

Лабораторна робота №2
з дисципліни «Програмування мікропроцесорних систем»

на тему

«Програмування мікропроцесорних систем.
Цифрові порти введення-виведення, широко-
імпульсна модуляція (ШІМ), робота з кнопкою»

Виконав:
студент групи ІК-02 Моїсєєв К.
Викладач:
доц. Голубєв Л. П.

Зміст

Зміст	2
1. Постановка задачі	3
2. Виконання	4
3. Контрольні питання	7
4. Висновок	11
5. Додатки	11

1. Постановка задачі

Мета: познайомити студентів з основами програмування на Arduino та використанням нової інформації для створення приладів, з використанням світлодіодів та кнопок керування.

Завдання до роботи:

Задача №1. «Боротьба» з брязкотом кнопки.

Розробити власну функцію боротьби з **брязкотом кнопки**, використовуючи відповідну затримку t в часі :

$$t = \begin{cases} 5, & \text{якщо } n \% 5 = 0 \\ 6, & \text{якщо } n \% 5 = 1 \\ 7, & \text{якщо } n \% 5 = 2 \\ 8, & \text{якщо } n \% 5 = 3 \\ 9, & \text{якщо } n \% 5 = 4 \end{cases}$$

Перевірити роботу програми на одному світлодіоді. Натискання кнопки вона загоряється, наступне натискання – гасне. Вхідний порт для світлодіодів – $n \% 13$;

Задача №2. Робота з циклом.

Створити проект , в якому світлодіод буде блимати в циклі по правилу:

1) вхідний порт для світлодіоду – номер студента за списком $n \% 13$;

2) час «горіння» та паузи співпадають та дорівнюють у мс:

початкове значення n ;

крок – $n+100$;

кінцеве значення параметру циклу $100*(n \% 5)+1000$.

Задача №3. Широко імпульсна модуляція.

Скласти проекти по зміні яскравості світлодіоду від \min до \max , та від \max до \min з кроком 1.

$\min = n$,

$\max = 255-n$.

Задача №4. Робота з функціями користувача, глобальними та локальними змінними.

В схемі присутня кнопка та світлодіод. Початковий стан – кнопка не горить.

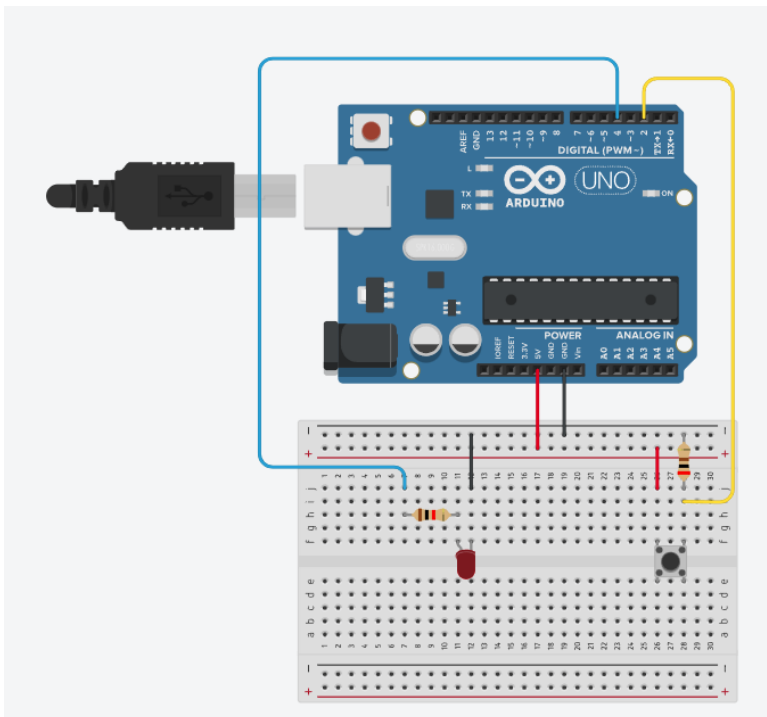
1. Натискування кнопки перший раз – світлодіод змінює яскравість від min до max;
2. Натискування кнопки другий раз – світлодіод змінює яскравість від max до min;
3. Натискування кнопки третій раз – світлодіод горить з максимальною яскравістю;
4. Натискування кнопки четвертий раз – світлодіод гасне і ми переходимо до початкового стану.

Відповідні значення для min та max та кроку взяти із задачі №3.

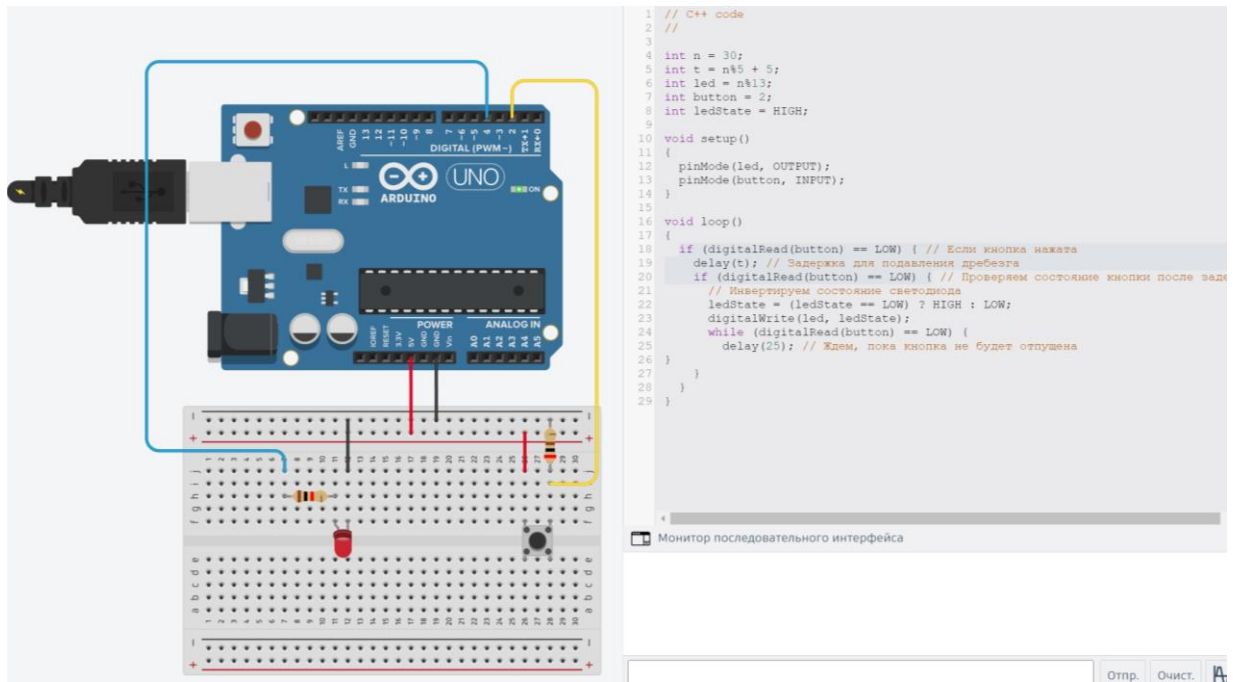
2. Виконання

Завдання 1:

Додаємо схему і на неї електрод та кнопки, ставимо по 2 резистори на 200 Ом, підключаємо землю і живлення, порти ставимо $30\%13 = 4$ за варіантом і застримку $30\%5 + 5 = 5$ за варіантом.



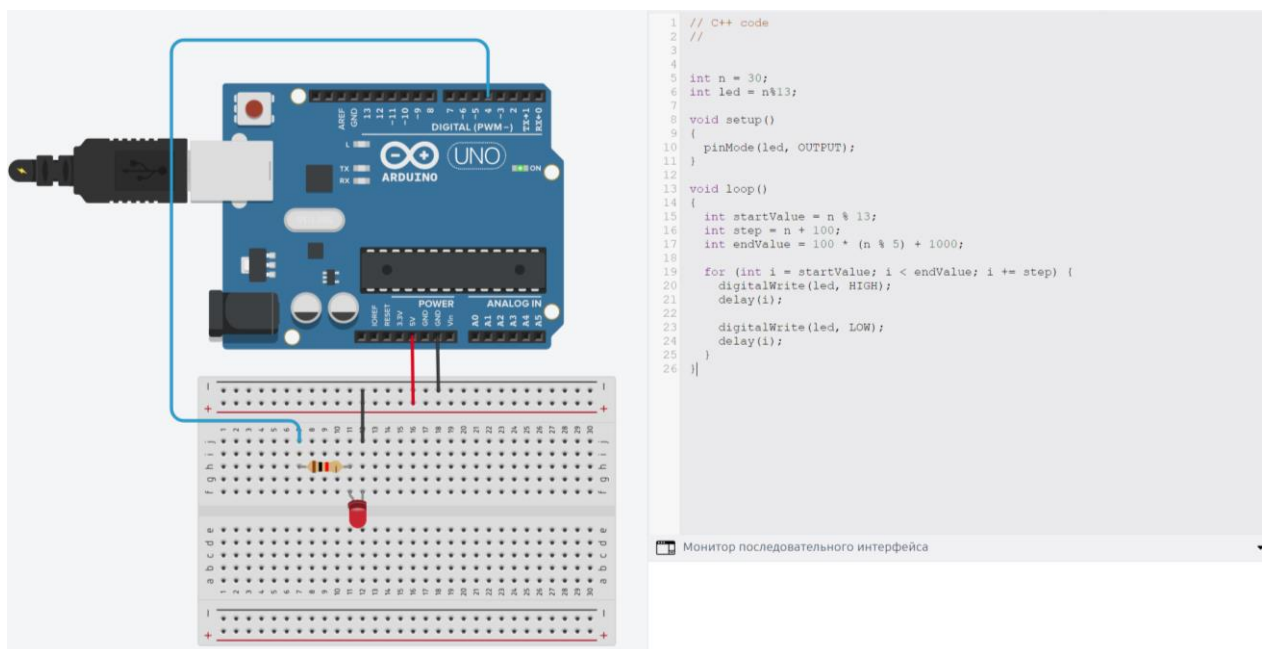
Пишімо код і запускаємо програму



Як ми бачимо, все працює.

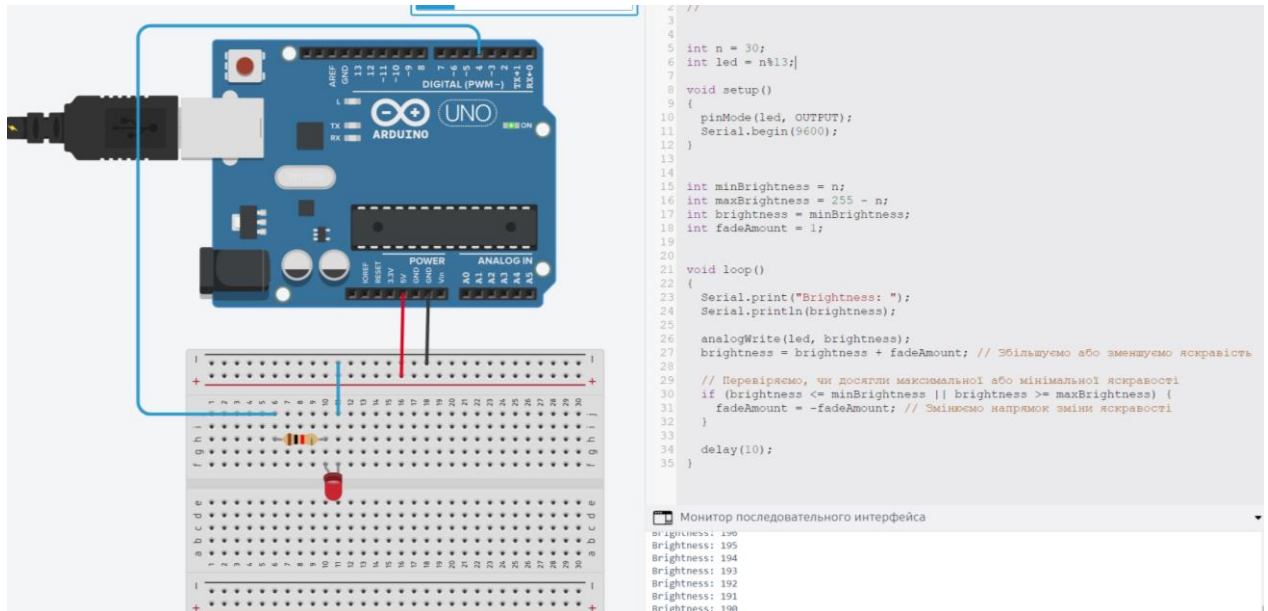
Завдання 2:

Будуємо таку ж схему, але без кнопки, у коді через цикл фор налаштуємо мигання діоду. В моєму випадку це були значення 30, 130, 1000, що були початковим значенням, кроком і кінцевим значенням, відповідно.



Завдання 3:

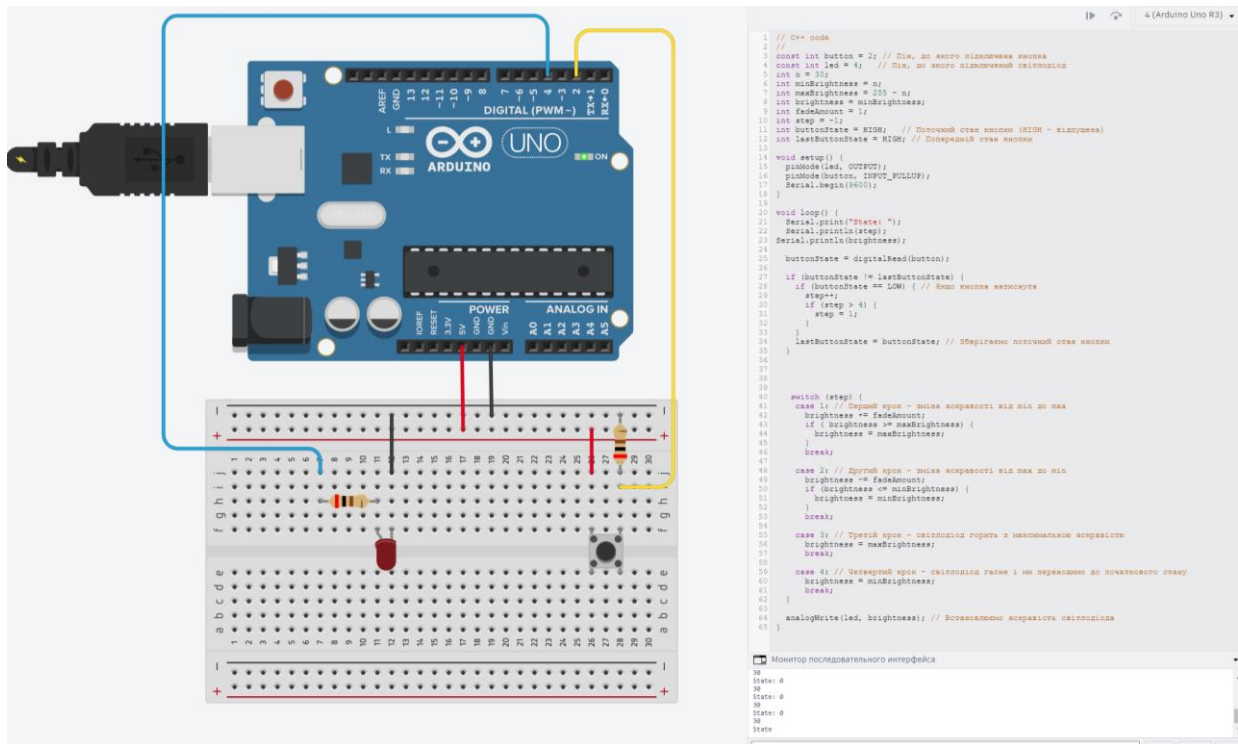
Знову повертаємось до наступної схеми, але тепер циклічно підвищуємо\понижуємо яскравість. В моєму випадку мінімальна – 30, максимальна – 225. Сенс полягає в тому, щоб спочатку підвищувати яскравість на 1 на кожному циклі, коли перевірка на >225 спрацює, то змінюємо показник на -1 і понижуємо, коли спрацює перевірка нижньої грані, то знову змінюємо.



Як бачимо, все працює.

Завдання 4:

Тут використовуємо схему з завдання 1.



Спочатку схема знаходиться у стані спокою. При першому натисканні – значення яскравості буде рости до максимального, при другому – знижуватися до мінімального, при третьому – ввімкнеться максимальна яскравість, а при четвертому – мінімальна. Потім цикл повторюється.

3. Контрольні питання

1. Що таке макетна плата? Як вона влаштована?

Макетна плата (breadboard) - це електронний пристрій, який використовується для створення прототипів електричних схем і з'єднання компонентів без необхідності паяння. Макетна плата має множини отворів, які з'єднані проводами і призначені для вставки і підключення різних електронних компонентів.

2. Що таке закон Ома?

Закон Ома - це фундаментальний закон електрики, який встановлює взаємозв'язок між напругою (V), опором (R) і силою струму (I) в електричному

колі. Закон Ома виражається формулою: $V = I * R$, де V - напруга (у вольтах), I - сила струму (у амперах), R - опір (у омах).

3. Як потрібно підключати світлодіоди?

Для підключення світлодіодів до Arduino зазвичай використовують резистор для обмеження сили струму, що проходить через світлодіод і захищає його від перевищення напруги. Анод світлодіода (довший ніж катод) підключається до позитивного виходу (піну) Arduino, а катод підключається до землі (GND) через резистор. Розмір резистора залежить від типу світлодіода і напруги Arduino.

4. Чим відрізняється константа від змінної?

Константа - це значення, яке не змінюється протягом виконання програми і завжди має фіксований рівень. Змінна, навпаки, може змінюватися під час виконання програми і може приймати різні значення.

5. Що таке цикл?

Цикл - це структура програмування, яка дозволяє виконувати певні дії або блок коду декілька разів підряд. Цикли використовуються для автоматизації повторюваних завдань.

6. Яка структура циклу for і як він працює?

Цикл for - це цикл, який має контрольований лічильник і виконується певну кількість разів. Його структура виглядає так: for (ініціалізація; умова; інкремент) { // код для виконання }. Умова перевіряється перед кожним повторенням циклу, і якщо вона істинна, цикл виконується.

7. Що таке ШІМ?

ШИМ (Широтно-імпульсна модуляція) - це метод керування сигналами, який дозволяє змінювати амплітуду сигналу шляхом варіювання часу тривалості імпульсів в сигналі. В Arduino ШІМ використовується для керування яскравістю світлодіодів, швидкістю моторів і багатьма іншими задачами.

8. Що таке скважність?

Скважність (Duty Cycle) - це відношення тривалості активного стану (час, коли сигнал увімкнений) до періоду циклу сигналу. Виражається у відсотках і визначає, як часто сигнал увімкнений протягом одного періоду.

9. Навіщо потрібна функція analogwrite()? Які у неї аргументи?

Функція analogWrite() в Arduino використовується для генерації сигналів ШІМ на певному піні для керування аналоговими пристроями, такими як світлодіоди. Її аргументи - номер піна і значення ШІМ (від 0 до 255), де 0 вимикає сигнал, а 255 робить його повністю увімкненим.

10. Як за допомогою ШІМ і циклу забезпечити плавне зростання яскравості світлодіоду?

Для плавного зростання яскравості світлодіода за допомогою ШІМ і циклу, ви можете використовувати цикл for, щоб змінювати значення ШІМ від 0 до 255 в певному інтервалі часу, створюючи ефект плавного переходу.

11. Навіщо потрібний «стягуючий» резистор?

"Стягуючий" резистор (пуль-ап резистор) використовується для підтягування сигналу до високого рівня (напруги +V) у випадках, коли зовнішні пристрої мають відкритий колектор або відкритий вивід. Це допомагає забезпечити стабільний рівень сигналу, коли пристрій не активний.

12.Що таке «брязкіт» кнопки? Чому він виникає? Як його усунути?

"Брязкіт" кнопки (debouncing) виникає внаслідок електричних перешкод при натисканні кнопки, що може призводити до декількох спрацювань кнопки при одному натисканні. Це може бути усунуто за допомогою програмного або апаратного антидребізгового фільтру.

13.Основні типи даних в Arduino.

Основні типи даних в Arduino включають цілі числа (int), знакові цілі числа (signed int), беззнакові цілі числа (unsigned int), дійсні числа (float), символи (char), та інші.

14.Арифметичні оператори в Arduino.

Арифметичні оператори в Arduino включають + (додавання), - (віднімання), * (множення), / (ділення) та % (відсоток).

15.Оператори порівняння в Arduino.

Оператори порівняння в Arduino включають == (рівність), != (нерівність), > (більше), < (менше), >= (більше або дорівнює), та <= (менше або дорівнює).

16.Логічні оператори в Arduino.

Логічні оператори в Arduino включають && (логічне "І"), || (логічне "АБО"), та ! (логічне заперечення).

17.Унарні оператори в Arduino.

Унарні оператори в Arduino включають - (змінний знак) і ++ (інкремент).

18.Основні оператори керування в Arduino.

Основні оператори керування в Arduino включають if, else, for, while, do...while.

19. Як зробити функцію користувача в Arduino?

Функцію користувача в Arduino можна створити, оголосивши її з вказанням типу повертаемого значення, імені функції та параметрів, які вона приймає. Приклад: `int myFunction(int param1, float param2) { // код функції }`.

20. Глобальні та локальні змінні в Arduino. Область видимості змінних.

Глобальні змінні в Arduino доступні для використання в будь-якій частині програми, тоді як локальні змінні обмежені областю видимості, яка визначається функцією, в якій вони оголошені. Локальні змінні не можуть бути використані за межами цієї функції.

4. Висновок

У цьому дослідженні ми детально розглянули тему програмування мікропроцесорних систем з основним акцентом на цифрових портах введення-виведення, широко-імпульсній модуляції (ШИМ) і роботі з кнопкою.

За допомогою цифрових портів введення-виведення, ми можемо забезпечити взаємодію мікропроцесора з навколишнім середовищем. Вони дозволяють нам зчитувати стан різних сенсорів, керувати зовнішніми пристроями і виконувати інші завдання з обміном даними.

5. Додатки

[Посилання на проект](#)

Код

Завдання 1

```
int n = 30;
```

```

int t = n%5 + 5;
int led = n%13;
int button = 2;
int ledState = HIGH;

void setup()
{
  pinMode(led, OUTPUT);
  pinMode(button, INPUT);
}

void loop()
{
  if (digitalRead(button) == LOW) { // Если кнопка нажата
    delay(t); // Задержка для подавлениядребезга
    if (digitalRead(button) == LOW) { // Проверяем состояние кнопки после
задержки
      // Инвертируем состояние светодиода
      ledState = (ledState == LOW) ? HIGH : LOW;
      digitalWrite(led, ledState);
      while (digitalRead(button) == LOW) {
        delay(25); // Ждем, пока кнопка не будет отпущена
      }
    }
  }
}

```

Завдання 2

```

int n = 30;
int led = n%13;

void setup()
{
  pinMode(led, OUTPUT);
}

void loop()
{
  int startValue = n % 13;
  int step = n + 100;
  int endValue = 100 * (n % 5) + 1000;

  for (int i = startValue; i < endValue; i += step) {

```

```
digitalWrite(led, HIGH);  
delay(i);  
  
digitalWrite(led, LOW);  
delay(i);  
}  
}
```

Завдання 3

```
int n = 30;  
int led = n%13;
```

```
void setup()  
{  
  pinMode(led, OUTPUT);  
  Serial.begin(9600);  
}
```

```
int minBrightness = n;  
int maxBrightness = 255 - n;  
int brightness = minBrightness;  
int fadeAmount = 1;
```

```
void loop()  
{  
  Serial.print("Brightness: ");  
  Serial.println(brightness);
```

```
  analogWrite(led, brightness);  
  brightness = brightness + fadeAmount; // Збільшуємо або зменшуємо яскравість
```

```

// Перевіряємо, чи досягли максимальної або мінімальної яскравості
if (brightness <= minBrightness || brightness >= maxBrightness) {
    fadeAmount = -fadeAmount; // Змінюємо напрямок зміни яскравості
}

delay(10);
}

```

Завдання 4

```

const int button = 2; // Пін, до якого підключена кнопка
const int led = 4; // Пін, до якого підключений світлодіод
int n = 30;
int minBrightness = n;
int maxBrightness = 255 - n;
int brightness = minBrightness;
int fadeAmount = 1;
int step = -1;
int buttonState = HIGH; // Поточний стан кнопки (HIGH - відпущена)
int lastButtonState = HIGH; // Попередній стан кнопки

void setup() {
    pinMode(led, OUTPUT);
    pinMode(button, INPUT_PULLUP);
    Serial.begin(9600);
}

void loop() {
    Serial.print("State: ");
    Serial.println(step);
    Serial.println(brightness);

    buttonState = digitalRead(button);

    if (buttonState != lastButtonState) {
        if (buttonState == LOW) { // Якщо кнопка натиснута
            step++;
            if (step > 4) {
                step = 1;
            }
        }
    }
}

```

```
    }  
  }  
  lastButtonState = buttonState; // Зберігаємо поточний стан кнопки  
}
```

```
switch (step) {  
  case 1: // Перший крок - зміна яскравості від min до max  
    brightness += fadeAmount;  
    if ( brightness >= maxBrightness) {  
      brightness = maxBrightness;  
    }  
    break;  
  
  case 2: // Другий крок - зміна яскравості від max до min  
    brightness -= fadeAmount;  
    if (brightness <= minBrightness) {  
      brightness = minBrightness;  
    }  
    break;  
  
  case 3: // Третій крок - світлодіод горить з максимальною яскравістю  
    brightness = maxBrightness;  
    break;  
  
  case 4: // Четвертий крок - світлодіод гасне і ми переходимо до початкового  
    стану  
    brightness = minBrightness;  
    break;  
}  
  
analogWrite(led, brightness); // Встановлюємо яскравість світлодіода  
}
```