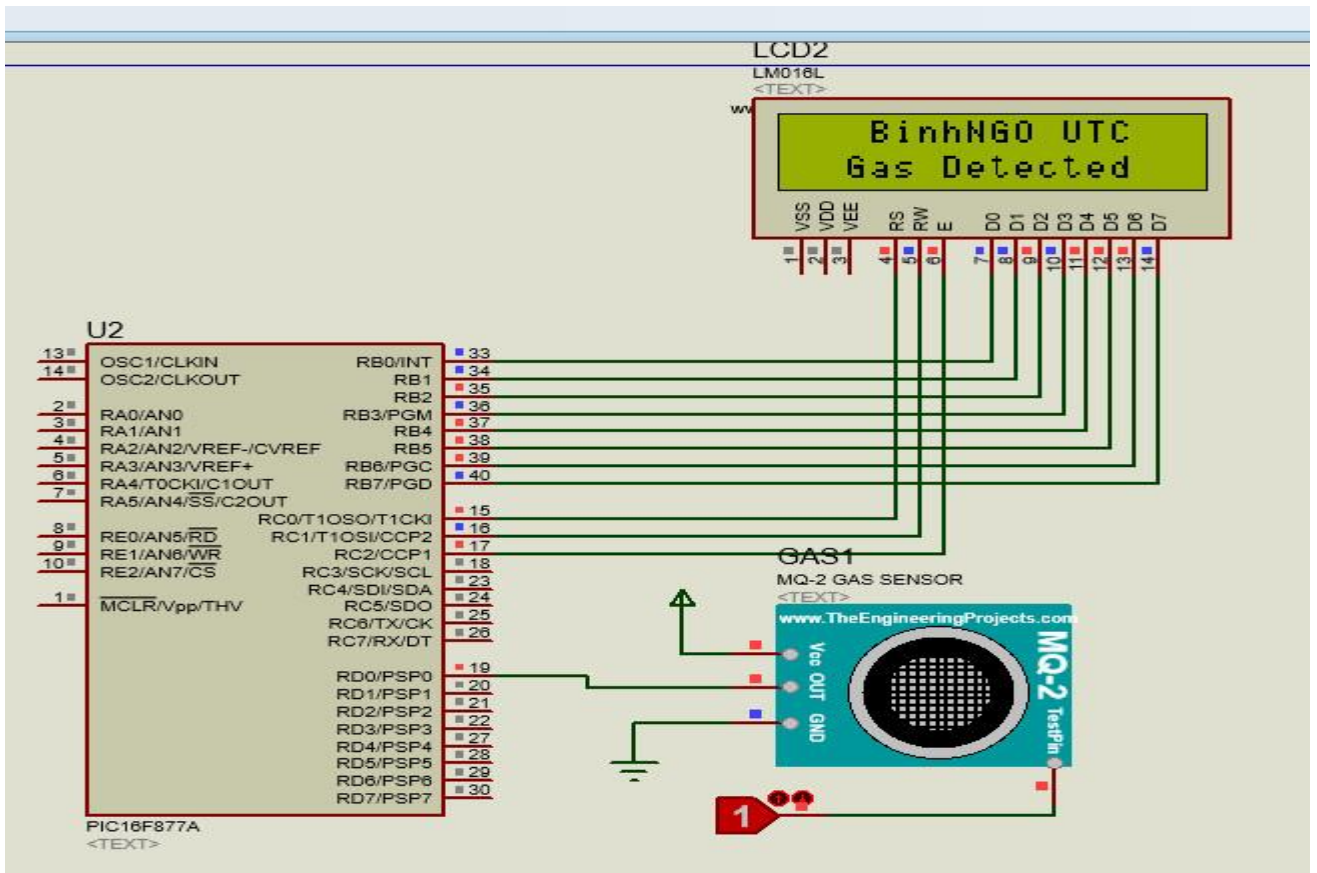
Code

```
#include<htc.h>
#include"lcd8_2.h"
__CONFIG(0x373A); //0b0011
0111 0011 1010

#define FLAME RD0
//Flame sensor

void main()
{
    TRISB=TRISC0=TRISC1=TRISC2=0;
    TRISD=0xff; //Port D
    act as Input
    lcd_init();
    cmd(0x80);
    show(" BinhNGO UTC ");
    while(1) {
        if(FLAME) {
            cmd(0xc0);
            show(" Flame Detected");
            delay;delay;
        } else {
            cmd(0xc0);
            show(" ");
        }
    }
}
```


7. Cảm biến khí gas LPG (LPG Gas Sensor);



Code

```
#include<htc.h>
#include"lcd8_2.h"

__CONFIG(0x373A); //0b0011
0111 0011 1010

#define GAS RD0
//GAS sensor Output

void main()
{
    TRISB=TRISC0=TRISC1=TRISC2=0;

    TRISD=0xff;
    //Port D act as Input

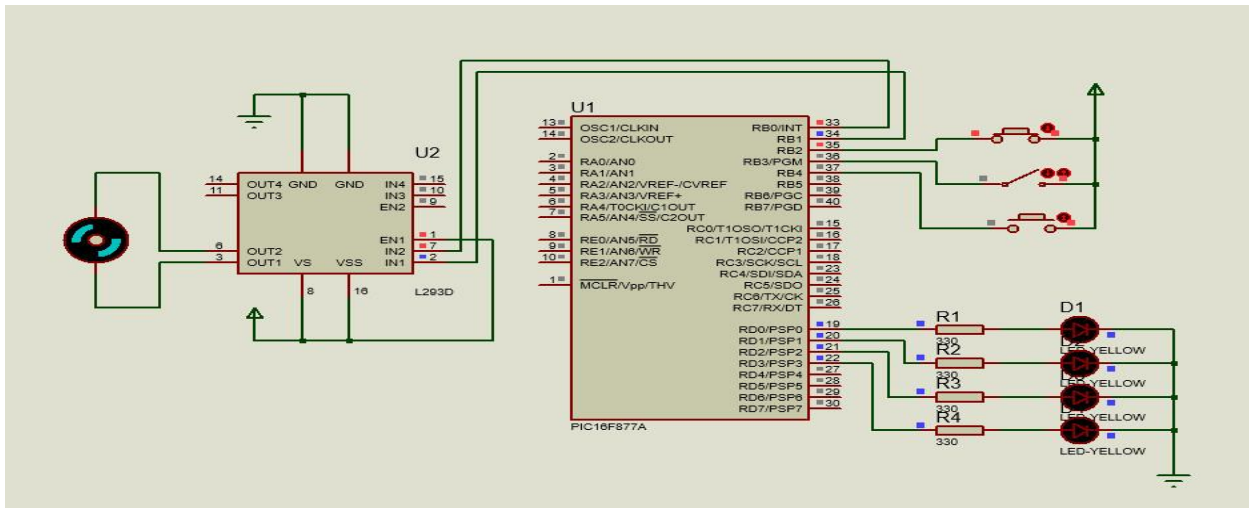
    lcd_init();
    cmd(0x80);

    show(" BinhNGO UTC ");
    while(1) {
        if(GAS) {
            cmd(0xc0);
            show(" Gas Detected");
            delay;delay;
        } else {
            cmd(0xc0);
            show(" "); } }

    en=1;
    delay;
    en=0;}

void show(unsigned char *s){
    while(*s) {
        dat(*s++); }}
```


8. Điều khiển động cơ 1 chiều



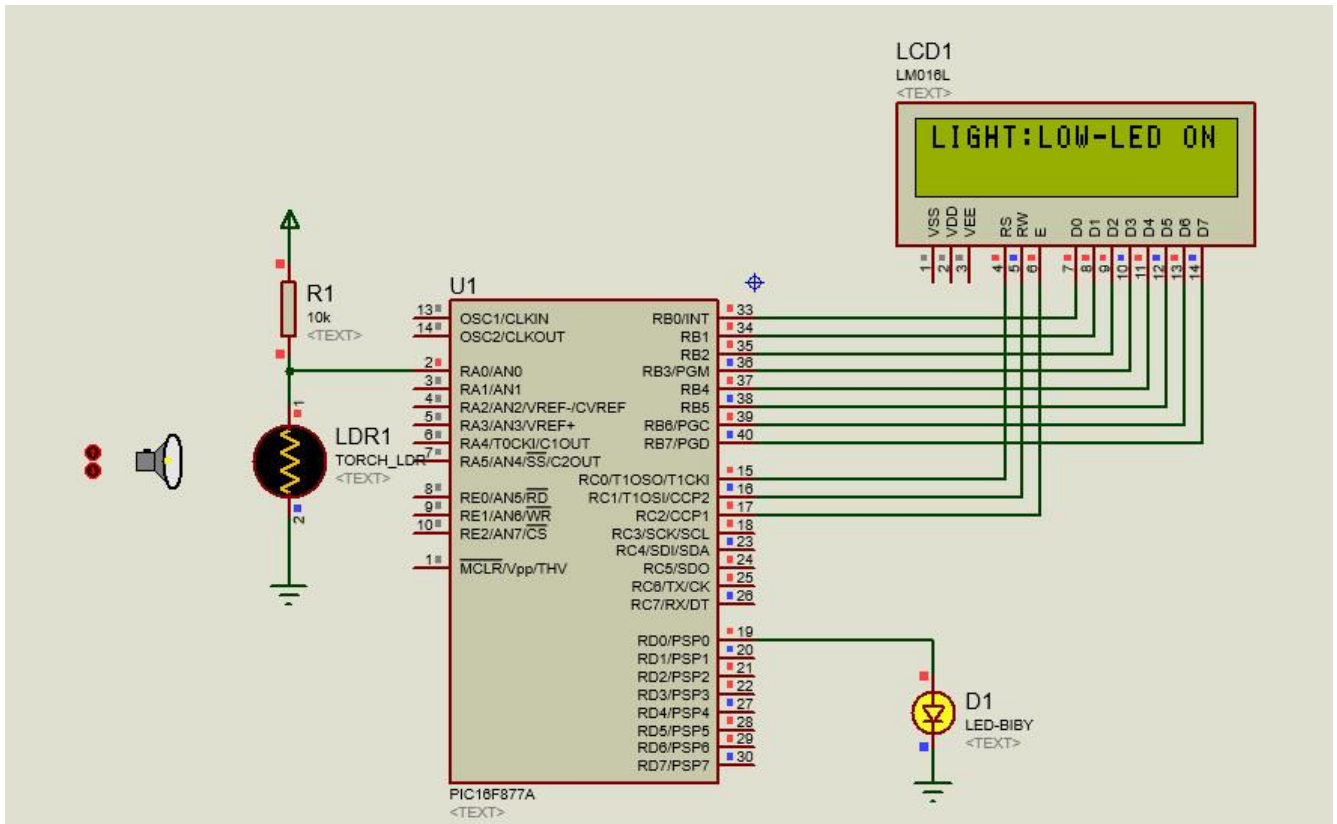
Code

```
#include<pic.h>
#define in1 RB0
#define in2 RB1
#define sw RB2
#define sw2 RB3
#define sw3 RB4

void main()
{
    TRISB0=0;
    TRISB1=0;
    TRISB2=TRISB3=TRISB4=1;
    while(1) {
        if(sw==1) {
            in1=1;
            in2=0;
            // LCD quay thuaan
        } else {
            in1=in2=0;
        }
    }
}
```

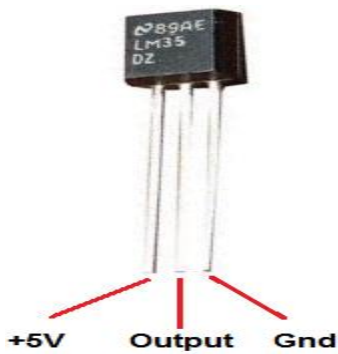
(này khỏi vẽ mấy con led cũng đc)

9. Cảm biến LDR



```
#include<pic.h>
#include"lcd8_2.h"
#define delay
for(i=0;i<=1000;i++)
    __CONFIG(0x373A); //0b0011
    0111 0011 1010
#define LED RD0
unsigned int adc();
void main()
{
    unsigned int val;
    TRISB=TRISC=0; //Port
    B and Port C is Output (LCD)
    TRISD = 0; //Port D is
    output LED
    TRISA0=1; //RA0 is
    input (ADC)
    lcd_int();
    while(1) {
        cmd(0x80);
        val = adc();
        show("LIGHT:");
        if(val>150) {
            show("LOW-LED ON");
            LED = 1;
        } else {
            show("HI-LED OFF");
            LED = 0; }
    }
}
unsigned int adc()
{
    unsigned int adcval;
    ADCON1=0xc0;
    //right justified
    ADCON0=0x85;
    //adc on, fosc/64
    while(GO_nDONE);
    adcval=((ADRESH<<8)|(ADRES
    L)); /
    adcval=(adcval/3)-1;
    return adcval;
}
```

10. Đo nhiệt độ hiển thị lên LCD

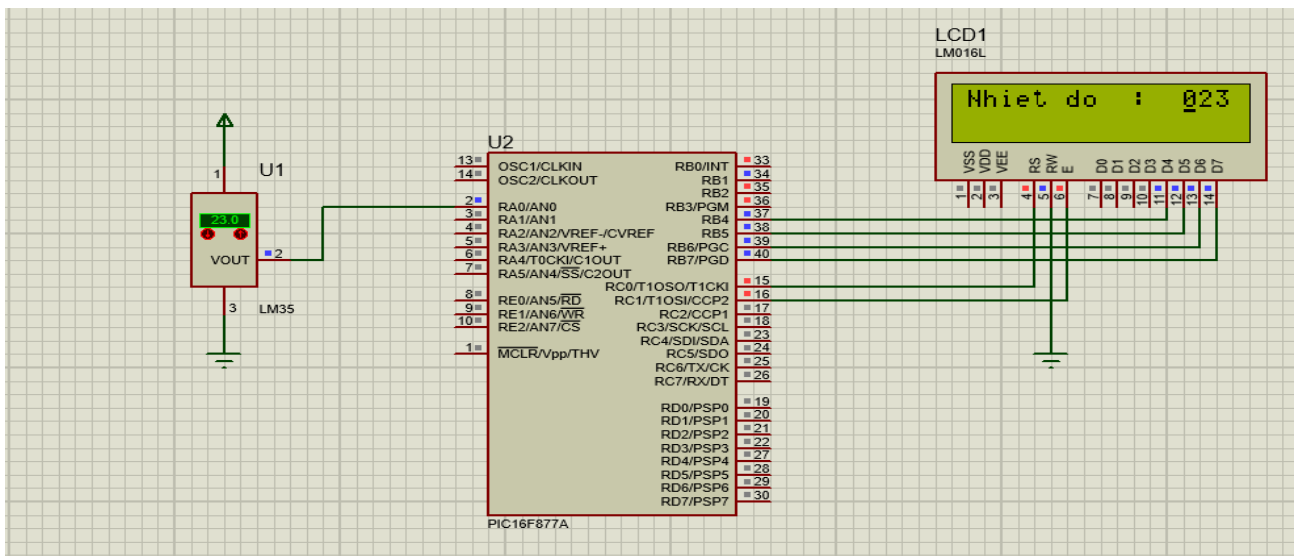


Chân dữ liệu của LM35 là chân ngõ ra điện áp dạng tuyến tính. Chân số 2 cảm biến xuất ra cứ $1\text{mV} = 0.1^{\circ}\text{C}$ ($10\text{mV} = 1^{\circ}\text{C}$). Để lấy dữ liệu ở dạng $^{\circ}\text{C}$ chỉ cần lấy điện áp trên chân OUT đem chia cho 10.

Chân 1 cấp điện áp 5V, chân 3 cấp GND, chân 2 là chân OUTPUT dữ liệu dạng điện áp.

Như vậy, bằng cách đưa vào chân bên trái của cảm biến LM35 hiệu điện thế 5V, chân phải nối đất, đo hiệu điện thế ở chân giữa bằng các pin A, bạn sẽ có được nhiệt độ (0-100°C) bằng công thức:

```
voltage = 5000.0f / 1023 * ADCvalue;
```



```
#include <pic.h>

#include "lcd4.h"

__CONFIG(0x373A); //0b0011

void setup() {

    TRISB = TRISC0 = TRISC1 =
0; // chan C0 C1 output

    TRISA0 = 1; // KHAI BAO
CHAN A0 LA INPUT

}

void adc() {

    unsigned int adcval;
```

```
ADCON1 = 0xc0; //right
justifie
ADCON0 = 0x85; //adc on, f
while (GO_nDONE); //wait
adcval = ((ADRESH << 8) |
(ADRESL)); //store the result
//adcval = (adcval / 3) - 1;

// dat((adcval / 1000) + 48);
adcval = (adcval / 2);
dat(((5000.0f/1023*adcval)/10));
// theo cong thuc
```

```

    dat(((adcval / 100) % 10) + 48);
    dat(((adcval / 10) % 10) + 48);
    dat((adcval % 10) + 48);}

void main() { setup();

    lcd_init();

    //cmd(0x90);

    show("Nhiet do : ");

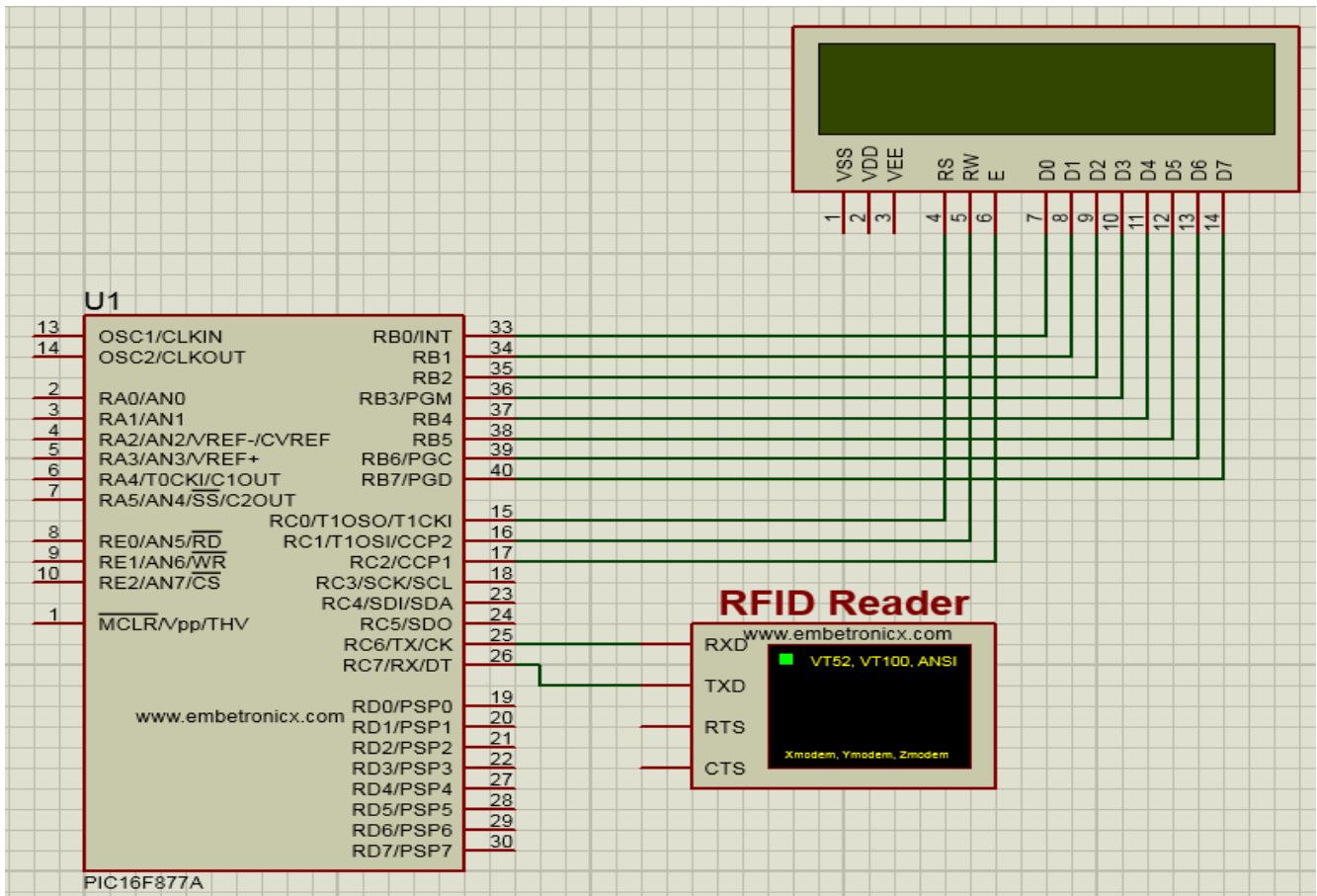
    while (1) {

        cmd(0x8c);

        adc(); } }

```

11.RFID



```
#include <pic.h>
```

```
#define lcd_dat PORTB
```

```
#define rs RC0
```

```
#define rw RC1
```

```
#define en RC2
```

```
#define delay for(i=0;i<1000;i++)
```

```
int i;
```

```
void lcd_init();
```

```
void cmd(unsigned char a);
```

```
void dat(unsigned char b);
```

```
void show(unsigned char *s);
```

```
_CONFIG( FOSC_HS &
WDTE_OFF & PWRTE_OFF &
CP_OFF & BOREN_ON &
LVP_OFF & CPD_OFF &
WRT_OFF & DEBUG_OFF);
```

```
void ser_int();
```

```
void tx(unsigned char);
```

```
unsigned char rx();
```

```
void txstr(unsigned char *);
```

```
void main()
```

```
{
```

```
TRISD=1;
```

```
int i;
```

```
unsigned char id[12];
```

```
TRISC6=TRISC7=1;
```

```
ser_int();
```

```
lcd_init();
```

```
cmd(0x80);
```

```
show("<<SHOW UR CARD>>");
```

```
cmd(0xc0);
```

```
for(i=0; i<12; i++) {
```

```
id[i]=rx();
```

```
dat(id[i]);
```

```
PORTD=0x00;
```

```
}
```

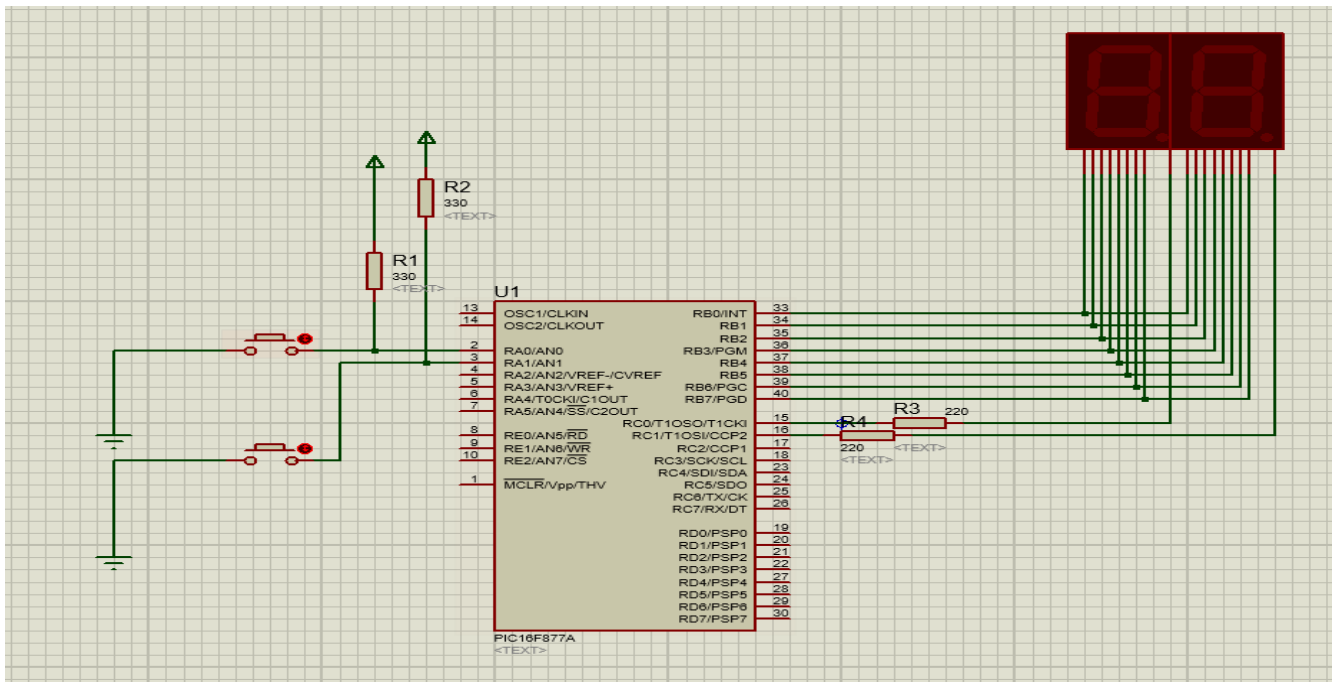
```
while(1);
```

```
}
```

<i>void ser_int()</i>	{	<i>delay;</i>
{	<i>while(*s)</i>	<i>en=0;</i>
<i>TXSTA=0x20;</i>	{	}
<i>RCSTA=0b10010000;</i>	<i>tx(*s++);</i>	
<i>SPBRG=17;</i>	}	<i>void dat(unsigned char b)</i>
<i>TXIF=RCIF=0;</i>	}	{
}		<i>lcd_dat=b;</i>
	<i>void lcd_init()</i>	<i>rs=1;</i>
<i>void tx(unsigned char a)</i>	{	<i>rw=0;</i>
{	<i>TRISC0=TRISC1=TRISC2=TRISB=</i>	<i>en=1;</i>
<i>TXREG=a;</i>	<i>0;</i>	<i>delay;</i>
<i>while(!TXIF);</i>	<i>cmd(0x38);</i>	<i>en=0;</i>
<i>TXIF = 0;</i>	<i>cmd(0x0c);</i>	}
}	<i>cmd(0x06);</i>	
	<i>cmd(0x80);</i>	<i>void show(unsigned char *s)</i>
<i>unsigned char rx()</i>	}	{
{		<i>while(*s)</i>
<i>while(!RCIF);</i>	<i>void cmd(unsigned char a)</i>	{
<i>RCIF=0;</i>	{	<i>dat(*s++);</i>
<i>return RCREG;</i>	<i>lcd_dat=a;</i>	}
}	<i>rs=0;</i>	}
	<i>rw=0;</i>	
<i>void txstr(unsigned char *s)</i>	<i>en=1;</i>	

II. LED 7 đoạn

1. Đếm từ 0-99 sử dụng nút bấm



Code

```
#include <pic.h>

#define _XTAL_FREQ 8000000

// CONFIG
__CONFIG(FOSC_HS & WDTE_OFF &
PWRTE_OFF & CP_OFF & BOREN_ON &
LVP_OFF & CPD_OFF & WRT_OFF &
DEBUG_OFF);

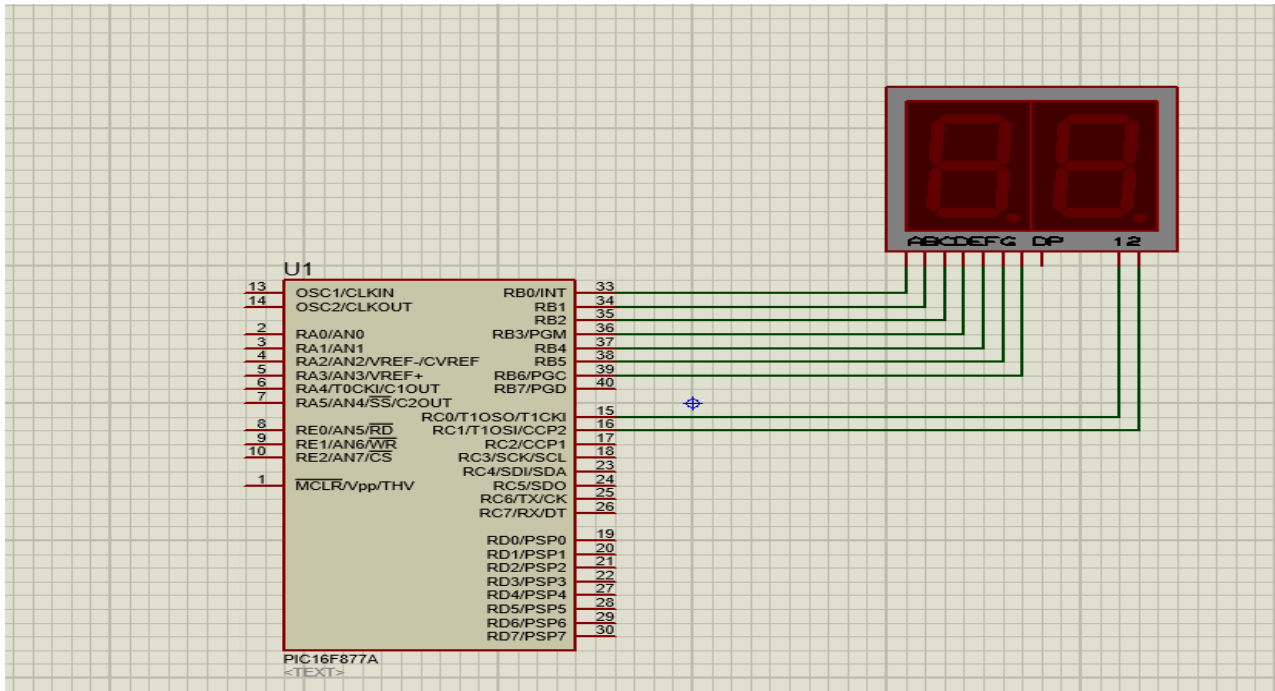
void displayed(unsigned int digit);

unsigned char const SEGMENT_MAP[10] =
{0xBF,0x86,0xDB,0xCF,0xE6,0xED,0xFD,0X
87,0xFF,0xEF};

void main(void) {
    ADCON1 = 0x07;
    TRISB = 0x00;    //Set PortB to all outputs
    TRISA0 = TRISA1 = 1;
    TRISC0 = TRISC1 = 0;
    char digit=0;
    while(1)
    {
        displayed(digit);
        if (!RA0)    //Check if switch SW1 is closed
        {
            __delay_ms(100);
        }
        if (!RA1)    //Check if switch SW1 is closed
        {
            __delay_ms(100);
        }
        if (digit==99)
        {
            digit=0;
        }
        digit++;
        if (digit==100)
        {
            digit=0;
        }
        __delay_ms(10); //wait for 100ms
    }
}

void displayed(unsigned int digit)
{
    unsigned int one,ten;
    ten = (digit)/10;
    one = (digit)%10;
    RC0 = 0;
    PORTB = (SEGMENT_MAP[ten]);
    __delay_ms(10);
    RC0 = 1;
    RC1 = 0;
    PORTB = (SEGMENT_MAP[one]);
    __delay_ms(10);
    RC1 = 1;
}
```

2. Đếm từ 0-99 sử dụng Timer0(99-0)



Code

```
#define _XTAL_FREQ 8000000
```

```
#include <xc.h>
```

```
// CONFIG
```

```
__CONFIG(FOSC_HS & WDTE_OFF &
PWRTE_OFF & CP_OFF & BOREN_ON &
LVP_OFF & CPD_OFF & WRT_OFF &
DEBUG_OFF);
```

```
unsigned int Count = 0;
```

```
unsigned int dem = 0, chuc, donvi;
```

```
const unsigned char maled[] = {0x3F, 0x06, 0x5B,
0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F};
```

```
void display(void);
```

```
void xuong(){
```

```
    if(TMR0IF == 1)
```

```
    {
```

```
        TMR0IF = 0;
```

```
        TMR0 = 55;
```

```
        Count++;
```

```
        if(Count == 15)
```

```
        {
```

```
            Count = 0;
```

```
            dem--; } }
```

```
void len(){
```

```
    if(TMR0IF == 1)
```

```
    {
```

```
        TMR0IF = 0;
```

```
        TMR0 = 55;
```

```
        Count++;
```

```
        if(Count == 15)
```

```
        {
```

```
            Count = 0;
```

```
            dem++; } }
```

```
void main(void)
```

```
{
```

```
    TMR0 = 55;
```

```
    OPTION_REG=0x07;
```

```
    TRISB = 0X00;
```

```
    PORTB = 0X00;
```

```
    TRISC = 0X00;
```

```
    while (1)
```

```
    {
```

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

```
            len();
```

```
            display();
```

```
    }
```

```
    for(dem = 99 ; dem>0;){
```

```
        xuong();
```

```
        display();
```

```
    } }
```

```
void display ()
```

```
{
```

```
    chuc = (dem)/10;
```

```
    donvi = dem%10;
```

```
    PORTB = maled[chuc];
```

```
    RC0=0;
```

```
    __delay_ms(5);
```

```
    RC0=1;
```

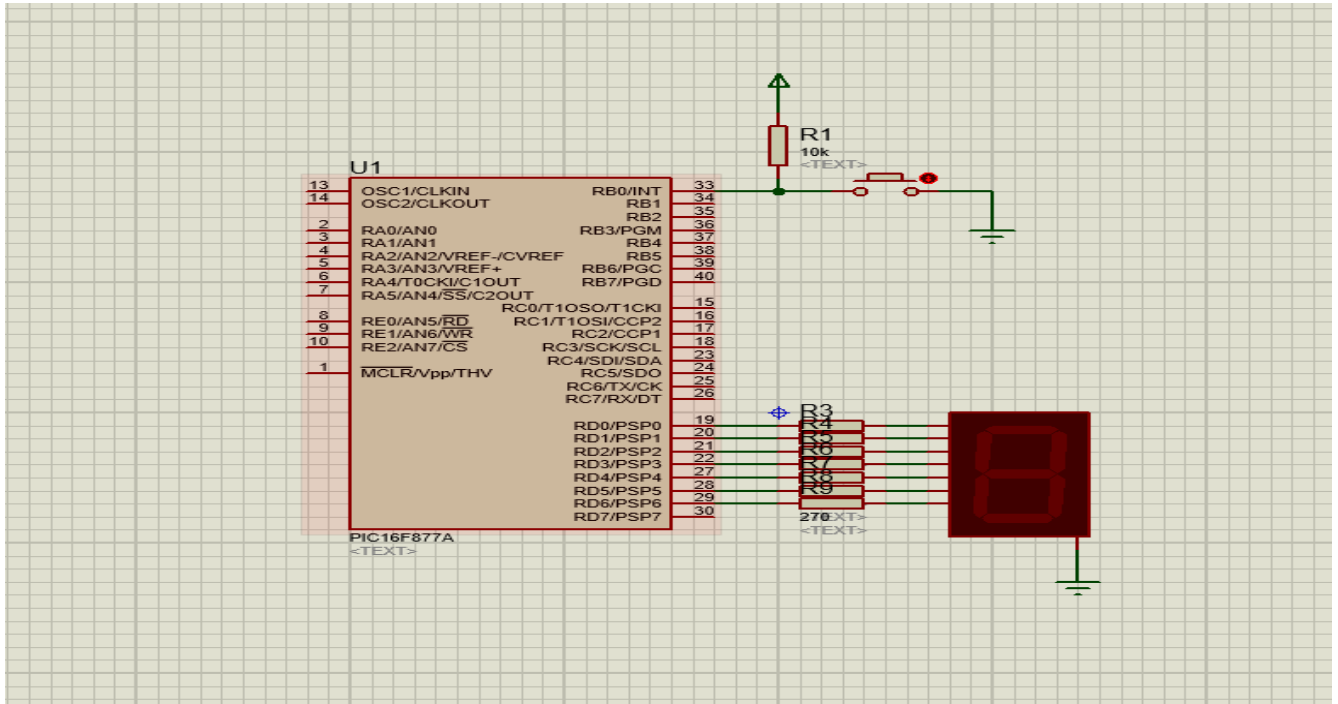
```
    PORTB = maled[donvi];
```

```
    RC1=0;
```

```
    __delay_ms(5);
```

```
    RC1=1;}
```

3. Nút bấm tăng từ 0-9 (ngắt RB0)



Code

```
#define _XTAL_FREQ 8000000
#include <xc.h>

const unsigned char Code7Seg[] =
{0x3F, 0x06, 0x5B, 0x4F, 0x66,
0x6D, 0x7D, 0x07, 0x7F, 0x6F};

unsigned int count=0;
void interrupt ISR(void);
void main (void)
{
    TRISD = 0x00;
    PORTD = 0X00;

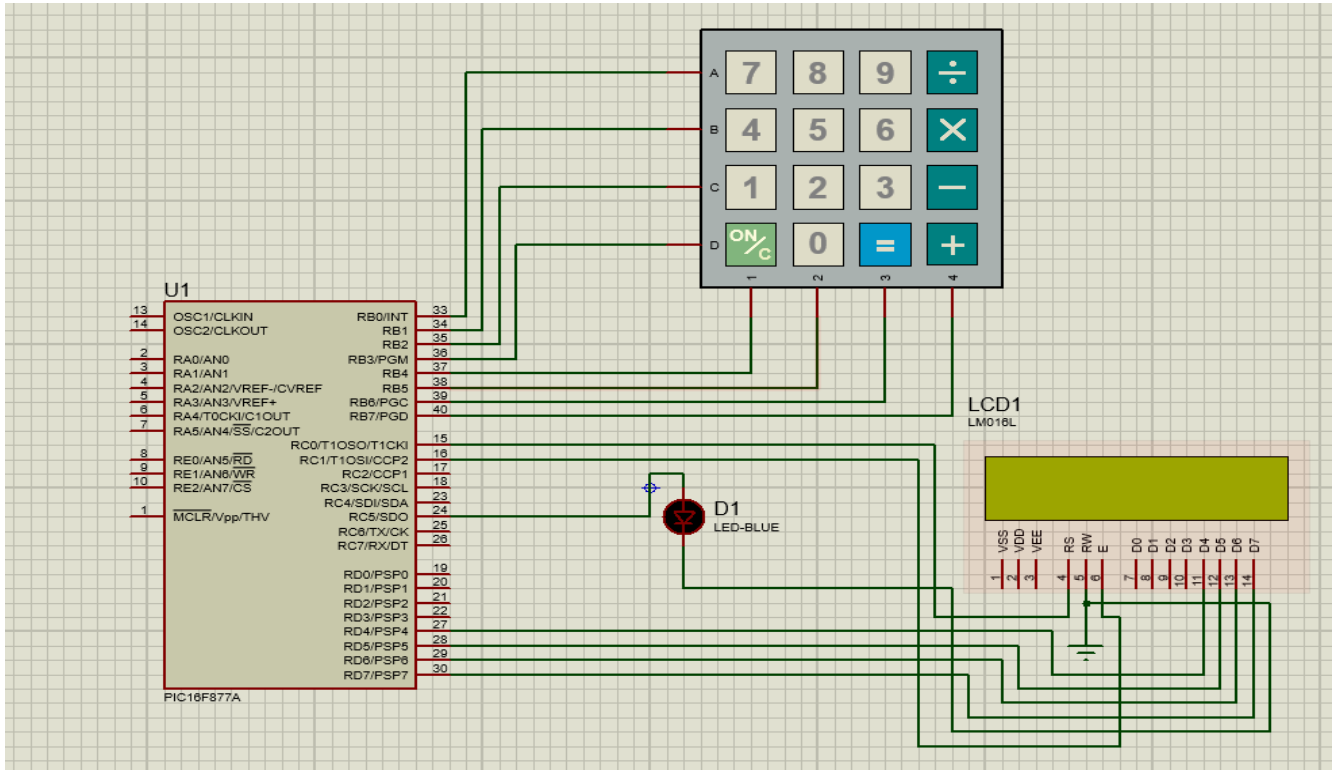
    TRISB0 = 1;
    PORTB = 0X00;
    GIE = 1;
    INTE = 1;
    INTEDG = 1;

    while(1)
    {
        PORTD = Code7Seg[count];
        if(count==10)
        {
            count=0;
        }
    }
}

void interrupt ISR(void)
{
    if(INTF==1)
    {
        count++;
        INTF=0;
    }
}
```


III. LCD

1. Ma trận phím hiển thị LCD (có thêm bóng led)



Code(bài này cũng sử dụng thư viện *lcd4.h* nhưng thay đổi **PORTB** thành **PORTD**. Ở thư viện *lcd4.h* chỉ cần thay đổi lại **PORTB** thành **PORTD** là được.)

```
#include <pic.h>
```

```
#include "key.h"
```

```
#include "lcd4.h"
```

```
__CONFIG(FOSC_HS & WDTE_OFF &  
PWRTE_OFF & CP_OFF & BOREN_ON &  
LVP_OFF & CPD_OFF & WRT_OFF &  
DEBUG_OFF);
```

```
void main() {
```

```
    unsigned int i;
```

```
    TRISD = 0;
```

```
    TRISC0=TRISC1 = 0;
```

```
    TRISC5 = 0;
```

```
    lcd_init();
```

```
    keyinit();
```

```
    unsigned char b;
```

```
    cmd(0x80);
```

```
    show(" Enter the Key ");
```

```
    while (1) {
```

```
        cmd(0xc7);
```

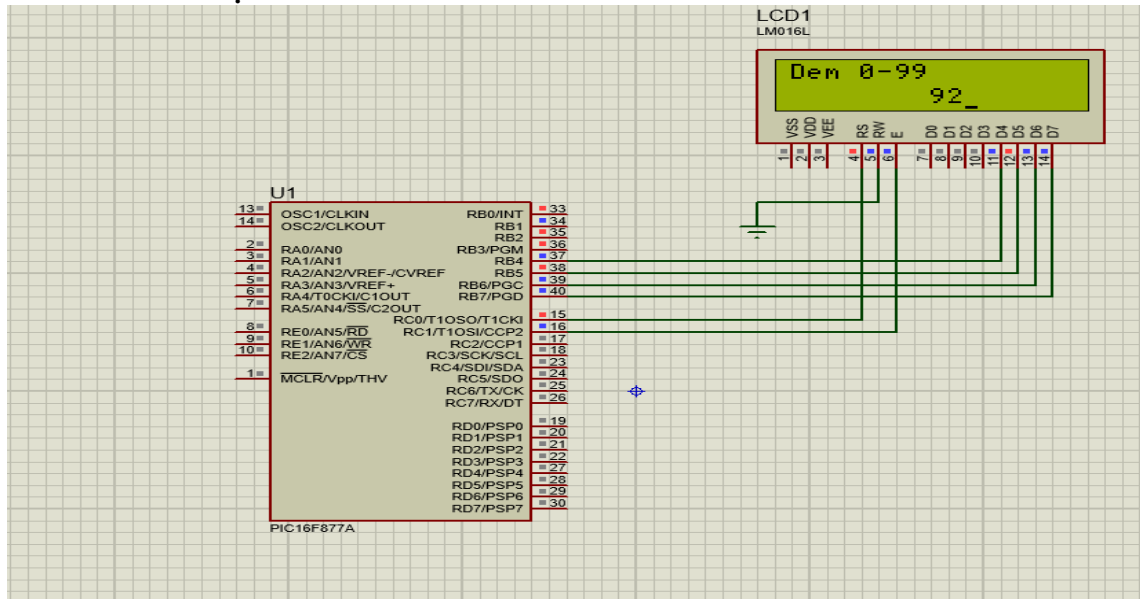
```
        b = key();
```

```
        dat(b);
```

```
    }
```

```
}
```

2. Đếm từ 0-99 hiển thị lcd



code (Để hiển thị số 1 thì cần ghi ra LCD giá trị: 1+48(Mã Ascii của 1)

Một số Compiler hỗ trợ luôn mấy hàm xuất số nguyên rồi, nếu bạn tự viết thì viết theo cách mình đưa ra đây_(Số muốn hiển thị(0-->9) + 48).

```
#include<pic.h>
#include "lcd4.h"

__CONFIG(FOSC_HS &
WDTE_OFF & PWRTE_OFF &
CP_OFF & BOREN_ON &
LVP_OFF & CPD_OFF &
WRT_OFF & DEBUG_OFF);

int a=0;
void delay(unsigned int a){
    unsigned int i;
    for (i=0; i<a ;i++){
        while(!TMR0IF);
        TMR0IF=0;
    }
}
unsigned int i;

void len() {
    for (i = 0; i < 100; i++) {

        cmd(0xc8);
        dat((i / 10) + 48);
        dat((i % 10) + 48);
        delay(21);
    }
}

void xuong() {
    for (i = 99; i > 0; i--) {
        cmd(0xc8);

        dat((i / 10) + 48);
        dat((i % 10) + 48);
        delay(42);
    }
}

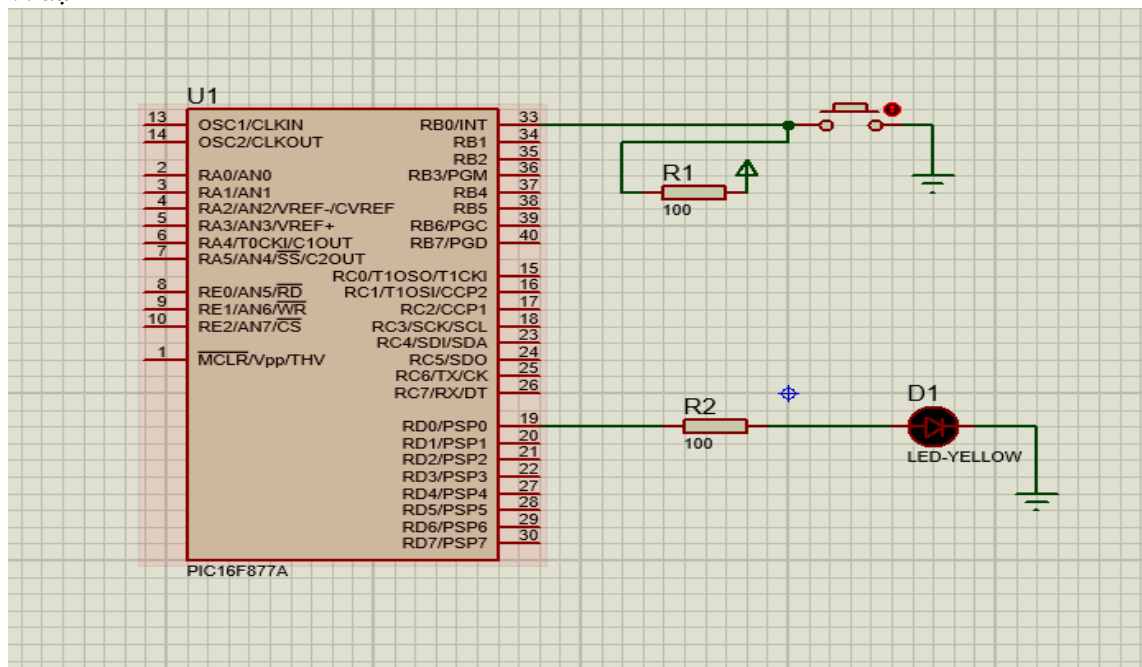
void main() {
    unsigned int i;
    TRISB = TRISC0 =
    TRISC1= 0;
    OPTION_REG=0X07;
    INTCON = 0X0E;
    lcd_init();
    // cmd(0x90);
    show("Dem 0-99");
    while (1) {
        len();
        xuong();
    }
}
```

IV. Timer , Ngắt ngoài , Counter

1. Ngắt RB0

Ngắt RB0: khi có điện áp thay đổi trên chân RB0 thì ngắt xảy ra. Ngoài các bit cần khai báo **GIE=1, INTE=1** ta cần đặt giá trị cho bit liên quan là (**OPTION_REG<6>**). *Khi **INTEDG =1** thì ngắt RB0 xảy ra khi có cạnh lên ở chân RB0, khi **INTEDG = 0** thì ngắt RB0 xảy ra khi có cạnh xuống ở chân RB0.* Khi có cạnh phù hợp với khai báo thì cờ báo ngắt **INTF** được set lên 1 và xảy ra ngắt . Cờ báo ngắt phải được hóa về 0 bằng phần mềm(lệnh) để ngắt tiếp theo có thể được tác động. Nếu không khai báo bit INTEDG thì mặc định ngắt cạnh lên ở chân RB0.

Ví dụ



Code

```
#define _XTAL_FREQ 8000000
#include <xc.h>

__CONFIG(FOSC_HS & WDTE_OFF &
PWRTE_OFF & CP_OFF & BOREN_ON &
LVP_OFF & CPD_OFF & WRT_OFF &
DEBUG_OFF);
```

```
void interrupt ISR(void);
```

```
void main (void)
```

```
{
```

```
    TRISD0= 0;
```

```
    RD0 = 0;
```

```
    TRISB0 = 1;
```

```
    PORTB = 0X00;
```

```
    GIE = 1;
```

```
    INTE = 1;
```

```
    INTEDG = 1;
```

```
    while(1)
```

```
    {
```

```
    }
```

```
}
```

```
void interrupt ISR(void)
```

```
{
    if(INTF==1)
    {
        {
            RD0=!RD0;
            INTF=0;
        }
    }
}
```

2. Timer 0

(8bit) $256 = \text{giá trị nạp} + (\text{giá trị mong muốn}/(\text{tỉ lệ} * \text{chu kì lệnh}))$

Chu kì lệnh = $4 * \text{chu kì máy}$

Chu kì máy = $1/\text{tần số của vdk}$

☐ Tỉ lệ chia tần số là một trong những từ sau: tương ứng với tỉ lệ chia

Ví dụ:

❖ Ví dụ sử dụng thạch anh 12 MHz, timer 0 (8 bit), bộ chia tần 4, trễ 200us thì ta tính theo công thức sau :

$$256 - \frac{200us}{4 \cdot \frac{4}{12MHz}} = 106$$

Giải thích các số trên như sau :

- 256 vì đây là Timer 8 bit.

- 200 : đếm 200us

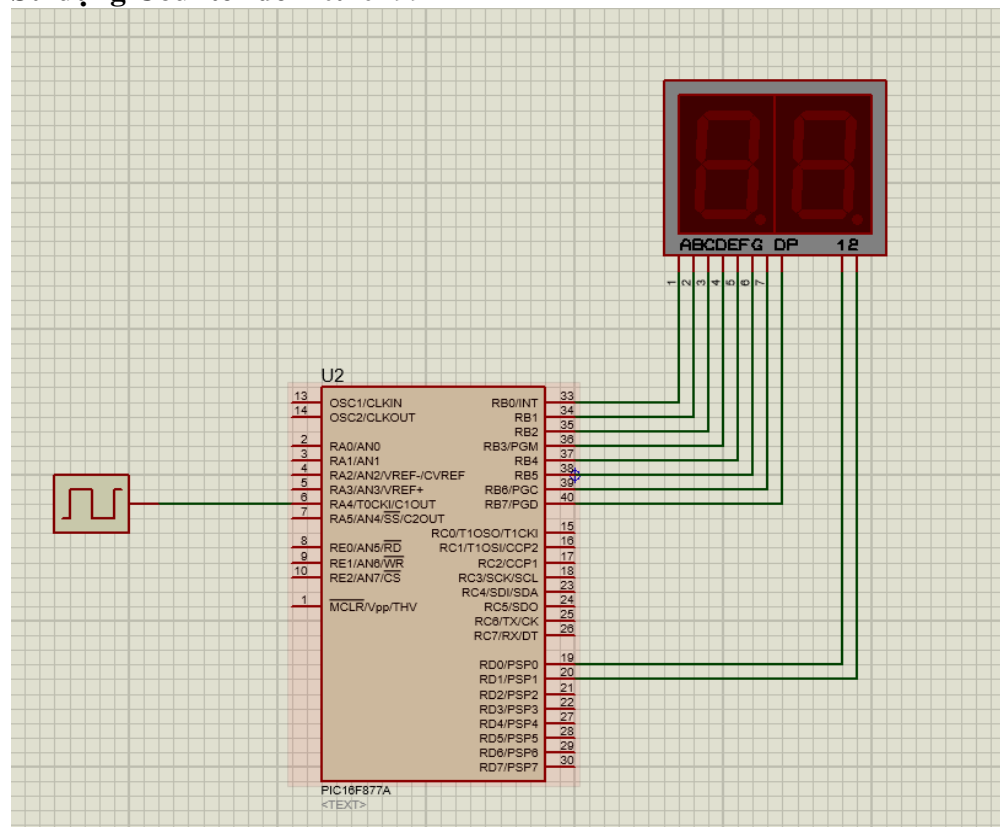
- 4 : bộ chia tần 4 = tỷ lệ 1:4 nghĩa là 1 chu kỳ máy thì mới tăng bộ đếm 1 đv

- 12MHz: thạch anh

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

Ví dụ đã có bài: **Đếm từ 0-99 sử dụng Timer0(99-0)**

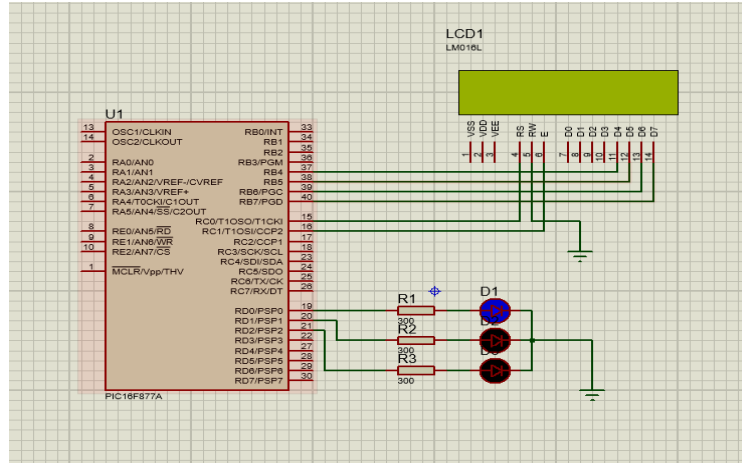
3. Sử dụng Counter đếm từ 0 -99



Code

<pre> #define _XTAL_FREQ 8000000 #include <xc.h> // CONFIG __CONFIG(FOSC_HS & WDTL_OFF & PWRTE_OFF & CP_OFF & BOREN_ON & LVP_OFF & CPD_OFF & WRT_OFF & DEBUG_OFF); unsigned int dem = 0,nghin,tram,chuc,donvi; const unsigned char maled[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F}; void display(void); void main(void) { OPTION_REGbits.T0CS = 1; // Dem xung ngoai </pre>	<pre> OPTION_REGbits.PSA = 1; // Su dung bo chia truoc WDT TRISB = 0; PORTB = 0; TRISDbits.TRISD0 = 0; TRISDbits.TRISD1 = 0; PORTD = 0; while (1) { display(); dem=TMR0; if(dem>=100) { TMR0=0; dem=0; } } </pre>	<pre> } void display (void) { chuc = dem/10; donvi = dem%10; PORTB = maled[chuc]; RD0=0; __delay_ms(10); RD0=1; PORTB = maled[donvi]; RD1=0; __delay_ms(10); RD1=1; } </pre>
---	--	---

Timer 0, timer1,timer2



Code

```
#include<pic.h>
#include"lcd4.h"
#include "timers.h"
#define delay for(z=0;z<=50000;z++)

__CONFIG(FOSC_HS & WDTE_OFF
& PWRTE_OFF & CP_OFF &
BOREN_ON & LVP_OFF & CPD_OFF
& WRT_OFF & DEBUG_OFF); //
unsigned int z;

int a=0,b=0,c=0;

void interrupt tmr0()
{
if(TMR0IF) {
a++;
if(a==42)
//1sec
{
cmd(0x80);
show("Timer 0 interrupt");
RD0= 1;//
RD1= 0;
RD2= 0;
a=0;
delay;
}
TMR0IF=0;
} else if(TMR1IF) {
b++;
if(b==84)
//2secs
{
cmd(0xc0);
show("Timer 1 interrupt");
RD0 = 0;
RD1 = 1;
b=0;
delay;
} else if(TMR2IF) {
c++;
if(c==2025)
//3secs
{
cmd(0x80);
show("Timer 2 interrupt");
RD0 = 0;
RD1 = 0;
RD2 = 1;
c=0;
delay;
}
}
}

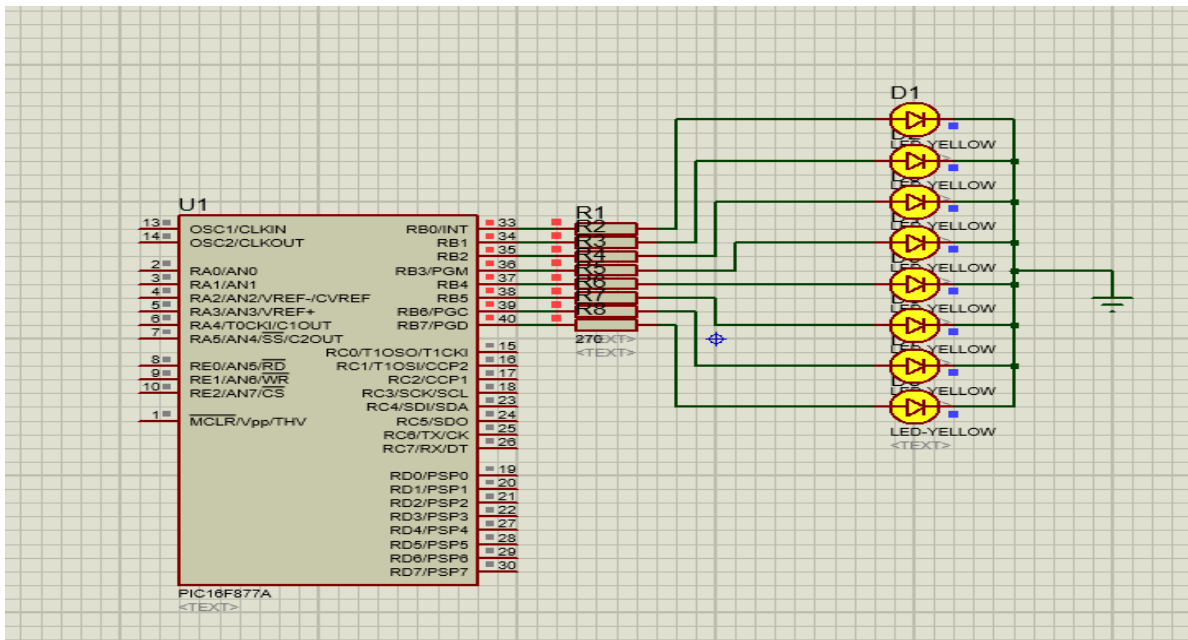
void intr_init()
{
INTCON=0xe0;
PIE1=0x03;
}

void timer_init()
{
OPTION_REG=0b00000111;
//internal clk,rising
edge,prescaler with tim0,256
T1CON=0b00000001;
//prescale=1,oscillator is
off,internal clk,timer on
T2CON=0b01111100;
//postscale=16,prescale=1,timer
on
}

void main()
{
//TRISD0 = TRISD1 = TRISD2 =
```

HIỆU ỨNG LED

1. Hiệu ứng Xen kẻ, nửa sáng nửa tắt,sáng hết tắt hết



```
Code
#include <stdio.h>

#include <stdlib.h>

#define _XTAL_FREQ 4000000

#include <xc.h>

// CONFIG
__CONFIG(FOSC_HS & WDTE_OFF &
PWRTE_OFF & CP_OFF & BOREN_ON &
LVP_OFF & CPD_OFF & WRT_OFF &
DEBUG_OFF); //

void GPIO_init (void);

void SANGHET (void);

void TATHET (void);

void XENKE (void);

void SANGNUA (void);

void main (void)
{
    GPIO_init();

    while(1)
    {
        SANGHET();

        __delay_ms(500);
```

```
TATHET();

__delay_ms(500);

XENKE();

__delay_ms(500);

SANGNUA();

__delay_ms(500);
}

void GPIO_init(void)
{
    TRISB = 0X00;

    PORTB = 0X00;
}

void SANGHET (void)
{
    PORTB = 0XFF;
}

void TATHET (void)
{
    PORTB = 0X00;
}

void XENKE (void)
```

```
{
    PORTB = 0XAA;

    __delay_ms(500);

    PORTB = 0X55;
}

void SANGNUA (void)
{
    PORTB = 0X0F;

    __delay_ms(500);

    PORTB = 0XF0;
}
```

2. Đuổi LED (Chaser LED), sáng dồn, dịch trái, dịch phải, sáng dần, tắt dần(hình giống bài trên)

Để dịch phải thì ta phải cho giá trị ban đầu của PORTB là :

```
PORTB = 0X01;
```

Sau đó chúng ta sử dụng lệnh DỊCH PHẢI để dịch LED :

```
PORTB = PORTB <<1;
```

Lệnh này chỉ dịch được 1 bit nên nếu muốn dịch nhiều bit hơn thì ta phải sử dụng vòng lặp FOR để dịch nhiều lần. Trong bài này mình dịch 8bit nên code sẽ viết như sau :

```
For(i=0;i<8;i++)
{
PORTB = PORTB<<1;
}
```

Như vậy thì ban đầu i=0 sau đó tăng I lên và mỗi lần tăng lên thì PORTB sẽ dịch qua 1 bit.

Tương tự lệnh DỊCH TRÁI thì chỉ cần sử dụng lệnh như sau :

```
For(i=0;i<8;i++)
{
PORTB = PORTB>>1;
}
```

Từ hai lệnh trên chúng ta có thể dịch 2 hoặc 3 led 1 lần, chỉ cần thay đổi giá trị ban đầu cho PORTB và tăng số bit cần dịch là được. Ví dụ dịch 1 lần 2 LED qua phải thì giá trị ban đầu cần nạp cho PORTB là PORTB = 0X03 và dịch qua 2 bit thì PORTB = PORTB<<2.

Code

```
#include <xc.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define _XTAL_FREQ 8000000
// CONFIG
__CONFIG(FOSC_HS & WDTE_OFF &
PWRTE_OFF & CP_OFF & BOREN_ON &
LVP_OFF & CPD_OFF & WRT_OFF &
DEBUG_OFF); //
```

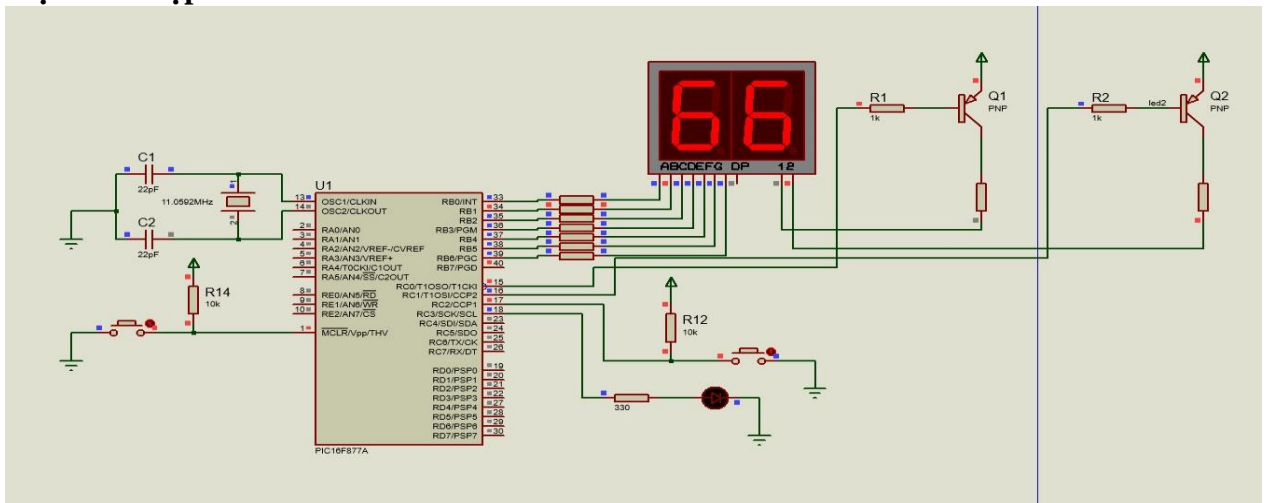
```
void GPIO_init(void);// gọi ham xuất nhập
void main (void)
```

```
{
    GPIO_init(); //
    unsigned char i,I,J,CD,D,Y,X;
    while(1)
    {
        //left dịch trái
        PORTB = 0X01;
        for(i=0;i<8;i++)
        {
            __delay_ms(300);
            PORTB = PORTB<<1;
        }
        // right dịch phải
        __delay_ms(300);
        PORTB = 0X80;
        for(i=0;i<8;i++)
        {
            __delay_ms(300);
            PORTB = PORTB>>1;
        }
        // left 2 dịch trái 2 led
        PORTB = 0X03;
        for(i=0;i<4;i++)
        {
            __delay_ms(300);
            PORTB = PORTB<<2;
        }
        // right 2 dịch phải hai led
        PORTB = 0XC0;
        for(i=0;i<4;i++)
        {
            __delay_ms(300);
            PORTB = PORTB>>2;
        }
        // chase left sáng dần
        __delay_ms(300);
        PORTB = 0X00;
        for(i=0;i<9;i++)
```

```
{
    __delay_ms(300);
    PORTB = PORTB>>1;
    PORTB = PORTB | 0X80;
}
// chase right sáng dần phải
__delay_ms(300);
PORTB = 0X00;
for(i=0;i<9;i++)
{
    __delay_ms(300);
    PORTB = PORTB<<1;
    PORTB = PORTB | 1;
}
//chase on left sáng dần tắt dần trái
__delay_ms(300);
PORTB = 0X00;
for(i=0;i<9;i++)
{
    __delay_ms(300);
    PORTB = (PORTB<<1) | 0X01;
}
// chase off left sáng dần tắt dần trái
__delay_ms(300);
for(i=0;i<9;i++)
{
    __delay_ms(300);
    PORTB = PORTB<<1;
}
// chase right on and off sáng dần tắt dần phải
__delay_ms(300);
X=0X00000000;
for(I=0;I<8;I++)
{
    X=(X<<1)+0X01;
    PORTB = X;
    __delay_ms(300);
}
for(I=0;I<8;I++)
{
    X=(X>>1);
    PORTB = X;
    __delay_ms(300);
}
// don 1 sáng dồn 1
__delay_ms(300);
CD=0X00000000;
for ( J=8;J>0;J--)
{
    D=0X00000001;
```

```
for(I=0;I<J;I++)
{
    Y=CD+D;
    PORTB = (Y);
    __delay_ms(300);
    D=(D<<1);
}
CD=Y;
}
// don 2 sáng dồn 2 led
__delay_ms(300);
CD=0X00;
for ( J=4;J>0;J--)
{
    __delay_ms(300);
    D=0X03//0x03
    for(I=0;I<J;I++)
    {
        __delay_ms(100);
        Y=CD+D;
        PORTB = (Y);
        __delay_ms(300);
        D=(D<<2);
    }
    CD=Y;
}
}
void GPIO_init(void)
{
    TRISB = 0X00; // cho porta la output
    PORTB = 0X00; //
}
```


Một số bài tập khác



Code

```
#include<pic.h>
#include<htc.h>
#define led1 RC0
#define led2 RC1
#define sang 0
#define tat 1
#define sw RC2
#define leddon RC3
__CONFIG( FOSC_HS & WDTE_OFF &
PWRTE_OFF & CP_OFF & BOREN_ON &
LVP_OFF & CPD_OFF & WRT_OFF &
DEBUG_OFF);
void delay_ms(unsigned t) // 1 ms
{
    while(t--){
        TMR0=83;
        TMR0IF=0;
        while(!TMR0IF);
    }
}
unsigned char
so[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,
0xf8,0x80,0x90};
int demlen,demlui;
unsigned char chuc,donvi,chuc1,donvi1;
char i,b;
void main()
{
    OPTION_REG=0x03;
    TRISB=0;
    TRISC0=TRISC1=TRISC3=0;
```

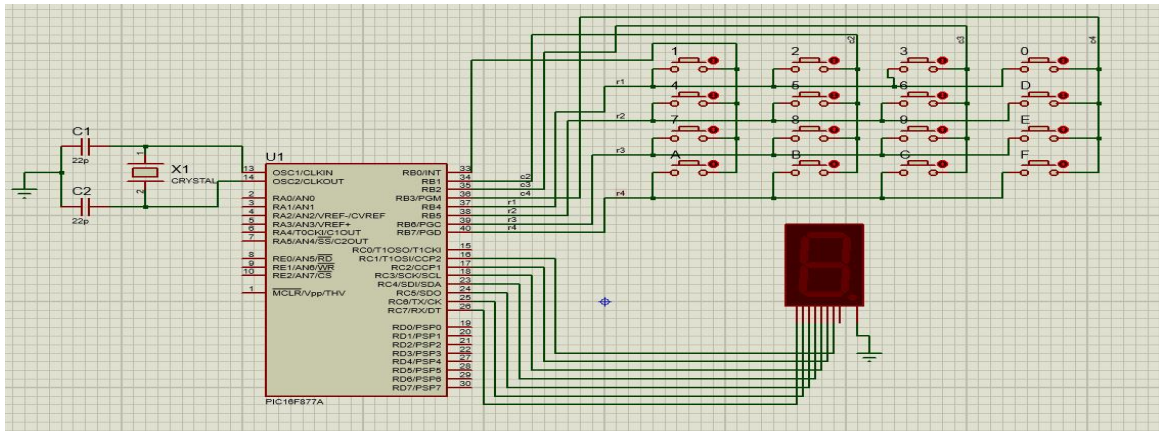
```
led1=led2=0;
TRISC2=1;
leddon=0;

while(1)
{
    for(demlen=50;demlen<=70;demlen++){
        chuc =demlen/10;
        donvi=demlen%10;
        for(int c=0;c<1;c++){
            if(!sw){
                delay_ms(0);
                leddon=1;
            }
            else leddon=0;
        }
        for(i=0;i<25;i++)
        {
            led1=sang;
            PORTB=so[chuc];
            delay_ms(10);
            led1=tat;

            led2=sang;
            PORTB=so[donvi];
            delay_ms(10);
            led2=tat;
```

```
led2=tat;
        }
    }
    for(demlui=70;demlui>=50;--demlui)
    {
        chuc1=demlui/10;
        donvi1=demlui%10;
        for(int c=0;c<1;c++){
            if(!sw){
                delay_ms(0);
                leddon=1;
            }
            else leddon=0;
        }
        for(b=0;b<50;b++)
        {
            led1=sang;
            PORTB=so[chuc1];
            delay_ms(10);
            led1=tat;

            led2=sang;
            PORTB=so[donvi1];
            delay_ms(10);
            led2=tat;
        }
    }
}
```



```
#include <pic.h>
#include <htc.h>
#define FREQ_OSC 11059200u1
void delay_ms(int x)
{
    int i,j;
    for(i = 0; i < x; i++)
    {
        for(j = 0 ; j < 123 ; j++)
        {
        }
    }
}

#define r4 RB7
#define r3 RB6
#define r2 RB5
#define r1 RB4
#define c4 RB3
#define c3 RB2
#define c2 RB1
#define c1 RB0
#define led PORTC

int returnKeyPad() //khi bam mot nut tra ve mot gia tri tuong ung
{
    if(r1==1 && r2==1 && r3==1 && r4==1)
        return 0;
    if(r1==0 && r2==1 && r3==1 && r4==1)
    {
        if(c1==0)
            return 1;
        if(c2==0)
            return 2;
        if(c3==0)
            return 3;
        if(c4==0)
            return 13;
    }
    if(r2==0 && r1==1 && r3==1 && r4==1)
    {
        if(c1==0)
            return 4;
        if(c2==0)
            return 5;
        if(c3==0)
            return 6;
        if(c4==0)
            return 14;
    }
    if(r1==1 && r2==1 && r3==0 && r4==1)
    {
        if(c1==0)
            return 7;
        if(c2==0)
            return 8;
        if(c3==0)
```

```
        return 9;
        if(c4==0)
            return 15;
    }
    if(r1==1 && r2==1 && r3==1 && r4==0)
    {
        if(c1==0)
            return 10;
        if(c2==0)
            return 11;
        if(c3==0)
            return 12;
        if(c4==0)
            return 16;
    }
}

void LED(int x) //voi moi gia tri tra ve thi co mot muc sang tuong ung
{
    switch(x)
    {
        case 1:
            led=0x60;
            while(r1==0);
            break;
        case 2:
            led= 0xda;
            while(r1==0);
            break;
        case 3:
            led= 0xf2;
            while(r1==0);
            break;
        case 4:
            led= 0x66;
            while(r2==0);
            break;
        case 5:
            led= 0xb6;
            while(r2==0);
            break;
        case 6:
            led= 0xbe;
            while(r2==0);
            break;
        case 7:
            led=0xe0;
            while(r3==0);
            break;
        case 8:
            led=0xfe;
            while(r3==0);
            break;
        case 9:
            led=0xE6;
            while(r3==0);
            break;
```

```

case 13:
    led= 0xfc;
    while(r1==0);
    break;
case 14:
    led= 0x7a;
    while(r2==0);
    break;
case 15:
    led= 0x9e;
    while(r3==0);
    break;
case 16:
    led= 0x8E;
    while(r4==0);
    break;
case 12:
    led=0x9C;
    while(r4==0);
    break;
case 11:
    led= 0x3E;
    while(r4==0);
    break;
case 10:
    led=0xee;
    while(r4==0);
    break;

}
}
void keyinit() {
    TRISB = 0XF0;
    OPTION_REG &= 0X7F; //ENABLE PULL UP
}
void scan(int i)
{
    if(i==0)
    {
        c1=0;
        c2=c3=c4=1;
        delay_ms(10);
    }
    if(i==1)
    {
        c2=0;
        c1=c3=c4=1;
        delay_ms(10);
    }
    if(i==2)
    {
        c3=0;
        c1=c2=c4=1;
        delay_ms(10);
    }
    if(i==3)
    {
        c4=0;
        c1=c2=c3=1;
        delay_ms(10);
    }
}

void main()
{
    TRISB=1;
    TRISC=0;
    keyinit();
    int i =0;

    while(1)
    {
        PORTC=0x02;

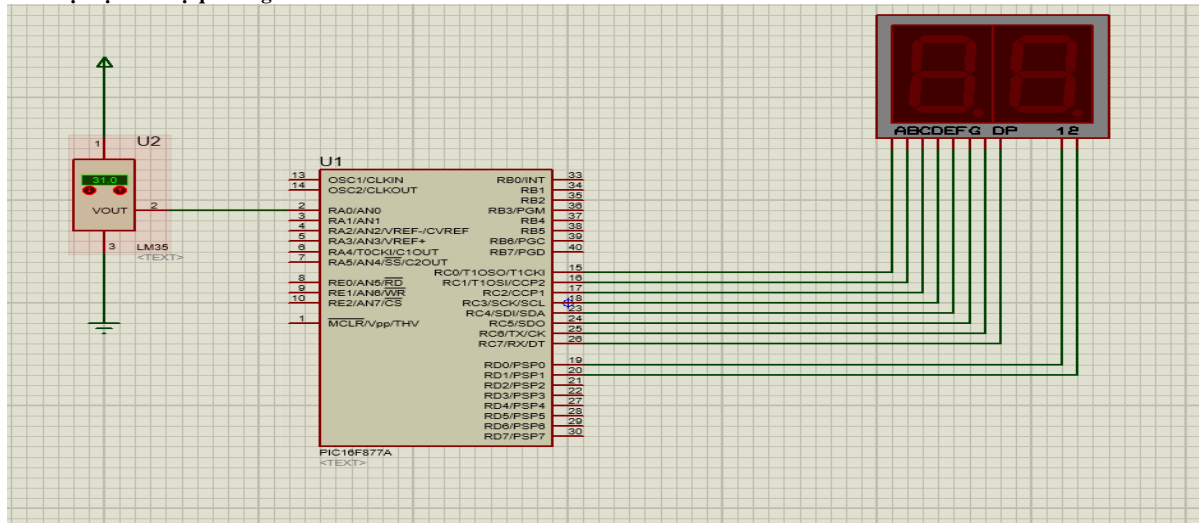
```

```

        scan(i);
        i++;
        if(i>=4)
            i=0;
        LED(returnKeyPad());
    }
}

```

Đo nhiệt độ hiển thị qua 7seg



Code

```
#include <stdio.h>
#include <stdlib.h>
#define _XTAL_FREQ 8000000 // if 4Mhz
is XT
#include <xc.h>
// CONFIG
__CONFIG(FOSC_HS & WDTE_OFF &
PWRTE_OFF & CP_OFF & BOREN_ON &
LVP_OFF & CPD_OFF & WRT_OFF &
DEBUG_OFF);

#define seg_data PORTC
const unsigned char maled7[] =
{0xBF,0X86,0XDB,0XCF,0XE6,0XED,0X
FD,0X87,0XFF,0XEF};
void ADCinit(void);
unsigned int ReadADC(void);
unsigned int ADCvalue = 0;
void msdelay(unsigned int itime);
void delay();
void display_digit(int);
int f;
unsigned int voltage;
void main()
{
    TRISA = 0xff;
    TRISC = 0X00;
    TRISD = 0X00;
    ADCinit();
    unsigned int ones,tens=0;
    while(1)
    {
        ADCvalue = ReadADC();
        voltage = 5000.0f / 1023 * ADCvalue;
        f = voltage / 10;
        ones=f%10;
        tens=(f/10)%10;
        display_digit(tens);
        RD0=1;
        delay();
        RD0=0;
        display_digit(ones);
        RD1=1;
        delay();
        RD1=0;
    }
}

void msdelay(unsigned int itime)
{
    unsigned int i,j;

    for(i=0;i<itime;i++)
    for(j=0;j<1275;j++);
}

void display_digit(int c)
{
    switch(c)
    {
        case 0:
            seg_data=0xC0;
            break;
        case 1:
            seg_data=0xF9;
            break;
        case 2:
            seg_data=0xA4;
            break;
        case 3:
            seg_data=0xB0;
            break;
        case 4:
            seg_data=0x99;
            break;
        case 5:
            seg_data=0x92;
            break;
        case 6:
            seg_data=0x82;
            break;
        case 7:
            seg_data=0xF8;
            break;
        case 8:
            seg_data=0x80;
            break;
        case 9:
            seg_data=0x90;
            break;
    }
}

void delay()
{
    long int k;
    for(k=0;k<100;k++);
}

void ADCinit(void)// adc
{
    // chọn tần số clock cho bộ adc
    ADCON1bits.ADCS2 = 0,
    ADCON0bits.ADCS1 =
    0,ADCON0bits.ADCS0 = 1;

    // chọn kênh adc là kênh an0
    ADCON0bits.CHS2 = 0,
    ADCON0bits.CHS1 = 0,
    ADCON0bits.CHS0 = 0;
    // chọn cách lưu data
    ADCON1bits.ADFM = 1;
    // cấu hình công vào
    ADCON1bits.PCFG3 = 1,
    ADCON1bits.PCFG2 = 1,
    ADCON1bits.PCFG1 = 1,
    ADCON1bits.PCFG0 = 0;
    // cấp nguồn cho khối adc
    ADCON0bits.ADON = 1;
}

unsigned int ReadADC(void)// doc len 7
doan
{
    unsigned int TempValue = 0;
    ADCON0bits.GO_nDONE = 1;
    while(ADCON0bits.GO_nDONE);
    TempValue = ADRESH*256 + ADRESL;
    return (TempValue);
}
```