

Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет Інформатики та Обчислювальної Техніки
Кафедра інформаційних систем та технологій

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 ІП-11
Дякунчак І.

Викладач:
доц. Голубєв Л. П.

Київ – 2024

Зміст

| | |
|----------------------------|---|
| Зміст..... | 2 |
| 1. Постановка задачі..... | 3 |
| 2. Виконання..... | 3 |
| 3. Контрольні питання..... | 6 |
| 4. Висновок..... | 7 |
| 5. Додатки..... | 8 |

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:

В кожній з робіт потрібно розробити схеми та заставити її працювати за правилами, що викладені в задачах 1-4. До кожної із задач у зошиті повинні бути намальовані відповідні схеми! Всюди $n=70$ – номер студента за списком.

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.

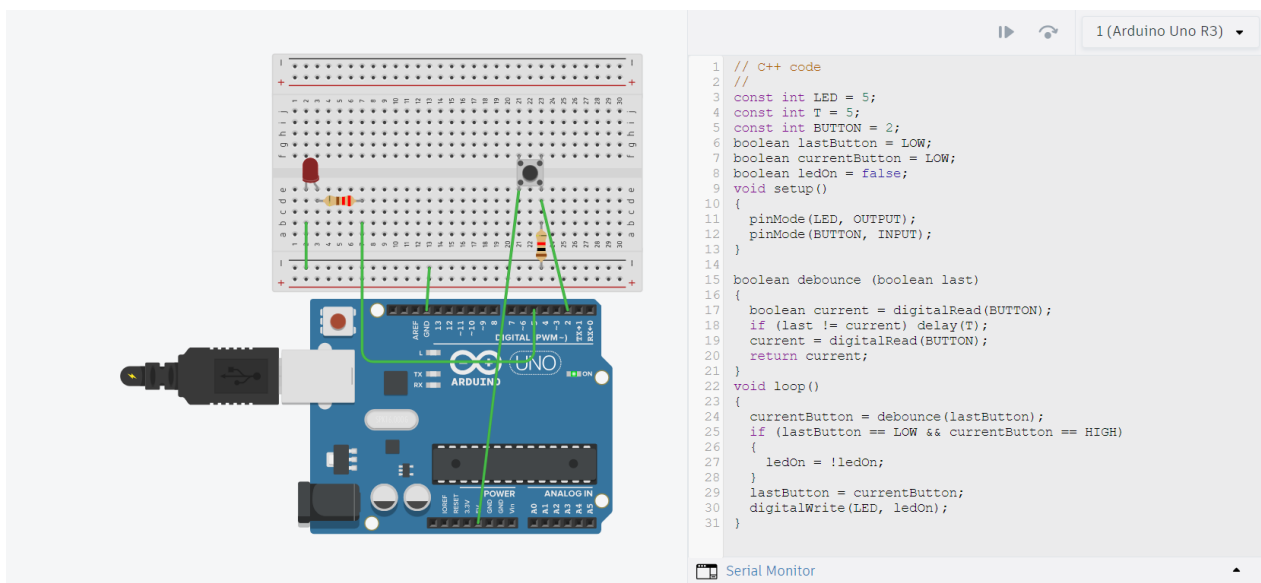
1. Натискування кнопки перший раз – світлодіод змінює яскравість від \min до \max ;

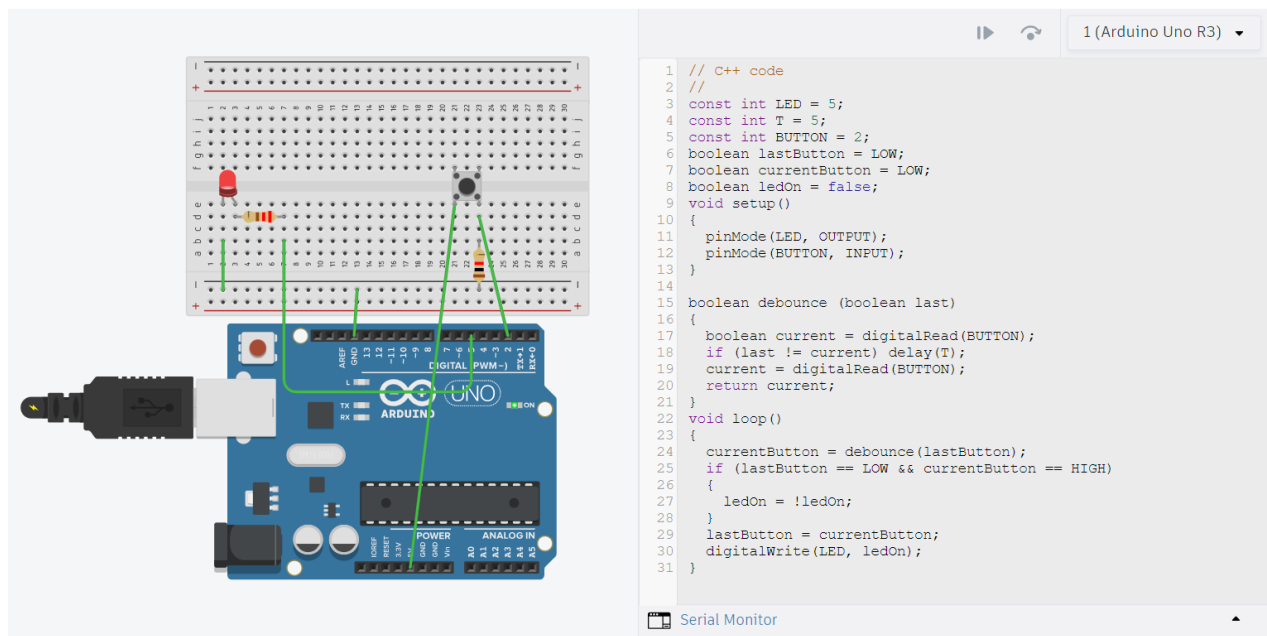
2. Натискування кнопки другий раз – світлодіод змінює яскравість від max до min;
3. Натискування кнопки третій раз – світлодіод горить з максимальною яскравістю;
4. Натискування кнопки четвертий раз – світлодіод гасне і ми переходимо до початкового стану.

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

2. Виконання

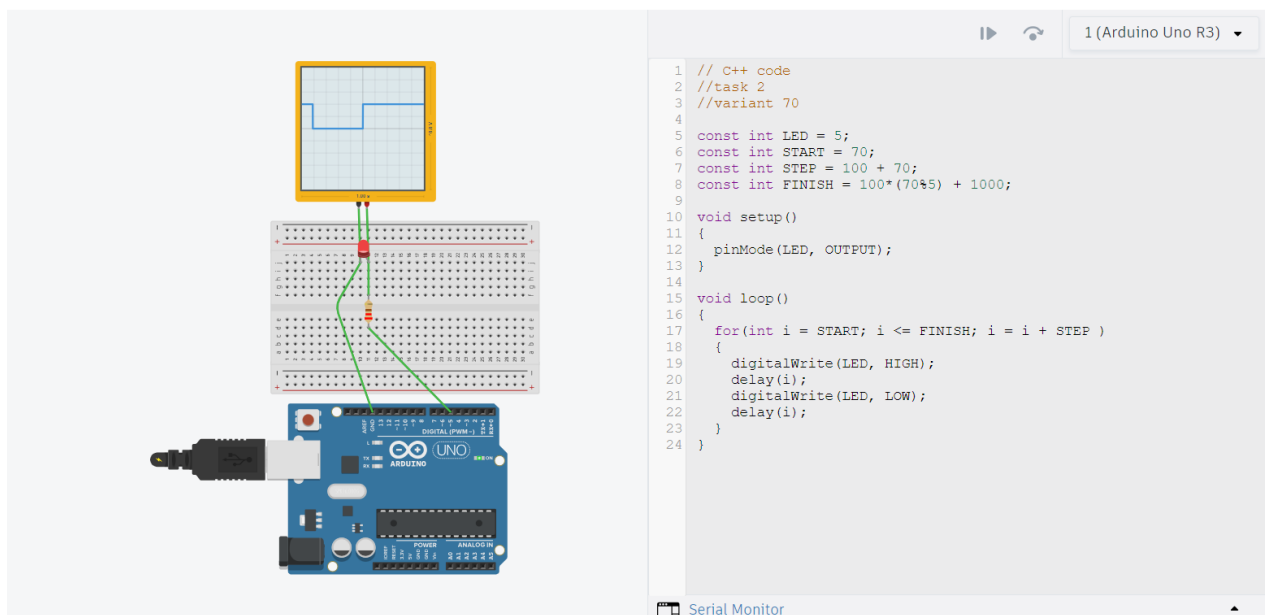
Спочатку потрібно було створити перший проект, суть якого полягає в розробці власної функції боротьби з брязкотом кнопки, код та знімки екрану з ввімкненим та вимкненим світлодіодом наведені нижче:



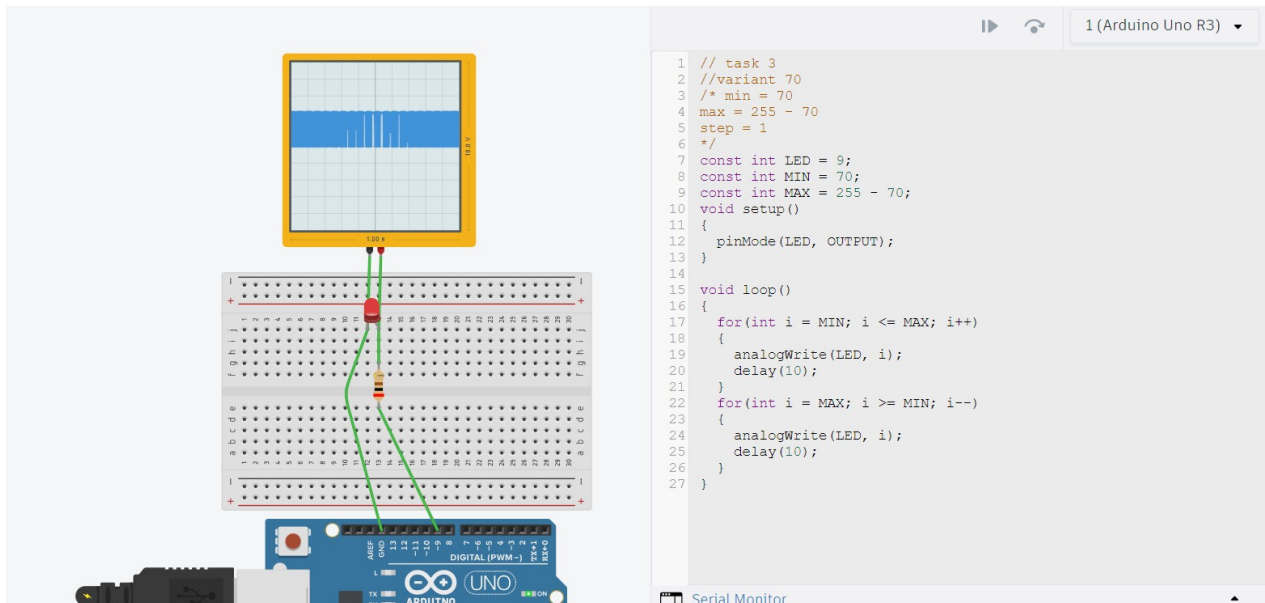


Як бачимо, усе працює, отже можна переходити до створення другого проекту, в якому світлодіод буде блимати в циклі.

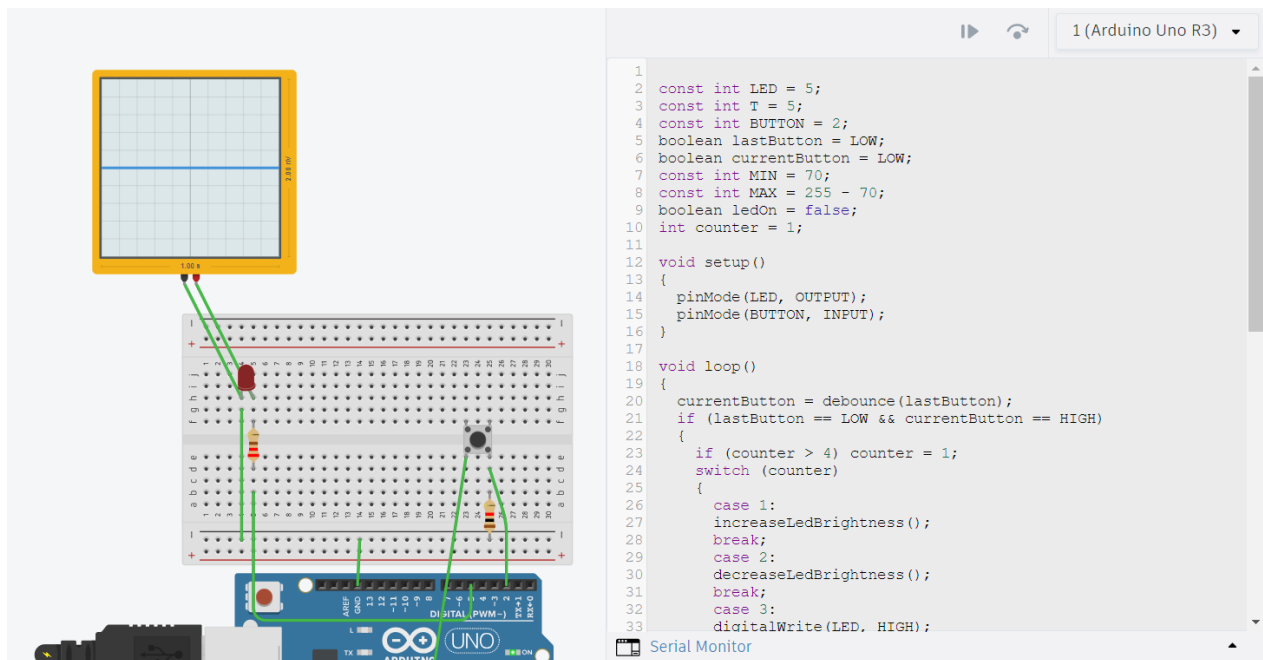
Результати роботи наведено на знімку екрану нижче, повний код програми можна знайти у додатках, також там знаходиться і посилання на проект.

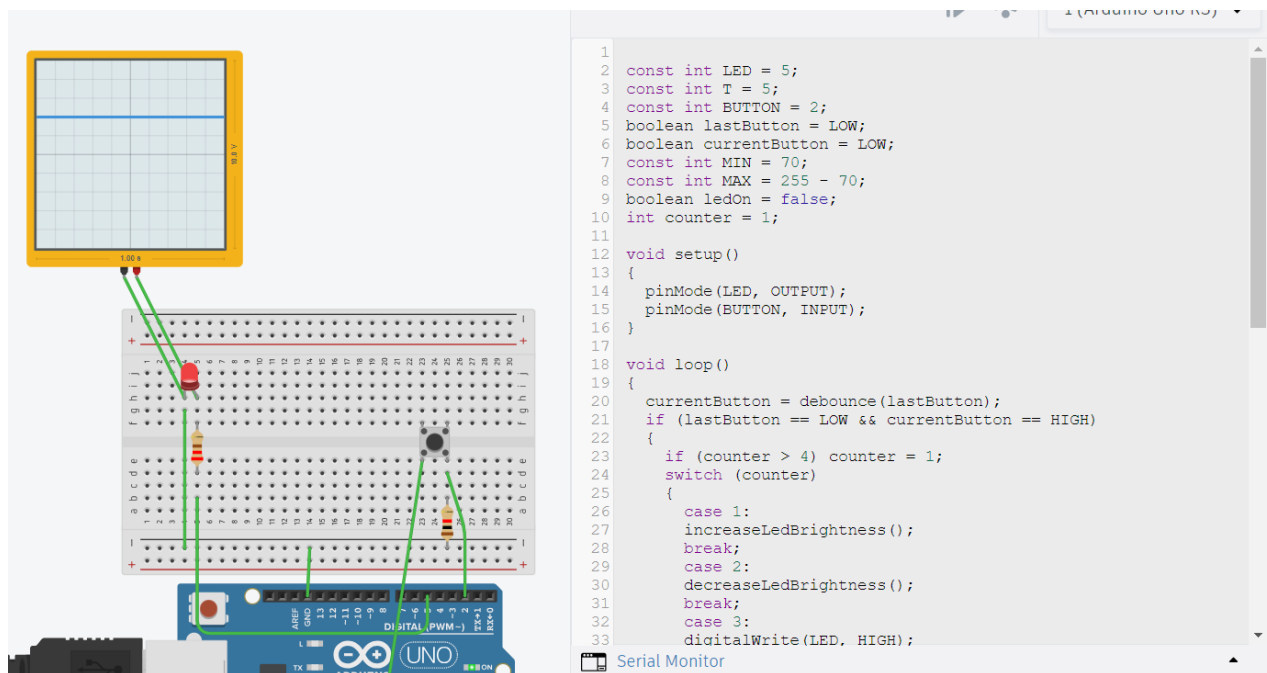
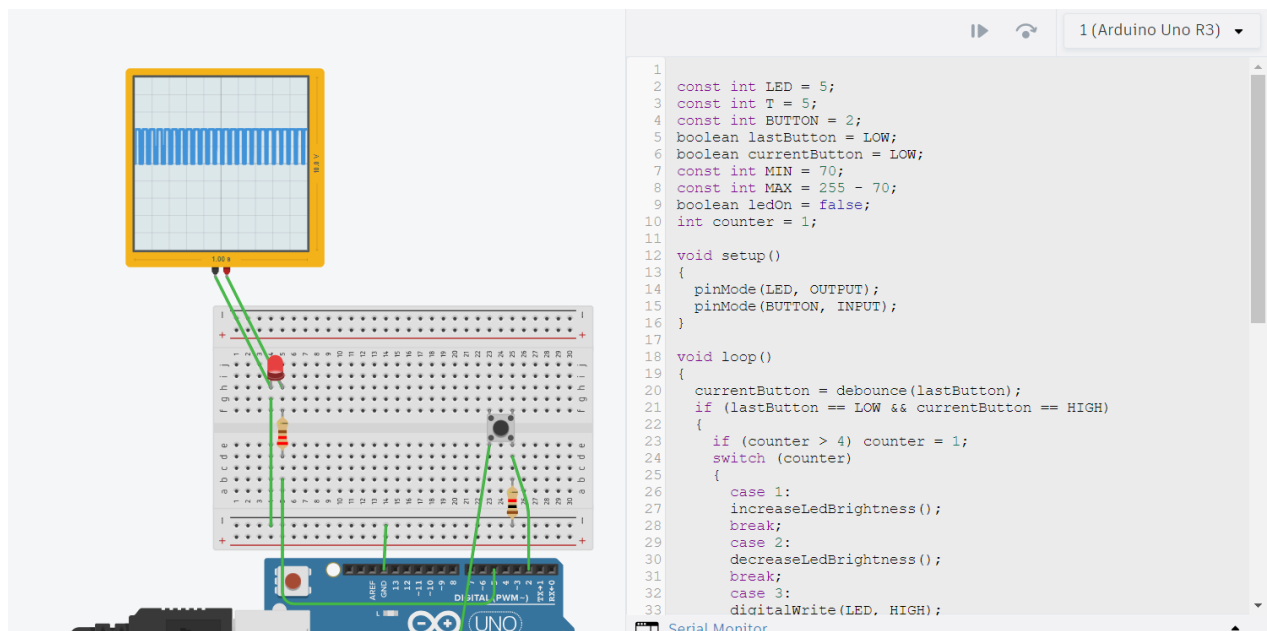


Як бачимо на екрані осцилографа, усе працює, отже можна переходити до створення третього проекту, в якому світлодіод буде змінювати яскравість від min до max.



Як бачимо на екрані осцилографа, усе працює, отже можна переходити до створення четвертого проекту.





3. Контрольні питання

1. Що таке макетна плата? Як вона влаштована?

Макетна плата – це плата для швидкого прототипування електронних схем, що має отвори з контактами для вставки елементів і провідників. Вона влаштована у вигляді сітки контактів.

2. Що таке закон Ома?

Закон Ома стверджує, що сила струму в провіднику прямо пропорційна напрузі і обернено пропорційна опору

3. Як потрібно підключати світлодіоди?

Світлодіоди потрібно підключати через резистор, щоб обмежити струм і запобігти їх перегріву.

4. Чим відрізняється константа від змінної?

Константа – це змінна, значення якої не змінюється в програмі, тоді як змінна може змінюватися в процесі виконання.

5. Що таке цикл?

Цикл – це конструкція програмування, яка дозволяє повторювати певний блок коду кілька разів.

6. Яка структура циклу for і як він працює?

Цикл for має структуру: for (ініціалізація; умова; інкремент/декремент) { код } і працює, виконуючи код, поки умова істинна, змінюючи змінні в кожному проході.

7. Що таке ШІМ?

ШІМ (широтно-імпульсна модуляція) – це метод управління середнім значенням напруги шляхом зміни ширини імпульсів, з якими подається сигнал.

8. Що таке скважність?

Скважність – це відношення часу, протягом якого сигнал перебуває у високому стані, до загального періоду сигналу, зазвичай виражається у відсотках.

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

Функція analogWrite() використовується для встановлення значення виходу ШІМ, приймає два аргументи: номер порту і значення від 0 до 255.

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

Щоб забезпечити плавне зростання яскравості світлодіоду, можна використовувати цикл, що змінює значення в analogWrite() від 0 до 255 з паузою між змінами.

11. Навіщо потрібний «стягуючий» резистор?

Стягуючий резистор необхідний для запобігання "плавання" сигналів або коротких сплесків, коли кнопка розімкнута.

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

Брязкіт кнопки – це засвітки контактів при натисканні кнопки через механічні коливання. Усувається програмним фільтром (дебаунс) або використанням конденсаторів.

13. Основні типи даних в Arduino.

Основні типи даних в Arduino: int, float, char, bool, byte.

14. Арифметичні оператори в Arduino.

Арифметичні оператори в Arduino: +, -, *, /, %.

15. Оператори порівняння в Arduino.

Оператори порівняння в Arduino: ==, !=, <, >, <=, >=.

16. Логічні оператори в Arduino.

Логічні оператори в Arduino: && (AND), || (OR), ! (NOT).

17. Унарні оператори в Arduino.

Унарні оператори в Arduino: ++ (інкремент), -- (декремент), - (унарний мінус).

18. Основні оператори керування в Arduino.

Основні оператори керування в Arduino: if, else, switch, while, for.

19. Як зробити функцію користувача в Arduino?

Щоб створити функцію користувача в Arduino, потрібно вказати тип повернення, ім'я функції та параметри, напр.: void myFunction(int x) { /* код */ }.

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

Глобальні змінні доступні в усій програмі, тоді як локальні змінні доступні лише в межах функції, в якій вони оголошені. Області видимості змінних визначають, де вони можуть використовуватися.

4. Висновок

У даній лабораторній роботі я познайомилась з основами програмування на Arduino та використанням нової інформації для створення приладів, з використанням світлодіодів та кнопок керування. В процесі виконання я створила 4 проекти з використанням ШІМ, функцій користувача, глобальних та локальних змінних. Усі результати наведені на знімках екрану вище, код програми та посилання на сам проект

5. Додатки

Посилання на перший проект: <https://www.tinkercad.com/things/74G2uldlnNf-lab21?sharecode=Pb2ZANNZ74Mo1bGBpjbfFNABlfbcL7WQHeTH8gmRx8>

Код проекту:

```
const int LED = 5;
```

```

const int T = 5;
const int BUTTON = 2;
boolean lastButton = LOW;
boolean currentButton = LOW;
boolean ledOn = false;
void setup()
{
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}

boolean debounce (boolean last)
{
  boolean current = digitalRead(BUTTON);
  if (last != current) delay(T);
  current = digitalRead(BUTTON);
  return current;
}
void loop()
{
  currentButton = debounce(lastButton);
  if (lastButton == LOW && currentButton == HIGH)
  {
    ledOn = !ledOn;
  }
  lastButton = currentButton;
  digitalWrite(LED, ledOn);
}

```

Посилання на другий проект :

Код проекту : https://www.tinkercad.com/things/2j4w0CljGEM-lab22?sharecode=xcGqyvnhL9-QSVhlpX9CPlco213_6AMYzawYVoFzNo

```

const int LED = 5;
const int START = 70;

```

```

const int STEP = 100 + 70;
const int FINISH = 100*(70%5) + 1000;

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

void loop()
{
  for(int i = START; i <= FINISH; i = i + STEP )
  {
    digitalWrite(LED, HIGH);
    delay(i);
    digitalWrite(LED, LOW);
    delay(i);
  }
}

```

Посилання на третій проект :

Код проекту : https://www.tinkercad.com/things/1LKfQl1Qh3D-lab23?sharecode=MRwT87ffDfkkdFD5NmhgDk2A_RqFDIXluhrrv4hCIRU

```

const int LED = 9;
const int MIN = 70;
const int MAX = 255 - 70;
void setup()
{
  pinMode(LED, OUTPUT);
}

void loop()
{
  for(int i = MIN; i <= MAX; i++)

```

```

{
  analogWrite(LED, i);
  delay(10);
}
for(int i = MAX; i >= MIN; i--)
{
  analogWrite(LED, i);
  delay(10);
}
}

```

Посилання на четвертий проект :

Код проекту : https://www.tinkercad.com/things/dFnNoTJ2Xk5-lab24?sharecode=oe1MbT_UZ4knjwjA3dKqY253pLTGasBNK8Xgd17VMgo

```

const int LED = 5;
const int T = 5;
const int BUTTON = 2;
boolean lastButton = LOW;
boolean currentButton = LOW;
const int MIN = 70;
const int MAX = 255 - 70;
boolean ledOn = false;
int counter = 1;

void setup()
{
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}

void loop()
{
  currentButton = debounce(lastButton);

```

```

if (lastButton == LOW && currentButton == HIGH)
{
    if (counter > 4) counter = 1;
    switch (counter)
    {
        case 1:
            increaseLedBrightness();
            break;
        case 2:
            decreaseLedBrightness();
            break;
        case 3:
            digitalWrite(LED, HIGH);
            break;
        case 4:
            digitalWrite(LED, LOW);
            break;
    }
    counter++;
}
lastButton = currentButton;
}

```

```

boolean debounce (boolean last)
{
    boolean current = digitalRead(BUTTON);
    if (last != current) delay(T);
    current = digitalRead(BUTTON);
    return current;
}

```

```

void increaseLedBrightness()
{
    for (int i = MIN; i <= MAX; i++)
    {

```

```
    analogWrite(LED, i);  
    delay(20);  
  }  
}  
  
void decreaseLedBrightness()  
{  
  for (int i = MAX; i >= MIN; i--)  
  {  
    analogWrite(LED, i);  
    delay(20);  
  }  
}
```