

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ ІМ. ТАРАСА ШЕВЧЕНКА ФІЗИЧНИЙ ФАКУЛЬТЕТ

Звіт

Про виконання лабораторної роботи №3 з курсу «Сучасна електроніка»

Роботу виконали:

Гриценяк Олександра Олександрівна

Мальцева Діана Сергіївна

Науковий керівник:

Єрмоленко Руслан Вікторович

Київ 2020

ЗМІСТ

1. Коротко про головне	3
2. Будова програми та використані команди	4
3. Проект «Блимаючий світодіод»	5
4. Проект «Світлофор»	7
5. Висновок	10

1. Коротко про головне

Мікроконтролер на платформі Arduino – це простий у використанні, але потужний одноплатний комп'ютер.

Arduino Uno – це плата з мікроконтролерів, створена на базі ATmega328P. Вона має 14 цифрових пінів входу/виходу (6 з яких можуть бути використані як PWM виходи), 6 аналогових входів, керамічний резонатор частотою 16 МГц (CSTCE16M0V53-R0), USB з'єднання, роз'єм живлення, ICSP регулятор і кнопку скидання. Плата містить все необхідне для забезпечення мікроконтролера; її можна легко під'єднати до комп'ютера за допомогою USB кабеля і забезпечити її живлення з використанням спеціального адаптера.



Рис. 1. Плата Arduino Uno.

Керувати платою Arduino можна за допомогою написання програмного коду на комп'ютері. Написання програми відбувається в середовищі розробки Arduino, а потім завантажуються в плату Arduino. Мова програмування Arduino – це спрощена версія C/C++. Зручність полягає у тому, що після створення програми управління на ПК та завантаження її в плату Arduino програма роботи не змінюється, доки не буде натиснуто кнопку скидання або завантажено нову програму. Якщо від'єднати USB-кабель від ПК і перейти на автономне живлення, програма продовжуватиме працювати автоматично.

2. Будова програми та використані команди

Будь яка програма Arduino має дві функції: налаштування – `setup()` та цикл – `loop()`. Команди, написані всередині `setup()`, виконуються тільки один раз, коли починається програма, і використовується для ініціалізації. Інструкції, поміщені в циклі `loop()`, виконуються багаторазово і утворюють основні завдання програми.

Використані команди:

- *pinMode*: використовується, щоб установити цифровий вивід (його номер записується першим аргументом функції) як вхід(INPUT) чи вихід(OUTPUT);
- *digitalWrite*: подає на вихід, номер якого вказано в аргументі функції, високу (+ 5 В) або низьку (0 В) напругу. Для цього в аргументі вказується відповідно HIGH або LOW;
- *delay*: затримує програму на вказане число мілісекунд.

Також продемонстровано використання коментарів, використання яких хоч і не є суттєвим у розглянутих випадках, але загалом є гарною звичкою для підвищення читабельності коду.

Крім того, показано, наскільки раціоналізує програму використання циклів.

3. Проект «Блимаючий світлодіод»

Першою спробою використання засобів Arduino було збирання схеми для реалізації блимаючого світлодіода та написання відповідного програмного коду.

Для цього проекту знадобилися:

- LED-світлодіод;
- резистор 330 Ом;
- кілька коротких шматків дроту;
- плата для макета;
- плата Arduino;
- USB-дріт для завантаження програми.

Орієнтовну схему наведено на рис. 2.

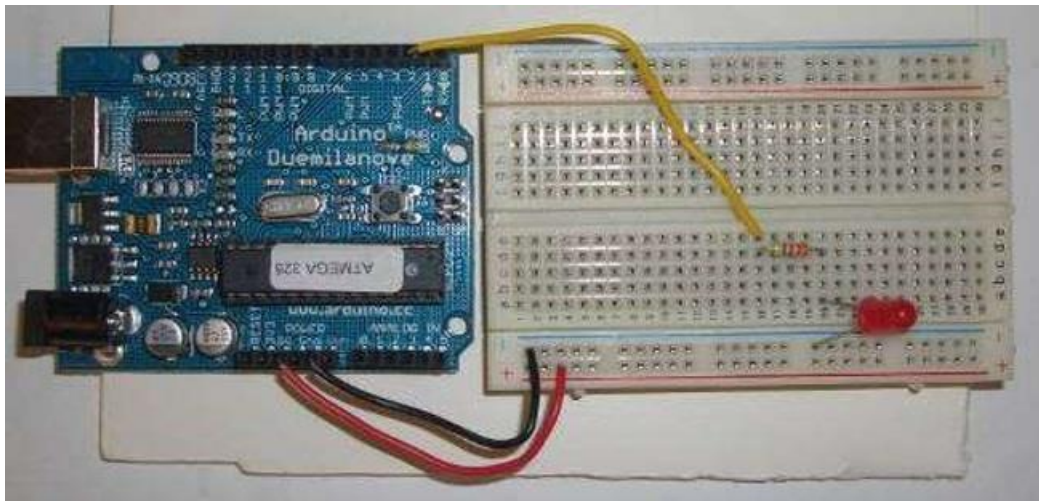


Рис. 2. Схема підключення світлодіода.

Найголовніше, що треба пам'ятати при підключенні світлодіода – він розрахований на порівняно невеликі величини сили струму, тому його треба підключати послідовно з резистором, який збільшує сумарний опір ділянки і таким чином запобігає перегоранню діода.

Як виглядав проект у реальному житті, можна побачити на рис. 3. Посилання на повне відео:

https://github.com/npd2020/electronics/blob/master/gr5b/Olexandra_Hrytseniak/lab_3/lab_3_1.mp4.

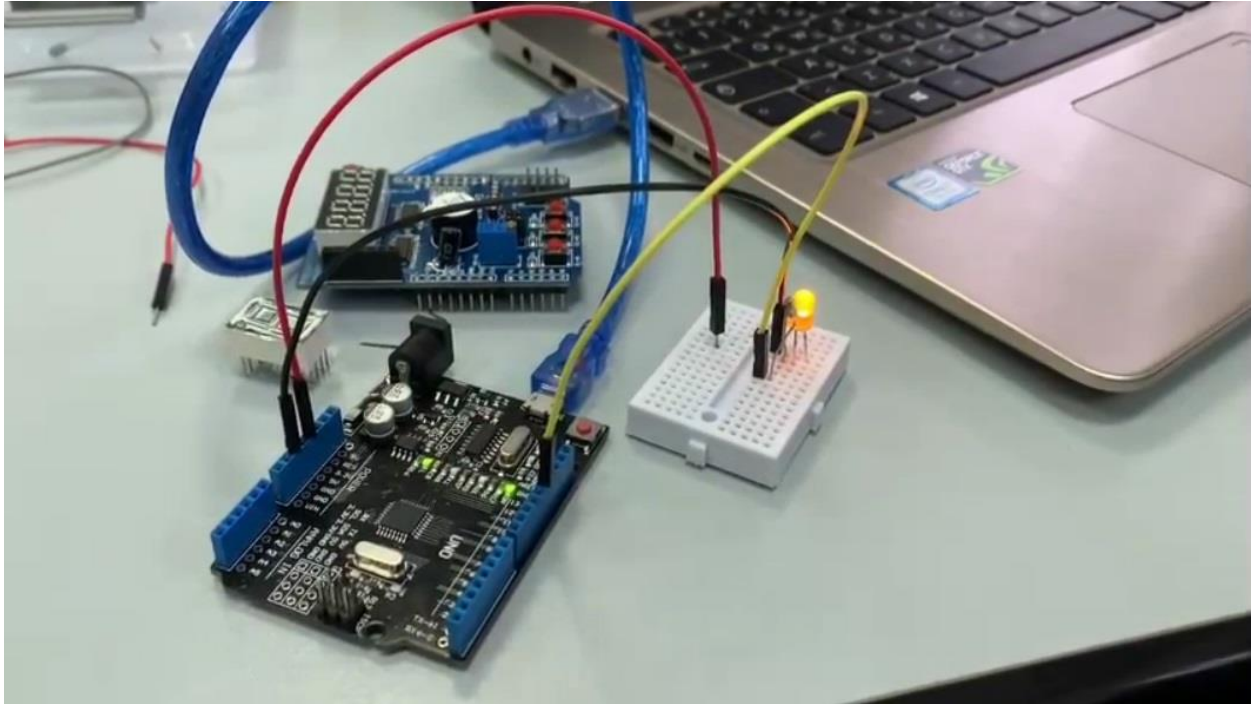


Рис. 3. Наш проект «Блимаючий світлодіод».

Код для реалізації такої схеми (рис. 4) вважається стандартним, і його можна знайти у бібліотеці стандартних скетчів.

```
void setup()
{
    pinMode(13, OUTPUT);          //робимо 13-ий пін виходом
}

void loop()
{
    digitalWrite(13, HIGH);       //подається напруга, світлодіод загоряється
    delay(1000);                  //затримка
    digitalWrite(13, LOW);        //напруга перетворюється на нуль
    delay(1000);                  //затримка
}
```

Рис. 4. Програма для реалізації проекту «Блимаючий світлодіод».

4. Проект «Світлофор»

Набір елементів для побудови світлофора з окремих світлодіодів мало відрізняється від списку для попереднього проекту. Необхідно взяти світлодіоди трьох різних кольорів (червоного, жовтого и зеленого) і підключити за тією ж схемою, підключивши кожен до свого піна. Нам пощастило отримати готовий елемент зі світлодіодами та опором, який ми напряму під'єднували до плати Arduino.

Далі найголовнішим було написати програму.

Для програмування світлофора було покладено такий порядок його роботи:

- Червоний – 6 секунд
- Червоний разом із жовтим – 3 секунди
- Зелений – 7 секунд
- Блімаючий зелений – 3 секунди
- Жовтий – 3 секунди

Програма вийшла такою, як показано на рис. 5.

```
void setup()
{
    pinMode(13, OUTPUT);          //вихід для червоного
    pinMode(12, OUTPUT);          //вихід для жовтого
    pinMode(11, OUTPUT);          //вихід для зеленого
}

void loop()
{
    digitalWrite(13, HIGH);        //запалюється червоний
    delay(6000);
    digitalWrite(12, HIGH);        //запалюється жовтий
    delay(3000);
    digitalWrite(13, LOW);         //вимикається червоний
    digitalWrite(12, LOW);         //вимикається жовтий
    digitalWrite(11, HIGH);        //запалюється зелений
    delay(7000);
    digitalWrite(11, LOW);
    delay(500);
    for(int i = 0; i < 3; i++)
    {
        digitalWrite(11, HIGH);    //блімання зеленого
        delay(500);
        digitalWrite(11, LOW);
        delay(500);
    }
    digitalWrite(12, HIGH);        //запалюється жовтий
    delay(3000);
    digitalWrite(12, LOW);         //вимикається жовтий
}
```

Рис. 5. Програма до проекту «Світлофор».

Не забуваємо про раціоналізацію програми. Наприклад, у програмі було використано цикл `for`. Які переваги його використання? Порівняємо два варіанти програми на рис. 6. Окрім очевидного скорочення програми це допомагає і її сприйняттю. За нескінченним повторюваним кодом, як у першому варіанті на рис. 6 втрачається розуміння сенсу того, що в ньому відбувається. Звісно, в нашому випадку програма невелика, і такі перетворення несуттєві, але у випадку значніших проєктів ці можливості є дуже корисними.

<pre> void setup() { pinMode(13, OUTPUT); //тут присутні жакливі повторення, так робити не треба pinMode(12, OUTPUT); pinMode(11, OUTPUT); } void loop() { digitalWrite(13, HIGH); delay(6000); digitalWrite(12, HIGH); delay(3000); digitalWrite(13, LOW); digitalWrite(12, LOW); digitalWrite(11, HIGH); delay(7000); digitalWrite(11, LOW); delay(500); digitalWrite(11, HIGH); delay(500); digitalWrite(11, LOW); //жакливі повторення delay(500); digitalWrite(11, HIGH); delay(500); digitalWrite(11, LOW); delay(500); digitalWrite(11, HIGH); delay(500); digitalWrite(11, LOW); delay(500); digitalWrite(12, HIGH); delay(3000); digitalWrite(12, LOW); } </pre>	<pre> void setup() { pinMode(13, OUTPUT); pinMode(12, OUTPUT); pinMode(11, OUTPUT); } void loop() { digitalWrite(13, HIGH); delay(6000); digitalWrite(12, HIGH); delay(3000); digitalWrite(13, LOW); digitalWrite(12, LOW); digitalWrite(11, HIGH); delay(7000); digitalWrite(11, LOW); delay(500); for(int i = 0; i < 3; i++) { digitalWrite(11, HIGH); delay(500); digitalWrite(11, LOW); delay(500); } digitalWrite(12, HIGH); delay(3000); digitalWrite(12, LOW); } </pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Рис. 6. Порівняння програм з та без використання циклу.

Тепер можемо поглянути на результат роботи. Перейти до відео можна за посиланням:

https://github.com/npd2020/electronics/blob/master/gr5b/Olexandra_Hrytseniak/lab_3/lab_3_2.mp4.

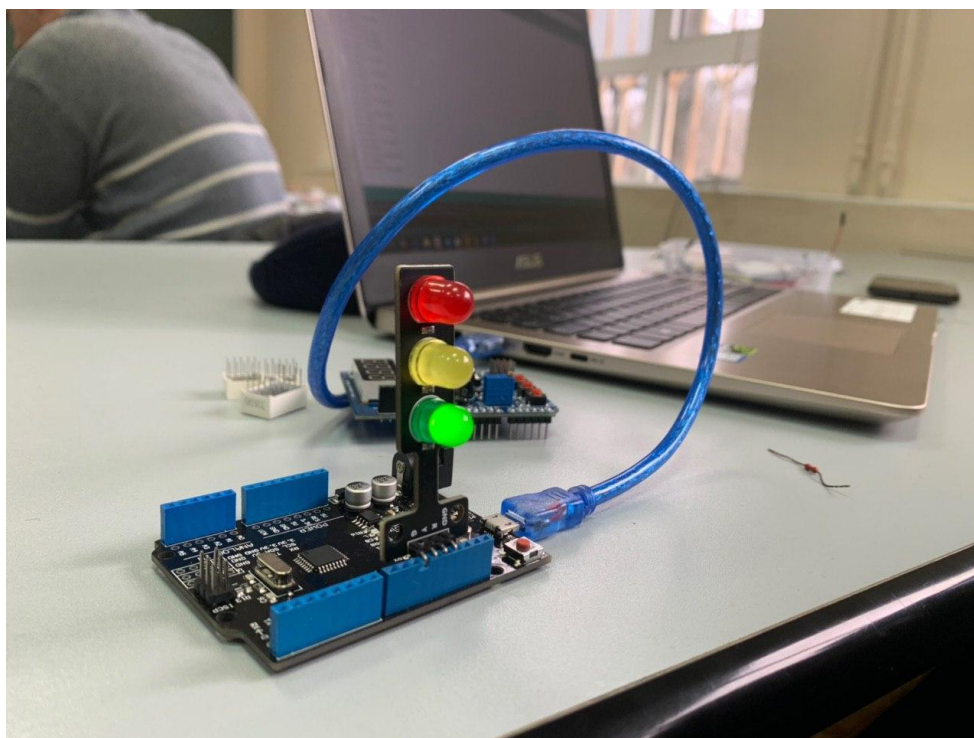


Рис. 7. Проект «Світлофор».

5. Висновок

Ми ознайомилися з принципом роботи плати Arduino і навчилися писати програми для керування нею у відповідному середовищі розробки. Код виявився похідним від C групи, тому при роботі не виникло проблем, структура та базові елементи програми було давно знайомими.

Arduino – це дуже зручний інструмент для роботи з електронними схемами, але потужний у плані можливостей. Розглянуті проекти є найпростішим, що можна зробити з його допомогою, а легкість у розумінні і роботі доводять, що з ним можна реалізовувати і ще набагато цікавіші та складніші ідеї.