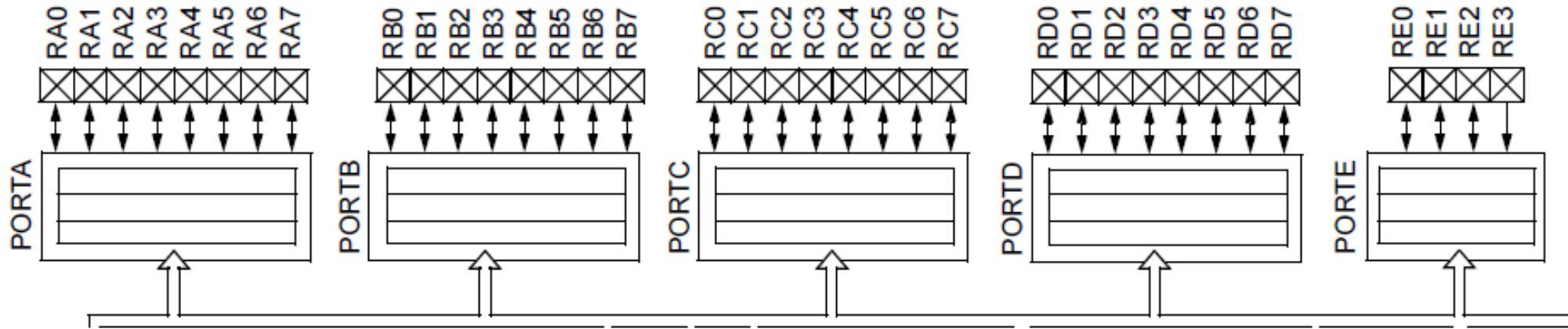


03. Mô đun xuất nhập (I/O Ports)

SƠ ĐỒ CHÂN



- PIC16F887 có **35** chân xuất nhập
- **RE3** có nhiệm vụ reset

THANH GHI PORTA VÀ TRISA

PORTA là 1 port 8 bit, 2 chiều xuất nhập (input/output) và được điều khiển bởi thanh ghi **TRISA**

- Set bit **TRISA** (bit = 1) thì bit tương ứng trên **PORTA** là **Input**
- Clear bit **TRISA** (bit = 0) thì bit tương ứng là **Output**

REGISTER 3-1: PORTA: PORTA REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
bit 7							bit 0

REGISTER 3-2: TRISA: PORTA TRI-STATE REGISTER

R/W-1 ⁽¹⁾	R/W-1 ⁽¹⁾	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
bit 7							bit 0

THANH GHI ANSEL

- Thanh ghi **ANSEL** quy định một chân I/O ở mode input là **Digital** hay **Analog**

REGISTER 3-3: ANSEL: ANALOG SELECT REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANS7 ⁽²⁾	ANS6 ⁽²⁾	ANS5 ⁽²⁾	ANS4	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

bit 7-0

ANS<7:0>: Analog Select bits

Analog select between analog or digital function on pins AN<7:0>, respectively.

1 = Analog input. Pin is assigned as analog input⁽¹⁾.

0 = Digital I/O. Pin is assigned to port or special function.

RA0/AN0/ULPWU/C12INO-

- a general purpose I/O
- an analog input for the ADC
- a negative analog input to Comparator C1 or C2
- an analog input for the Ultra Low-Power Wake-up

RA1/AN1/C12IN1-

- a general purpose I/O
- an analog input for the ADC
- a negative analog input to Comparator C1 or C2

RA2/AN2/VREF-/CVREF/C2IN+

- a general purpose I/O
- an analog input for the ADC
- a negative voltage reference input for the ADC and CVREF
- a comparator voltage reference output
- a positive analog input to Comparator C2

RA3/AN3/VREF+/C1IN+

- a general purpose input
- an analog input for the ADC
- a positive voltage reference input for the ADC and CVREF
- a positive analog input to Comparator C1

RA4/T0CKI/C1OUT

- a general purpose I/O
- a clock input for Timer0
- a digital output from Comparator C1

RA5/AN4/SS/C2OUT

- a general purpose I/O
- an analog input for the ADC
- a slave select input
- a digital output from Comparator C2

RA6/OSC2/CLKOUT

- a general purpose I/O
- a crystal/resonator connection
- a clock output

RA7/OSC1/CLKIN

- a general purpose I/O
- a crystal/resonator connection
- a clock input

THANH GHI PORTB VÀ TRISB

- Điều khiển xuất nhập **PORTB** là **Input** hay **Output**

REGISTER 3-5: PORTB: PORTB REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
bit 7							bit 0

REGISTER 3-6: TRISB: PORTB TRI-STATE REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
bit 7							bit 0

THANH GHI ANSELH

- Dùng để cấu hình chân input trên **PORTB** là **Analog** hay **Digital**
- Đây là các bit cao của module Analog

REGISTER 3-4: **ANSELH: ANALOG SELECT HIGH REGISTER**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	
—	—	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	
bit 7								bit 0

THANH GHI WEAK PULL-UP PORTB

- Cấu hình mỗi chân của **PORTB** có dạng điện trở kéo lên nội

REGISTER 3-7: WPUB: WEAK PULL-UP PORTB REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0
bit 7							bit 0

THANH GHI INTERRUPT-ON-CHANGE PORTB

- Cấu hình mỗi chân của **PORTB** là chân ngắt hay không

REGISTER 3-8: IOCB: INTERRUPT-ON-CHANGE PORTB REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0
bit 7							bit 0

THANH GHI PORTC VÀ TRISC

- Điều khiển xuất nhập **PORTC** là **Input** hay **Output**

REGISTER 3-9: **PORTC: PORTC REGISTER**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
bit 7							bit 0

REGISTER 3-10: **TRISC: PORTC TRI-STATE REGISTER**

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1 ⁽¹⁾	R/W-1 ⁽¹⁾
TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
bit 7							bit 0

THANH GHI PORTD VÀ TRISD

- Điều khiển xuất nhập **PORTD** là **Input** hay **Output**

REGISTER 3-11: PORTD: PORTD REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
bit 7							bit 0

REGISTER 3-12: TRISD: PORTD TRI-STATE REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0
bit 7							bit 0

THANH GHI PORTE VÀ TRISE

- Điều khiển xuất nhập **PORTE** là **Input** hay **Output**

REGISTER 3-13: PORTE: PORTE REGISTER

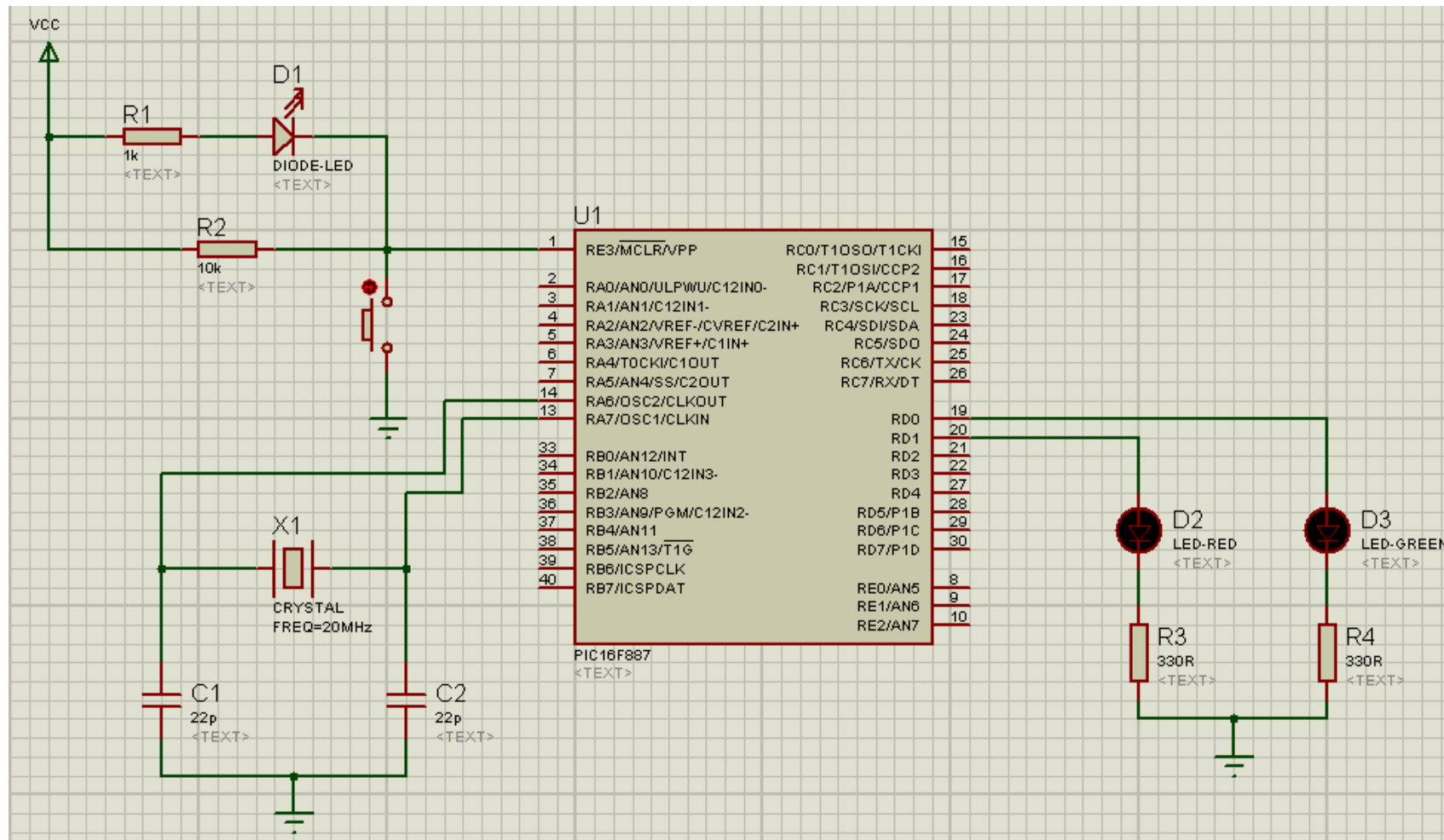
U-0	U-0	U-0	U-0	R-x	R/W-x	R/W-x	R/W-x
—	—	—	—	RE3	RE2	RE1	RE0
bit 7				bit 0			

REGISTER 3-14: TRISE: PORTE TRI-STATE REGISTER

U-0	U-0	U-0	U-0	R-1 ⁽¹⁾	R/W-1	R/W-1	R/W-1
—	—	—	—	TRISE3	TRISE2	TRISE1	TRISE0
bit 7				bit 0			

VÍ DỤ 1: XUẤT LED (1)

Mạch điện trong phần mềm **Proteus**



VÍ DỤ 1: XUẤT LED (2)

Viết code **CCS** làm sáng đèn **D2** trong **1** giây rồi tắt, đến lượt đèn **D3** sáng **1** giây rồi tắt, cứ như thế lập đi lập lại

```
#include <16F887.h>
#use delay (clock=20000000)
#fuses HS,NOWDT,PUT,NOLVP,NOPROTECT,BROWNOUT
#byte PORTD=0x08

void main()
{
    set_tris_d(0);
    while(TRUE)
    {
        OUTPUT_D(0b00000001);
        delay_ms(1000);
        OUTPUT_D(0b00000010);
        delay_ms(1000);
    }
}
```

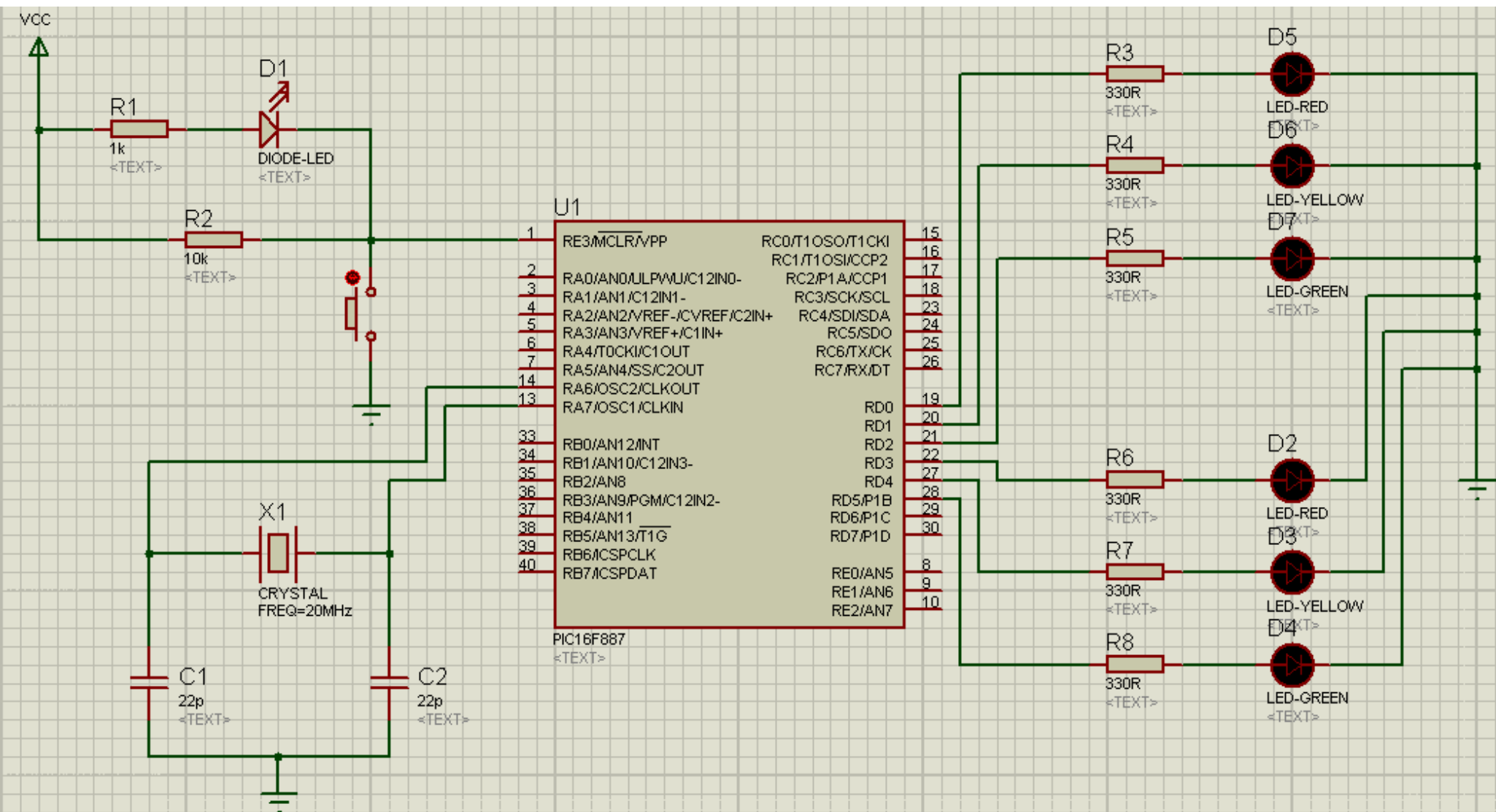
MỘT SỐ LỆNH CƠ BẢN TRONG CCS-C

- **#include < >** or **#include ' '** // Declare the mcu
- **byte id = x** // Ex: byte PortB = 0x06;
- **define id text** // Ex: define a 1234567
- **#use 'function name'(parameters)** // #use delay (clock = 20M)
- **delay_cycles(a)** // a: instruction cycle
- **delay_us(a)**
- **delay_ms(a)**
- **output_low(pin)**
- **output_high(pin)**
- **Output_X(value)** // output 1 byte to X port on MCU
- **input(pin)**
- **input_X(value)** // input from X port on MCU
- **Set_tris_X(value)** // X: name of port on MCU

VÍ DỤ 2: ĐÈN GIAO THÔNG (1)

- Thiết kế trong **Proteus** mạch điều khiển đèn giao thông tại một ngã tư trong thành phố?
- Viết code **CCS** điều khiển hệ thống đèn giao thông trên: đèn xanh sáng **25** giây, đèn vàng **5** giây, đèn đỏ **30** giây?

VÍ DỤ 2: ĐÈN GIAO THÔNG (2)



VÍ DỤ 2: ĐÈN GIAO THÔNG (3)

```
#include <16F887.h>
#fuses HS, NOWDT, NOPROTECT, NOLVP, PUT, BROWNOUT
#use delay(clock=20000000)
#byte PORTD=0x08

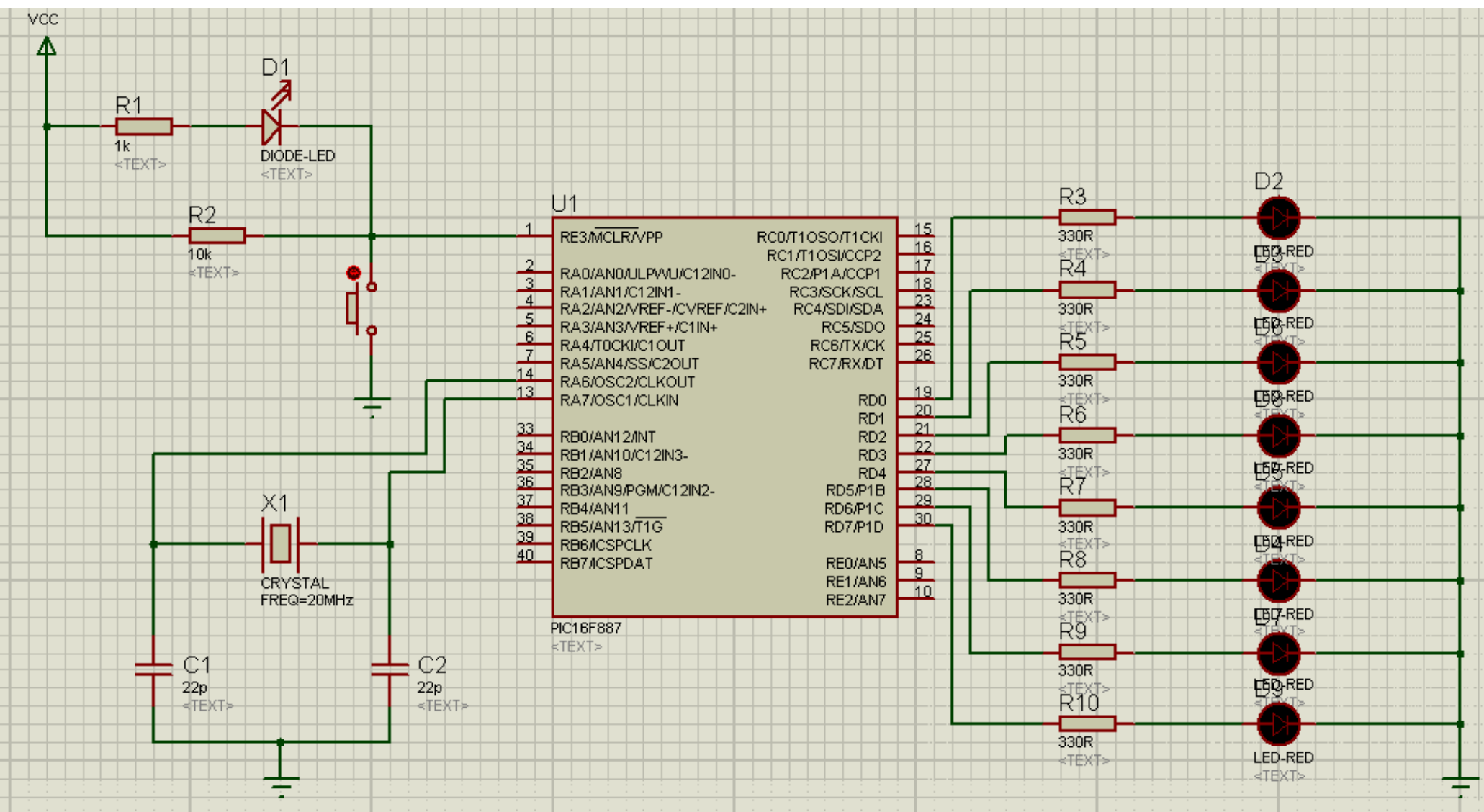
void main()
{
    SET_TRIS_D(0);

    while(TRUE)
    {
        OUTPUT_D(0b00100001);
        delay_ms(25000);
        OUTPUT_D(0b00010001);
        delay_ms(5000);
        OUTPUT_D(0b00001100);
        delay_ms(25000);
        OUTPUT_D(0b00001010);
        delay_ms(5000);
    }
}
```

VÍ DỤ 3: DỊCH LED (1)

- Thiết kế trong **Proteus** mạch điện sử dụng **8** đèn Led nối vào 1 Port của VĐK?
 - Viết code **CCS** làm sáng đèn **D1** trong **1** giây rồi tắt, đến lượt đèn **D2** sáng **1** giây rồi tắt, cứ như thế cho đến đèn **D8** sáng **1** giây rồi tắt, sau đó quay lại **D1**?
1. Sử dụng các lệnh xuất LED thông thường
 2. Sử dụng chương trình con với hàm “**a<<1**”

VÍ DỤ 3: DỊCH LED (2)



VÍ DỤ 3: DỊCH LED (3)

```
#include <16F887.h>
#use delay (clock=20000000)
#fuses HS,NOWDT,PUT,NOLVP,NOPROTECT,BROWNOUT
#byte PORTD=0x08
```

```
void led()
{
    int a,i;
    a = 0x01;
    for(i=1;i<=8;i++)
    {
        OUTPUT_D(a);
        a = a<<1;
        delay_ms(1000);
    }
}
```

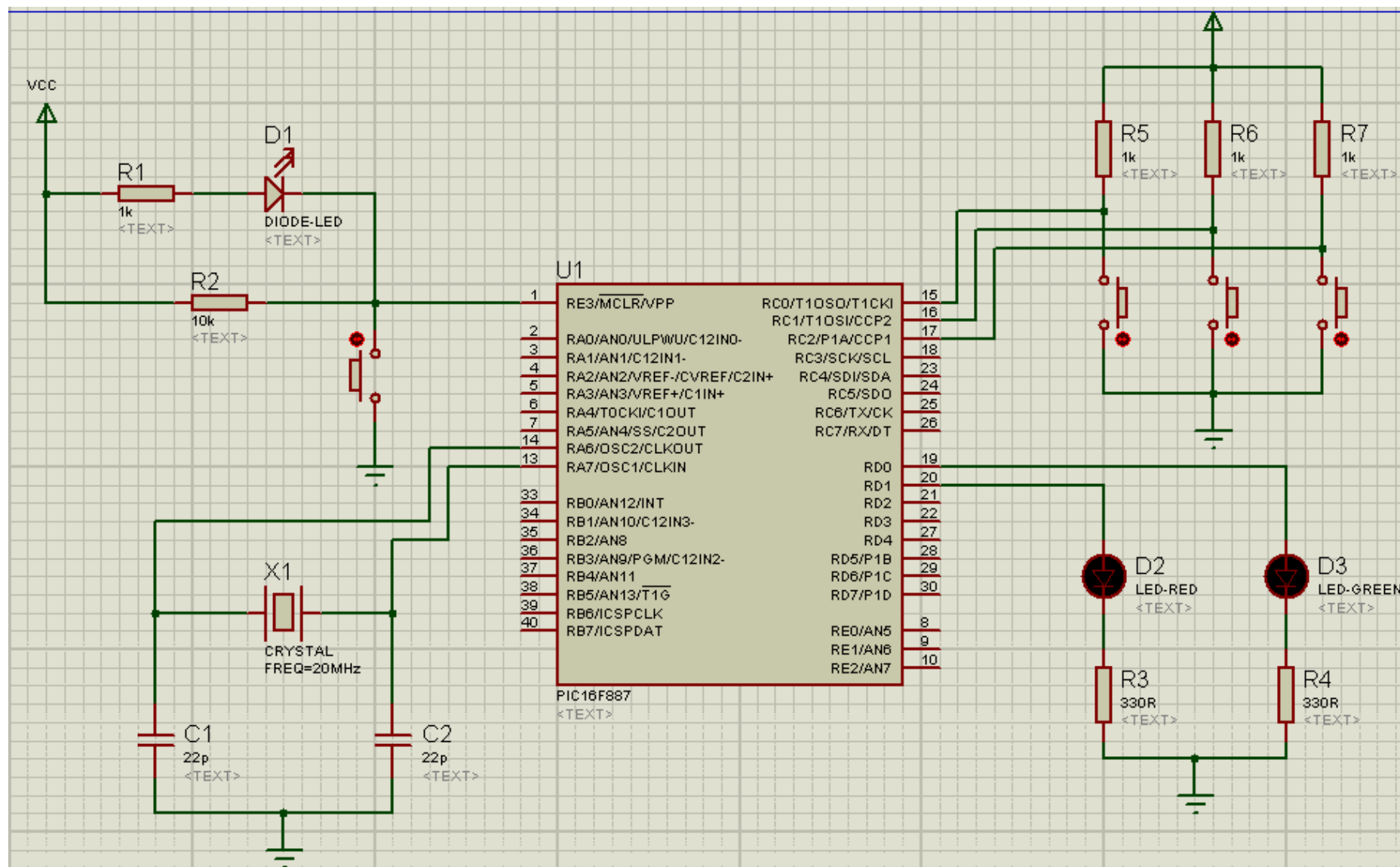
```
void main()
{
    SET_TRIS_D(0);
    while(TRUE)
    {
        led();
    }
}
```

VÍ DỤ 4: SỬ DỤNG NÚT NHẤN (1)

Lấy lại hình mạch điện trong ví dụ 1, thiết kế thêm 3 nút nhấn. Viết code CCS sao cho:

- Nhấn nút **SW1** thì đèn **LED-RED** sáng
- Nhấn nút **SW2** thì đèn **LED-GREEN** sáng
- Nhấn nút **SW3** thì cả **2** đèn sáng

VÍ DỤ 4: SỬ DỤNG NÚT NHẤN (2)



VÍ DỤ 4: SỬ DỤNG NÚT NHẤN (3)

```
#include <16F887.h>
#use delay (clock=20000000)
#fuses HS,NOWDT,PUT,NOLVP,NOPROTECT,BROWNOUT
#byte PORTD=0x08

void main()
{
    SET_TRIS_D(0);
    SET_TRIS_C(255);
    PORTD = 0;

    while(TRUE)
    {
        if (INPUT(PIN_C0)==0)
        {
            PORTD = 2;
        }

        if (INPUT(PIN_C1)==0)
        {
            PORTD = 1;
        }

        if (INPUT(PIN_C2)==0)
        {
            PORTD = 3;
        }

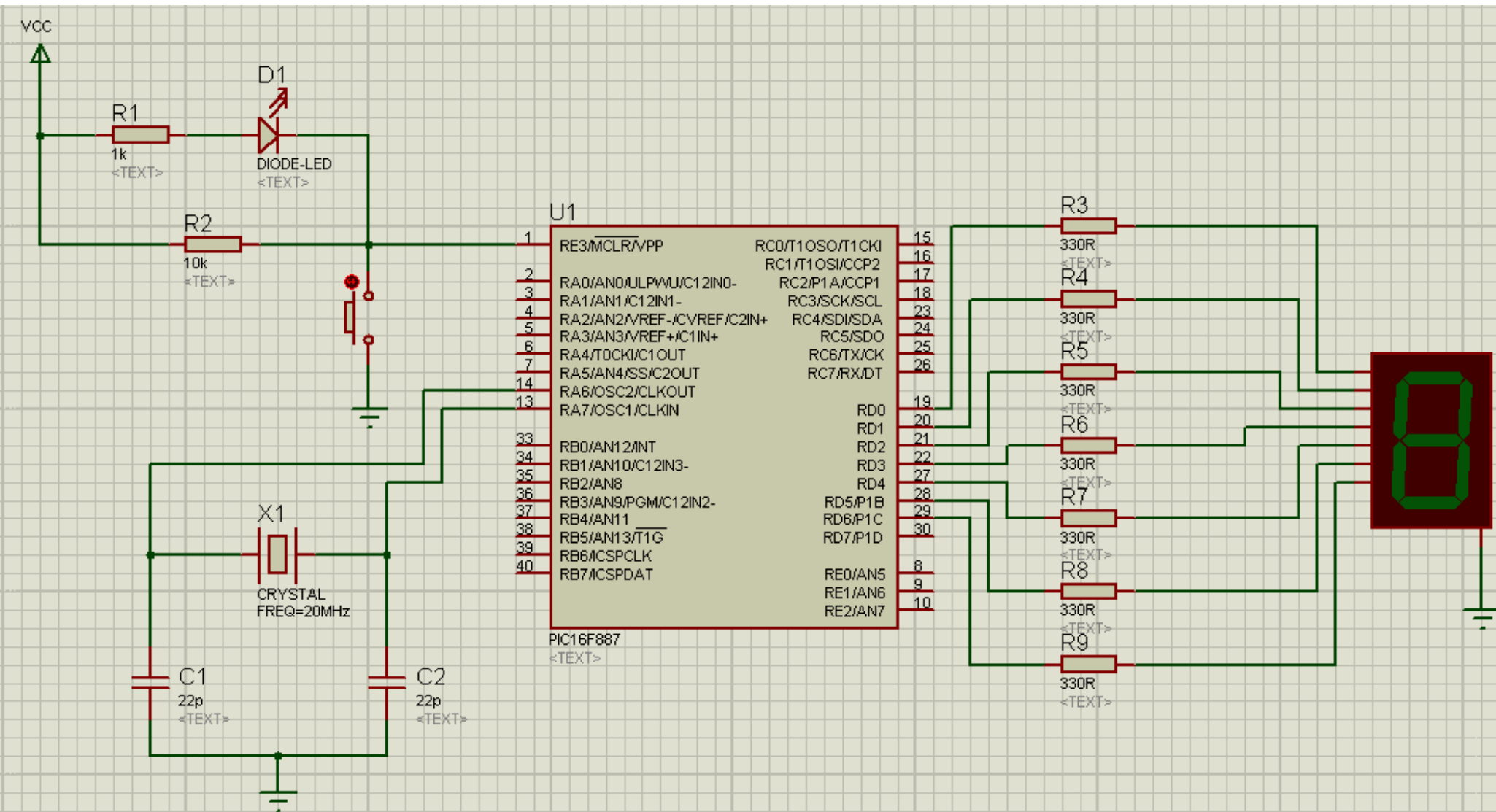
        if ((INPUT(PIN_C0)==1) &&
            (INPUT(PIN_C1)==1) &&
            (INPUT(PIN_C2)==1))
        {
            PORTD = 0;
        }
    }
}
```

VÍ DỤ 5: 1 LED 7 ĐOẠN (1)

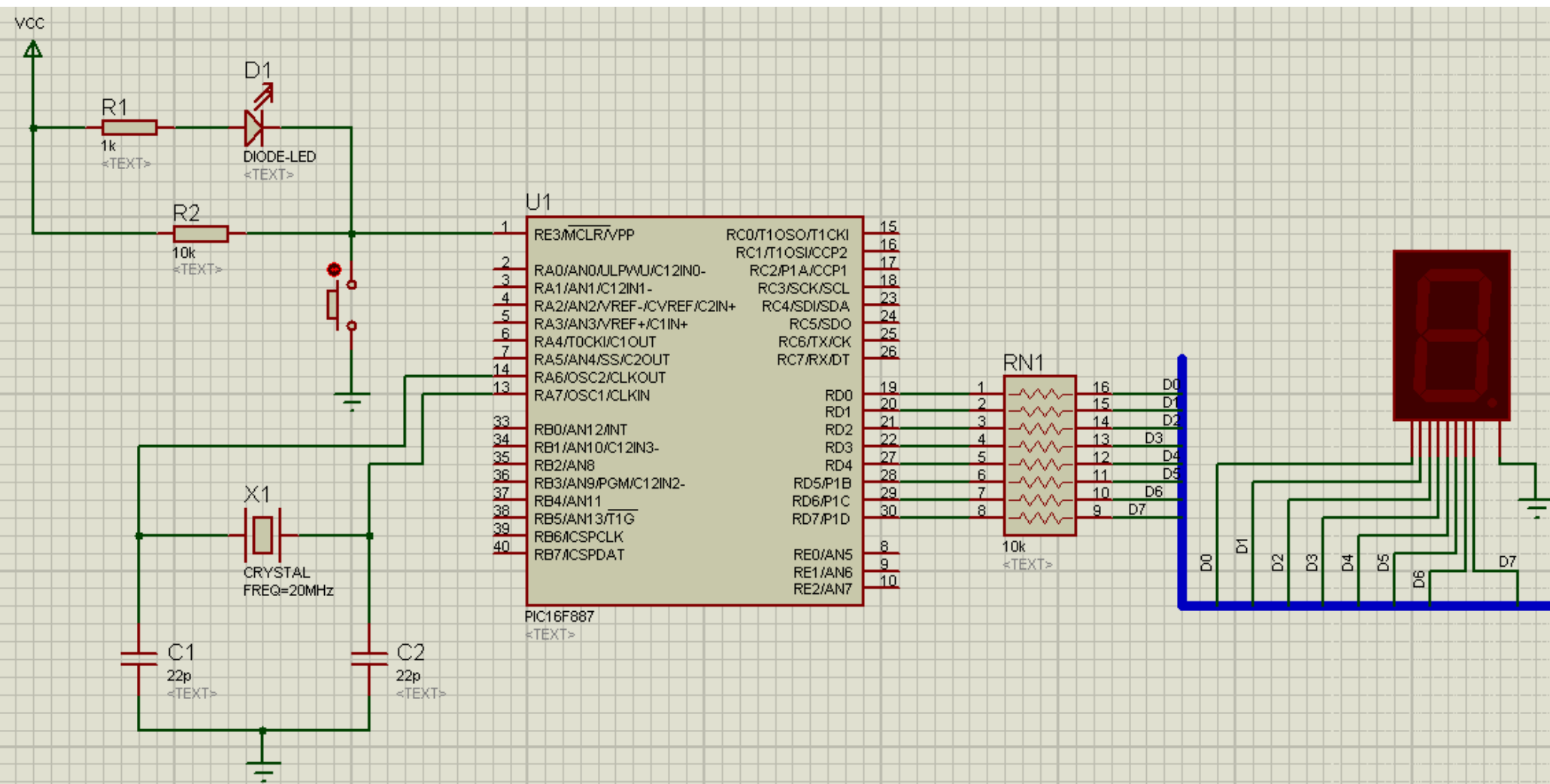
Viết code **CCS** hiển thị số **0** đến **9** cứ sau **2** giây

- Sử dụng linh kiện 7SEG-COM-CAT-GRN
- Sử dụng linh kiện 7SEG-MPX1-CC, RES16DIPIS và đường BUS

VÍ DỤ 5: 1 LED 7 ĐOẠN (2)



VÍ DỤ 5: 1 LED 7 ĐOẠN (3)



VÍ DỤ 5: 1 LED 7 ĐOẠN (4)

```
#include <16F887.h>
#use delay (clock=20000000)
#fuses HS,NOWDT,PUT,NOLVP,NOPROTECT,BROWNOUT
#byte PORTD=0x08

Char maso[10] = {0b00111111, 0b00000110, 0b01011011, 0b01001111,
0b01100110, 0b01101101, 0b01111101, 0b00000111, 0b01111111,
0b01101111};
int i;

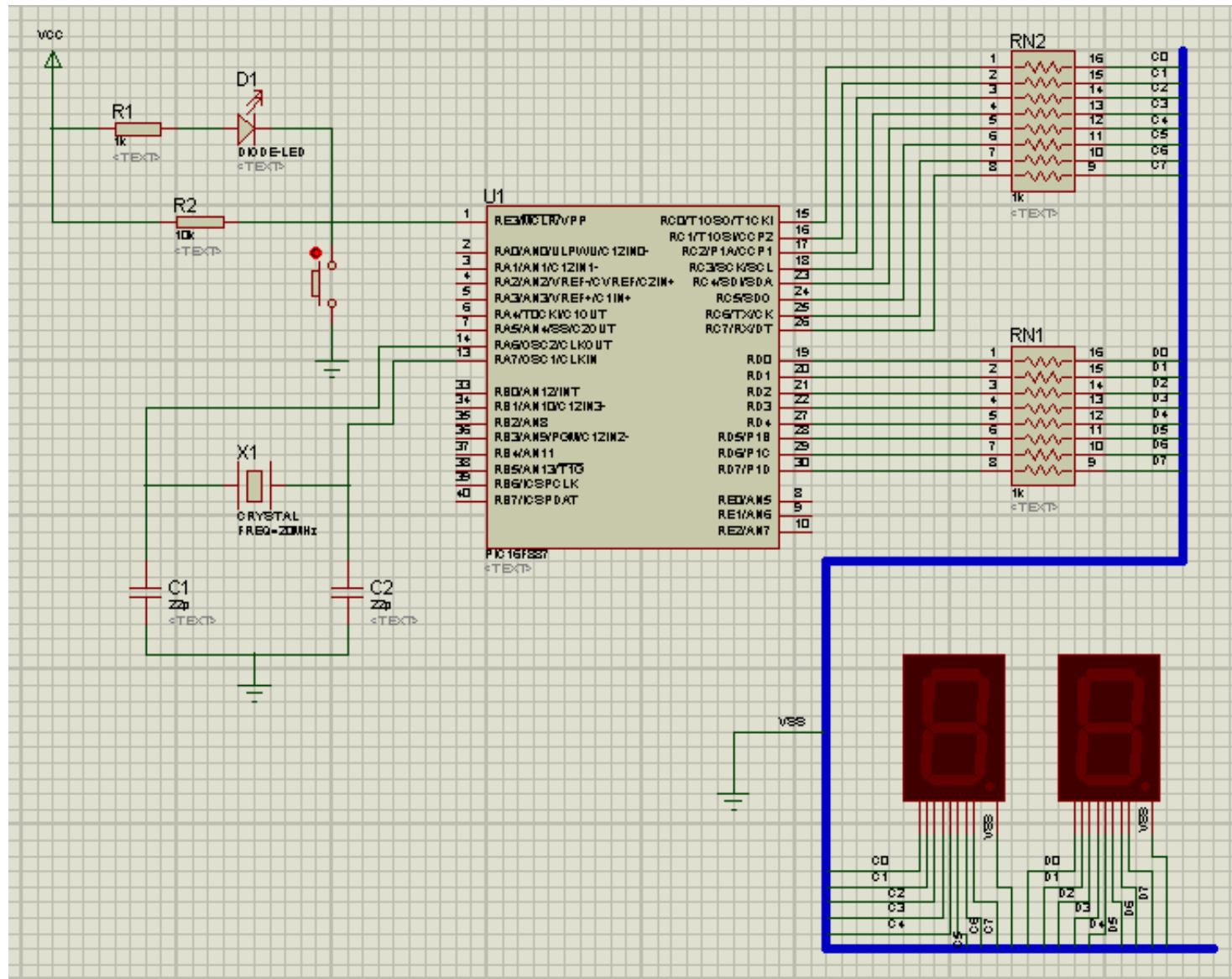
void xuat( char so)
{
    output_D(maso[so]);
}

void main()
{
    while(TRUE)
    {
        for(i=0;i<=9;i++)
        {
            xuat(i);
            delay_ms(2000);
        }
    }
}
```

VÍ DỤ 6: 2 LED 7 ĐOẠN (1)

Thiết kế và Viết code CCS hiển thị số: 09, 18, 27, 36, 45, 54, 63, 72, 81, 90 cứ sau 2 giây sử dụng linh kiện 7SEG-MPX1-CC, RES16DIPIS và đường BUS?

VÍ DỤ 6: 2 LED 7 ĐOẠN (2)



VÍ DỤ 6: 2 LED 7 ĐOẠN (3)

```
#include <16F887.h>
#use delay (clock=20000000)
#fuses HS,NOWDT,PUT,NOLVP,NOPROTECT,BROWNOUT
#byte PORTD=0x08

Char maso[10] = {0b00111111, 0b00000110, 0b01011011, 0b01001111,
0b01100110, 0b01101101, 0b01111101, 0b00000111, 0b01111111,
0b01101111};
int i;

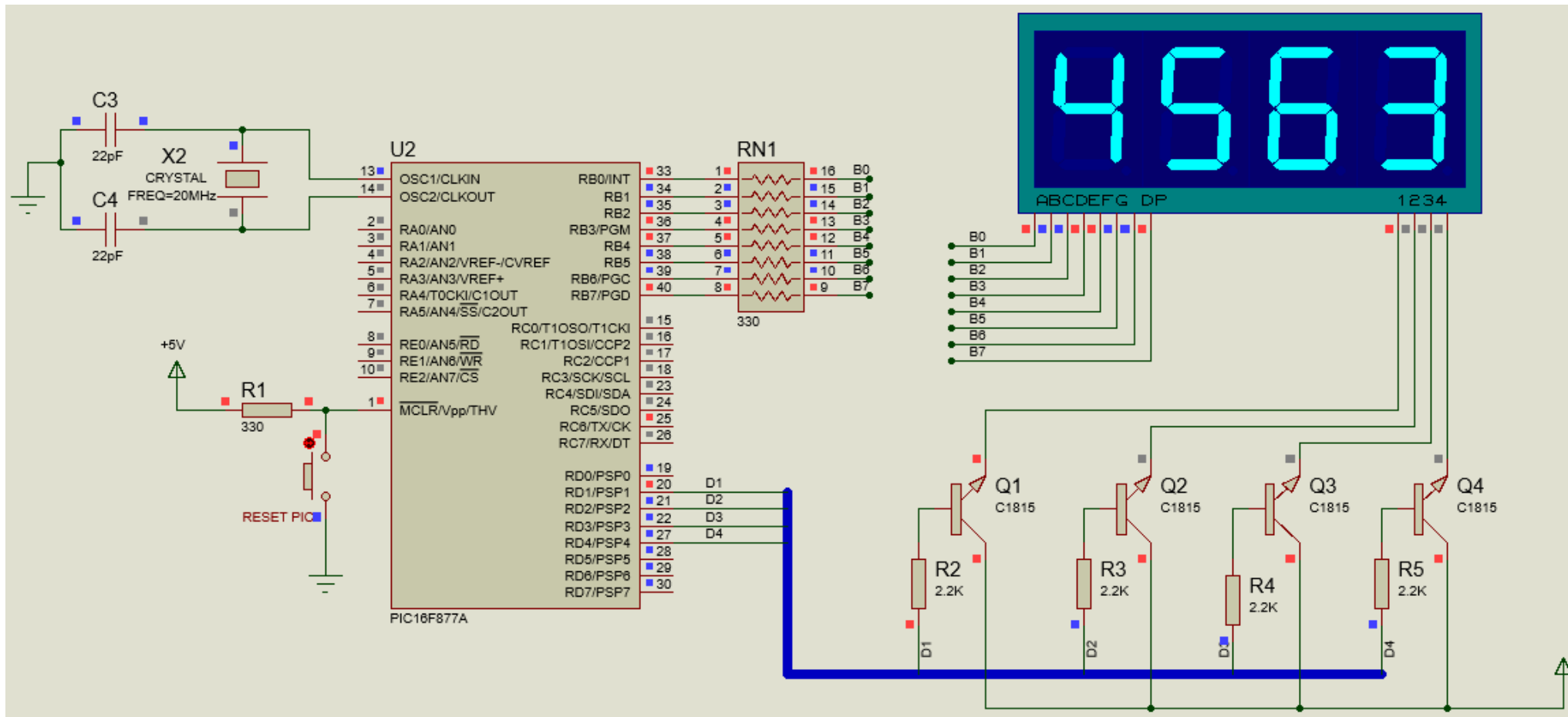
void xuat( char so)
{
    output_D(maso[9-so]);
    output_C(maso[so]);
}

void main()
{
    while(TRUE)
    {
        for(i=0;i<=9;i++)
        {
            xuat(i);
            delay_ms(2000);
        }
    }
}
```

VÍ DỤ 7: QUÉT LED (1)

Thiết kế và Viết code CCS hiển thị số: 0000, 1111, 2222, 3333, 4444, 5555, 6666, 7777, 8888 và 9999 cứ sau 1 giây sử dụng linh kiện 7SEG-MPX4-CC, RES16DIPIS và đường BUS?

VÍ DỤ 7: QUÉT LED (3)



VÍ DỤ 7: QUÉT LED (4)

```
#include <16F887.h>
#use delay (clock=20000000)
#fuses HS,NOWDT,PUT,NOLVP,NOPROTECT,BROWNOUT
#byte PORTD=0x08

Char maso[10] = {0b00111111, 0b00000110, 0b01011011, 0b01001111,
0b01100110, 0b01101101, 0b01111101, 0b00000111, 0b01111111,
0b01101111};
int i, nghin, tram, chuc, donvi;
int16 j;
int16 hien[10] = {0000,1111,2222,3333,4444,5555,6666,7777,8888,9999};
```

VÍ DỤ 7: QUÉT LED (5)

```
void hienthi(int16 a)
{
    nghin=a/1000;
    a=a%1000;
    tram=a/100;
    a=a%100;
    chuc=a/10;
    donvi=a%10;
    output_C(0b11111101);
    output_D(maso[nghin]);
    delay_us(300);
    output_C(0b11111011);
    output_D(maso[tram]);
    delay_us(300);
    output_C(0b11110111);
    output_D(maso[chuc]);
    delay_us(300);
    output_C(0b11101111);
    output_D(maso[donvi]);
    delay_us(300);
}
```

```
void main()
{
    i=0;
    output_C(0b11111111);
    while (1)
    {
        for (j=0;j<=666;j++)
        {
            hienthi(hien[i]);
        }
        i++;
        if(i>9)
        {
            i=0;
        }
    }
}
```