

Міністерство освіти і науки України  
Українська академія друкарства

**ЗВІТ**  
з лабораторної роботи № 3  
на тему «Вивчення портів мікроконтролера у режимі  
цифрового вводу інформації. Кнопка»

Виконала:

студ. групи КН-41

Гончарук Т. О.

Перевірив:

к. ф-м. н., доц. Пушак А. С.

**Тема роботи:** Вивчення портів мікроконтролера у режимі цифрового вводу інформації. Кнопка.

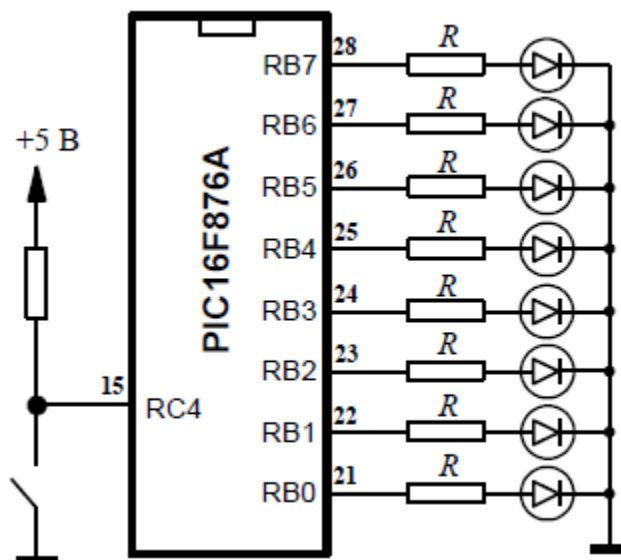
**Мета роботи:** Ознайомитись з портами мікроконтролера PIC16F876A. Налаштувати виводи портів на ввід інформації. Ознайомитись із схемами підключення кнопок до мікроконтролера.

### Хід роботи

1. У середовищі Proteus скласти схему зображену на рис. 3.1.
2. У середовищі MPLAB створити проєкт з кодом для зміни напрямку "біжучого вогники" за натискання кнопки.
3. Перевірити його роботу у середовищі Proteus.

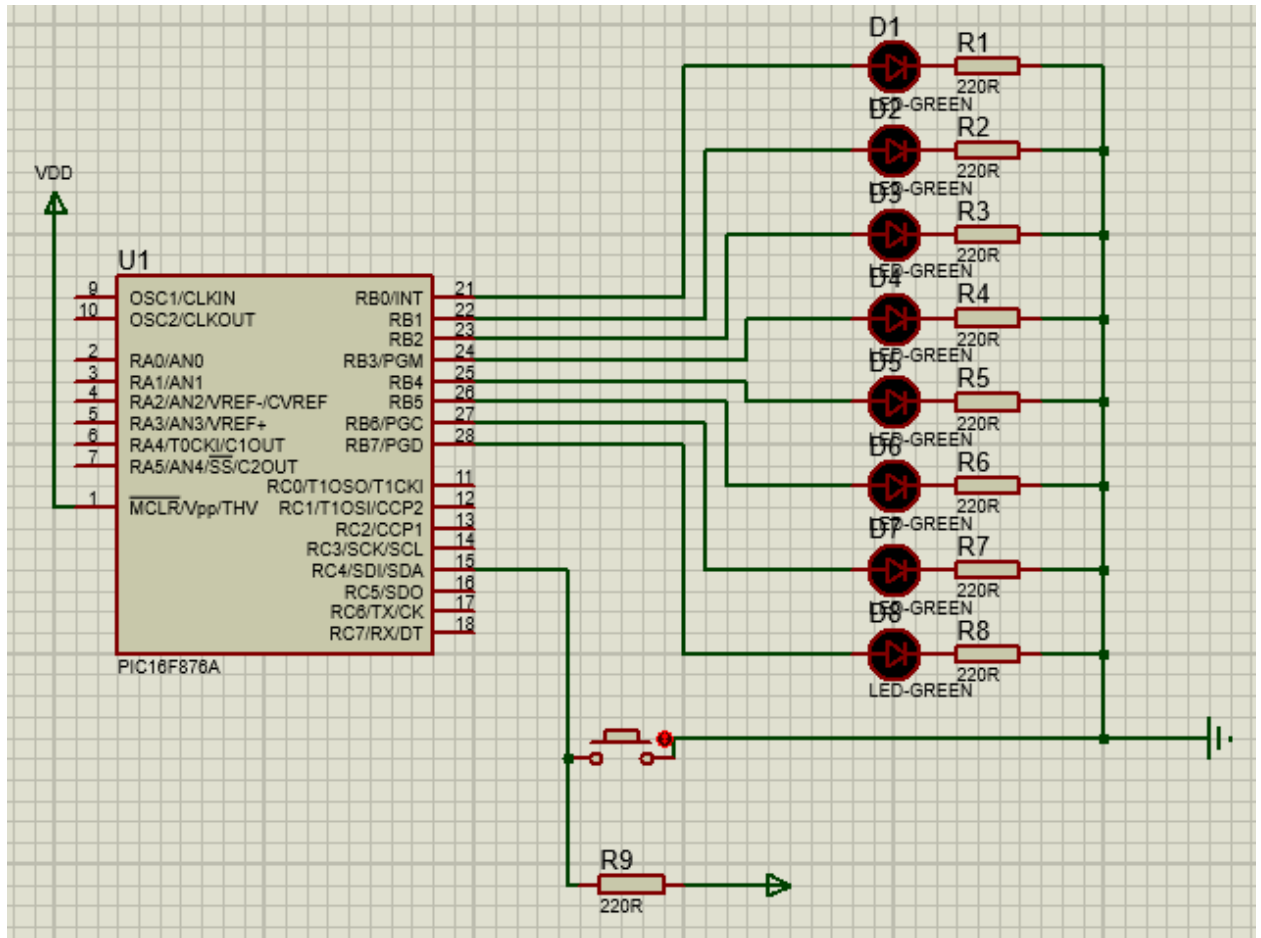
### Результати виконання

1. У середовищі Proteus складіть схему, приведену на рисунку 3.1.



**Рис. 3.1. Схема експериментальної плати для вивчення роботи кнопки**

У середовищі Proteus дана схема виглядає наступним чином:



**Рис. 3.2. Схема в симуляторі Proteus**

2. У середовищі MPLAB створити проєкт з кодом для зміни напрямку "біжучого вогники" за натискання кнопки.

```
#pragma config FOSC = HS
#pragma config WDTE = OFF
#pragma config PWRTE = OFF
#pragma config BOREN = OFF
#pragma config LVP = OFF
#pragma config CPD = OFF
#pragma config WRT = OFF
#pragma config CP = OFF

#include <xc.h>
#define _XTAL_FREQ 20000000

unsigned int i=1;
void main (void)
{
    TRISB=0b00000000;
    PORTB=0b00000000;
    TRISC=0b00010000;
```

```

while (1)
{
    PORTB = i;
    __delay_ms (300);
    i=i*2;
    if(i>256)
    {
        i=1;
    }

    if(RC4==0)
    {
        while(RC4==0)
        {
            i=i/2;
            PORTB = i;
            __delay_ms (300);
            if(i<1)
            {
                i = 256;
            }
        }
        while(RC4==1)
        {
            i=i/2;
            PORTB = i;
            __delay_ms (300);
            if(i<1)
            {
                i = 256;
            }
        }
    }
}
}

```

### 3. Перевірити роботу коду на симуляторі Proteus.

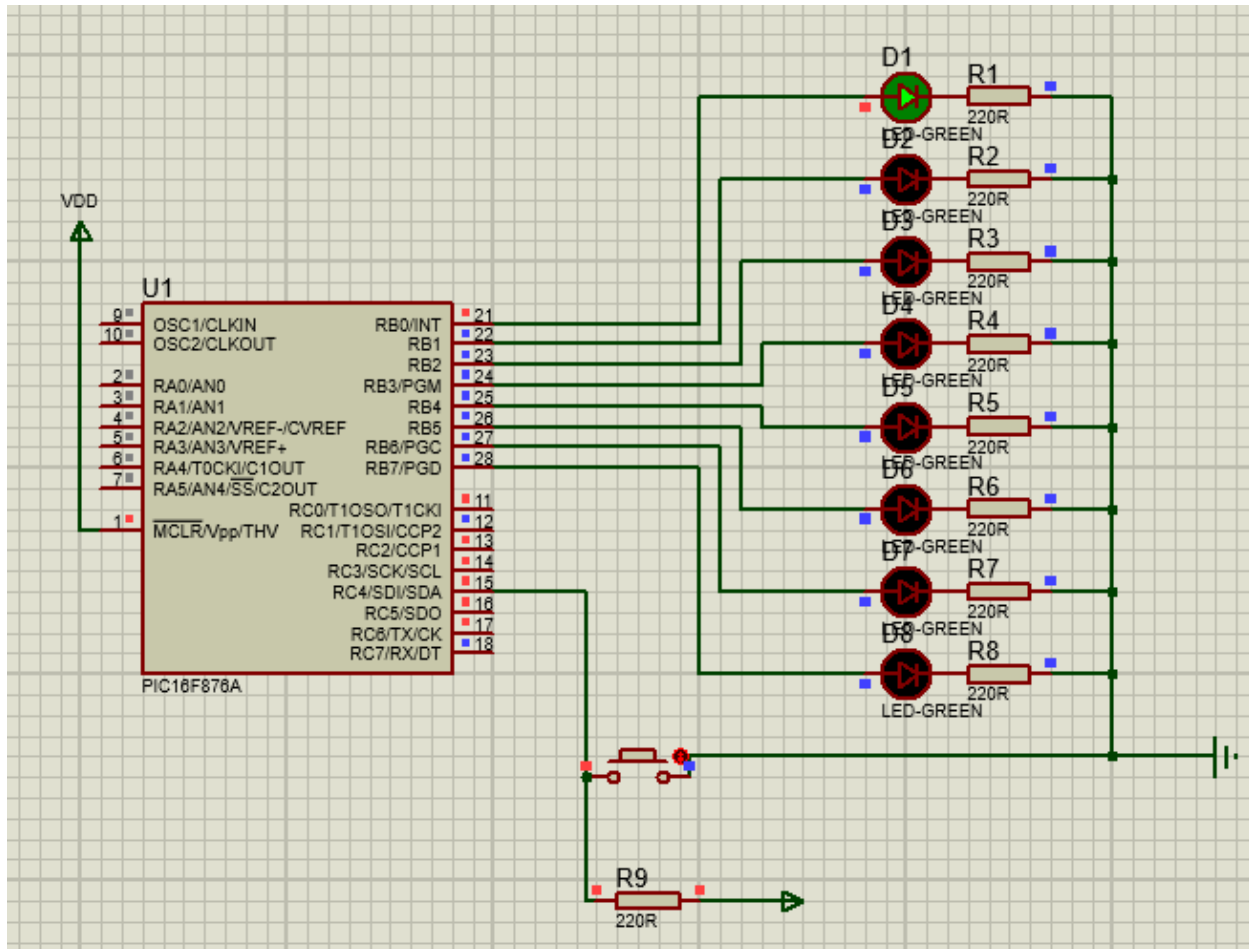


Рис. 3.3. Симуляція коду в програмі Proteus

#### Контрольні запитання

##### 1. Як налаштувати виводи порту на ввід інформації?

Порти виводу інформації налаштовуються на ввід інформації за допомогою команди  $TRIS(\text{назва порту})=0b*****$ , де замість \* використовують 1, якщо необхідно налаштувати на ввід, та 0, якщо вивід.

##### 2. Опишіть способи під'єднання кнопок до виводів мікроконтролера?

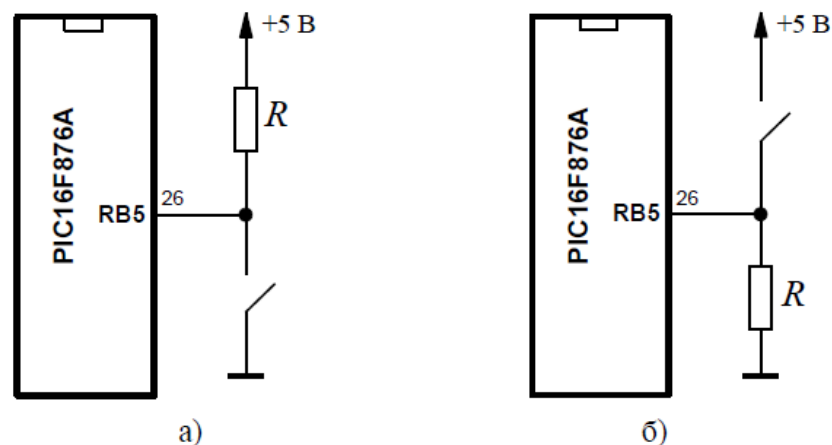


Схема а). Коли кнопка не натиснута, на вивід мікроконтролера подається високий рівень напруги, що відповідає значенню логічної “1”. При натисканні кнопки відбувається замикання виводу порту на “землю”, що приводить до появи на цьому виводі нульового потенціалу, що відповідає значенню логічного “0”. Отже, при натисканні даної кнопки на виводі мікроконтролера відбувається зміна напруги з високого логічного рівня до низького логічного рівня.

Схема б). Схема підключення кнопки протилежна за функціональністю до попередньої схеми, тобто, при натисканні кнопки на виводі мікроконтролера відбувається зміна напруги з низького логічного рівня до високого логічного рівня.

### 3. Що таке брязкіт контактів кнопок?

В момент натискання може виникнути часте короткотривале замикання і розмикання контактів.

### 4. Як усунути вплив брязкоту контактів кнопок у цифрових вузлах?

Зробити певну часову затримку за умови першої зміни рівня напруги перед зчитуванням рівня напруги на виводі мікроконтролера. Протягом цієї затримки можливий брязкіт контактів минає і настає новий стабільний рівень напруги, який тепер можна зчитувати.

### 5. Опишіть алгоритм опрацювання умови натискання кнопки?

```
unsigned char knopka (void)
{
    unsigned char result=0; // введемо змінну result
    unsigned int k=5000, i=0; // локальні змінні
    while (RC4==0) // натиснули кнопку
    {
        if (i<k)
        {
            i=i+1;
        }
        else
        {
            result=1;
            break;
        }
    }
}
```

```
return result;  
}
```

При натисканні кнопки програма попадає в тіло циклу з умовою натиснутої кнопки `while (RC4 == 0)`. В цьому циклі відбувається інкрементування певної локальної змінної ( $i = i + 1$ ;) до заданої величини ( $i < k$ ) за умови, що кнопка натиснута. Значення межі інкрементування  $k$  підбирається експериментально для кожного виду кнопок. Таке інкрементування триває певний час, за який брязкіт минає. Як тільки порушиться умова ( $i < k$ ), програма попадає в блок `else` де змінній `result` присвоюється значення рівне одиниці і відбувається вихід із циклу (`break`;) а далі функція `кнопка` повертає значення змінної `result` (`return result`;) . Весь цей час кнопка була натиснута, функція повернула значення змінної `result` рівне одиниці.

Коли хід виконання програми є в циклі `while (RC4 == 0)` і раптом через брязкіт контактів порушується умова (`RC4 == 0`), тоді цикл закінчується і функція `кнопка` повертає значення змінної `result` рівне нулю, як присвоєно по замовчуванні.

### **Висновки**

Під час виконання лабораторного практикуму я написала алгоритм для зміни напряму "біжучого вогника" після натискання кнопки. Даний алгоритм був протестований у симуляторі Proteus на відповідній схемі.