

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

**Навчально-науковий інститут атомної та теплової енергетики  
Кафедра цифрових технологій в енергетиці**

## **ЗВІТ**

**з лабораторної роботи №3**

**з дисципліни «Розробка застосунків інтернету речей та  
сенсорних мереж»**

**Тема: «Ознайомлення з поняттям мікроконтролера і його  
основними функціями, створення базової схеми»**

**Варіант №17**

**Виконав:**

**Студент групи ТР-12**

**Ковальов Олександр Олексійович**

**Дата здачі: 03.03.2025**

**Мета роботи.** Розробка і програмування мікроконтролерної системи для керування світлодіодами з використанням послідовності Фібоначчі та реалізація простого цифрового замка на основі мікроконтролера з введенням пароля через кнопку клавіатури.

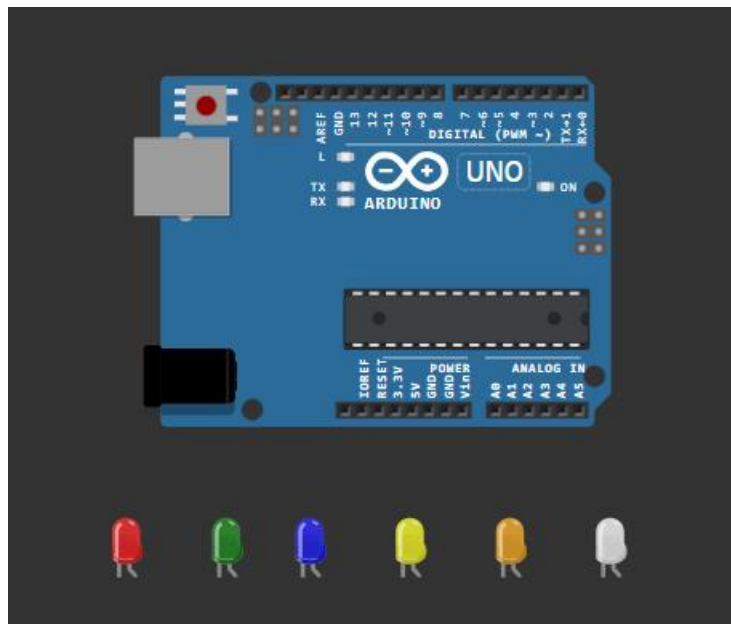
**Індивідуальне завдання:**

- 1) Дослідити поняття мікроконтролера та ознайомитися з його властивостями і можливостями, використовуючи доступну літературу та ресурси.
- 2) Для першої частини завдання – створити схему з одним мікроконтролером і шістьма LED-лампочками, які будуть під'єднані до різних портів контролера. Для даної схеми необхідно написати програмний код для мікроконтролера для послідовного ввімкнення і вимикання лампочок з інтервалом в 1 секунду, що відповідають їхньому розташуванню у двійковому коді відповідно до послідовності чисел Фібоначчі, які не перевищують значення вашого номера варіанта. Наприклад: якщо ваш варіант — 10, то останнє значення числа Фібоначчі буде 8 — ввімкнені лампочки будуть: 3 (оскільки 8 у двійковому коді 001000). Відповідно, для *17 варіанту* – число 13 (8 число зі списку), 13 в двійковій формі – 001101.
- 3) Для другої частини завдання – створити схему з одним мікроконтролером, 10 кнопками (switch) та 1 LED-лампочкою. Для даної схеми написати програмний код для мікроконтролера, який відслідковує стан кнопок і порівнює їх із введеним паролем (номером варіанта). Наприклад: якщо ваш варіант — 10, то при ввімкнених кнопках тільки 1 та 0 буде вмикатися лампочка. Відповідно, для *17 варіанту* – треба щоб були увімкнені кнопки 10 та 7.

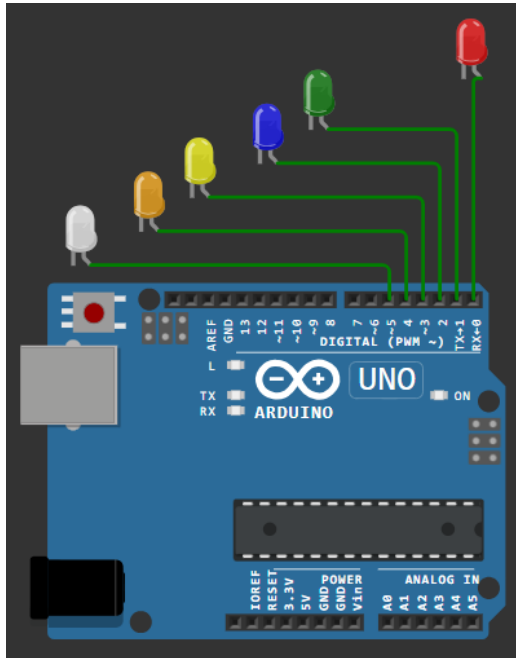
**Хід роботи.**

Робота буде відбуватись на сайті з <https://wokwi.com> платою Arduino Uno.

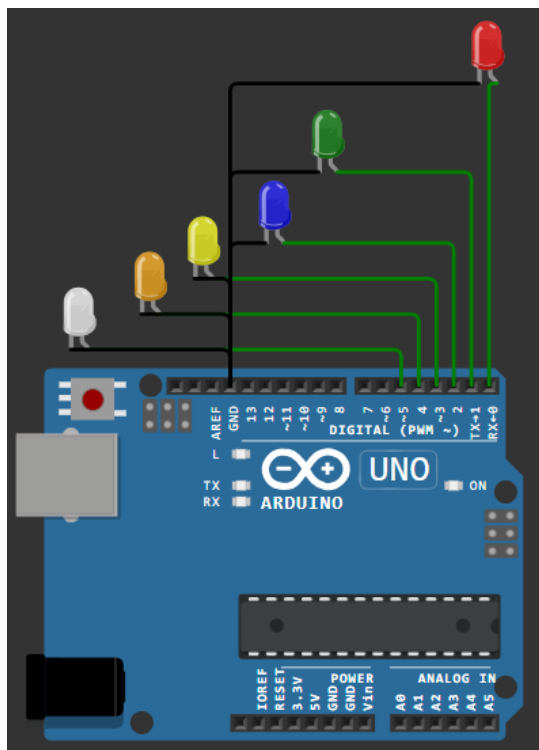
Були додані 6 LED ламп на робочу поверхню:



Були підключені лампи (порт A) до плати (порти з 0 до 5):



Також, вони були підключені до порту GND.



Був написаний код для першого завдання:

```
int VAR = 17;

void setup() {
    // No need to do anything here for now
}

int fibonacci(int n) {
    if (n <= 1) {
```

```

        return n;
    } else {
        return fibonacci(n - 1) + fibonacci(n - 2);
    }
}

void output(int n) {
    for (int i = 0; n > 0; i++) {
        pinMode(i, OUTPUT);
        if (n % 2) {
            digitalWrite(i, 1);
        }
        n /= 2;
    }

    delay(500);

    for (int i = 0; i < 8; i++) {
        digitalWrite(i, 0);
    }

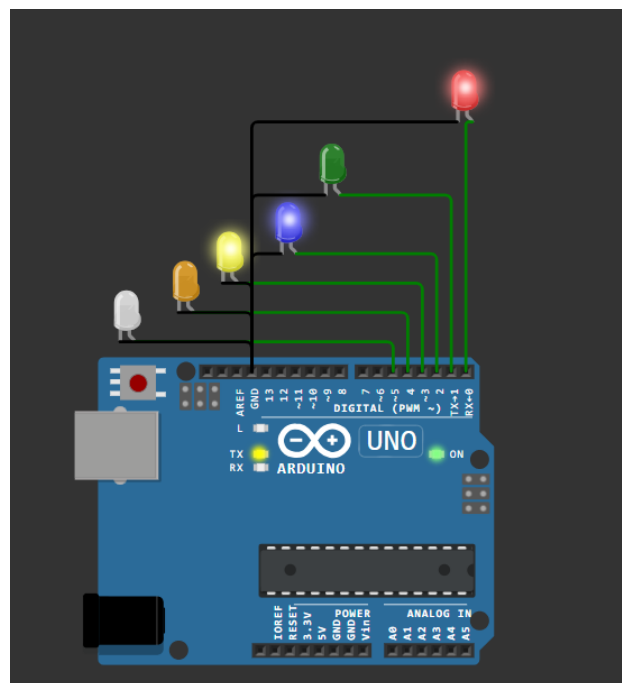
    delay(500);
}

void loop() {
    static int i = 1; // Keep track of Fibonacci index
    int fib = fibonacci(i);

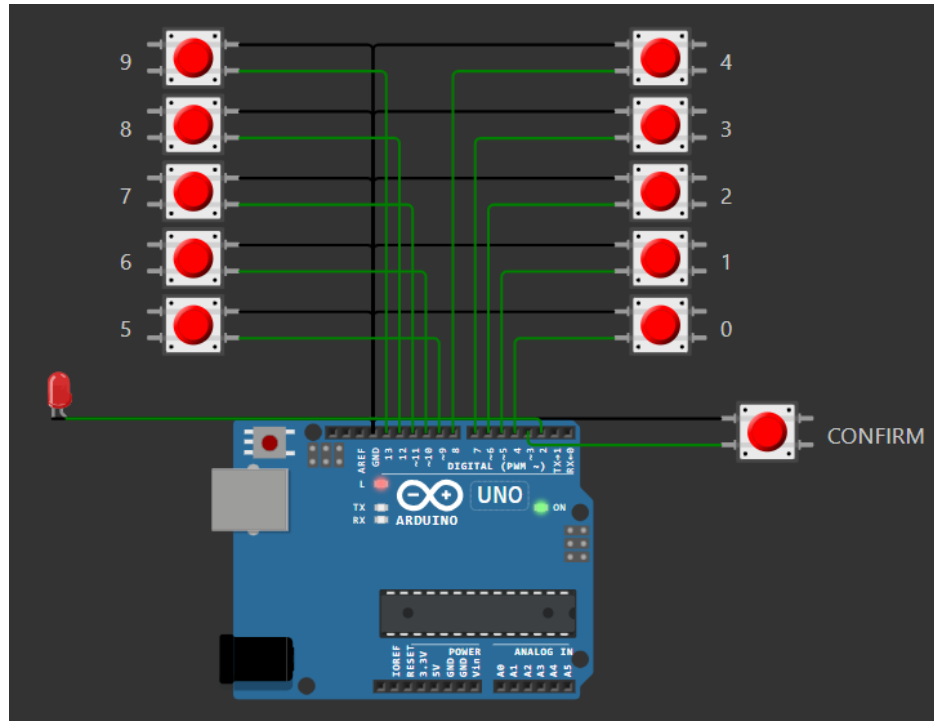
    if (fib > VAR) {
        i = 1; // Restart when exceeding VAR
    } else {
        output(fib);
        i++;
    }
}

```

Результат (лампочки горять послідовно по всім позиціям чисел Фібоначчі, до 13):



В другому завданні треба було реалізувати механізм «замку з паролем». Тому, було додано 10 кнопок, кнопка підтвердження та LED-лампа.



Код:

```
#define LED_PIN 2
#define BUTTONS_AMOUNT 10
#define CONFIRM_BUTTON_PIN 3

const int password = 17;
const int buttonPins[BUTTONS_AMOUNT] = {4, 5, 6, 7, 8, 9, 10, 11, 12, 13};
bool password_bin[BUTTONS_AMOUNT];
bool buttonState[BUTTONS_AMOUNT];

void setup() {
    pinMode(LED_PIN, OUTPUT);
    pinMode(CONFIRM_BUTTON_PIN, INPUT_PULLUP);

    for (int i = 0; i < BUTTONS_AMOUNT; i++) {
        pinMode(buttonPins[i], INPUT_PULLUP);
    }

    // Setting up password
    int tens = password / 10;
    int units = password % 10;
    if (tens > 0) {
        password_bin[tens] = true;
    }
    password_bin[units] = true;
}

void loop() {
    read_state();

    if (digitalRead(CONFIRM_BUTTON_PIN) == LOW) {
        bool isCorrect = check_state();
    }
}
```

```

    digitalWrite(LED_PIN, isCorrect ? HIGH : LOW);
    if (isCorrect) {
        delay(500);
        digitalWrite(LED_PIN, LOW);
    }
    clear_state();
}

void read_state() {
    for (int i = 0; i < BUTTONS_AMOUNT; i++) {
        if (!buttonState[i]) {
            buttonState[i] = (digitalRead(buttonPins[i]) == LOW);
        }
    }
}

bool check_state() {
    for (int i = 0; i < BUTTONS_AMOUNT; i++) {
        if (buttonState[i] != password_bin[i]) {
            return false;
        }
    }
    return true;
}

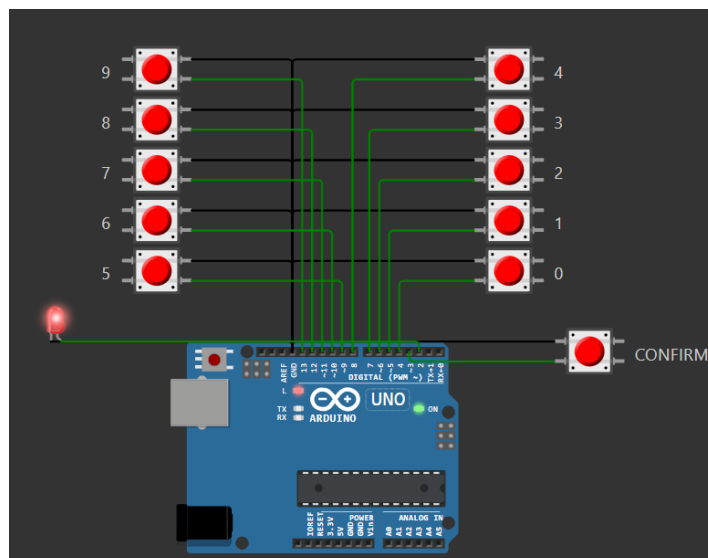
void clear_state() {
    for (int i = 0; i < BUTTONS_AMOUNT; i++) {
        buttonState[i] = false;
    }
}

```

Встановлюються режими пінів, на вхід та вихід. Так як вхід PULLUP, то результати будуть інвертовані. Далі, встановлюється пароль в «бінарному вигляді», тобто заповнюється масив bool значень.

Після цього, в циклі оновлюються значення. Якщо значення true, то воно не оновлюється. Якщо натиснута кнопка confirm, перевіряємо стан, чи сходиться з паролем. Якщо так, то вмикаємо лампочку, і після невеликої затримки вимикаємо. Далі очищаємо стан.

Стан лампочки, якщо натиснути на 1, 7 і Confirm:



**Висновок:** У процесі виконання лабораторної роботи було успішно розроблено мікроконтролерну систему для керування світлодіодами та реалізовано цифровий замок на основі мікроконтролера. Перш за все, дослідження мікроконтролера дозволило ознайомитись з його властивостями та можливостями, що стало основою для подальшої реалізації обох частин завдання. У першій частині створено схему з шістьма світлодіодами, яка працює згідно з послідовністю чисел Фібоначчі. Кожен світлодіод вмикається або вимикається в залежності від двійкової форми числа Фібоначчі, що відповідає номеру варіанту. У другій частині було створено схему з кнопковою клавіатурою для введення пароля, який перевіряється мікроконтролером і вмикає світлодіод при правильному введенні пароля. Усі завдання були успішно реалізовані, що підтверджує ефективність використаних методів та можливостей мікроконтролера.

### **Контрольні питання:**

#### *1. Що таке IoT? Яка архітектура IoT?*

IoT (Інтернет речей) – це концепція, за якою фізичні об'єкти, обладнання або пристрої можуть збирати, обмінюватися та аналізувати дані через Інтернет. Архітектура IoT зазвичай складається з трьох основних рівнів: пристрої для збору даних (сенсори та актуатори), мережевий рівень для передачі даних (наприклад, через Wi-Fi, Bluetooth або мобільні мережі), а також платформи обробки даних, що здійснює аналіз та прийняття рішень.

#### *2. Як застосовується мікроконтролер в понятті IoT?*

Мікроконтролери є основою багатьох IoT-пристроїв, оскільки вони забезпечують обробку даних, взаємодію з сенсорами та актуаторами, а також комунікацію з іншими пристроями або сервісами через мережу. Мікроконтролер отримує дані від сенсорів, обробляє їх і, при необхідності, передає або виконує дію через підключені пристрої.

#### *3. З чого складається мікроконтролер?*

Мікроконтролер складається з процесора (CPU), пам'яті (ROM, RAM), входів/виходів (I/O порти), тактового генератора та периферійних модулів, таких як АЦП, ЦАП або комунікаційні інтерфейси. Завдяки таким компонентам мікроконтролер може виконувати різні завдання від збору інформації до управління іншими пристроями.

#### *4. Яким чином відбувається взаємодія мікроконтролера з іншими пристроями?*

Взаємодія мікроконтролера з іншими пристроями відбувається через його виводи та периферійні інтерфейси. Це може бути цифровий або аналоговий вхід/вихід, серійні інтерфейси, такі як UART, SPI або I2C, або бездротові зв'язки, як Wi-Fi чи Bluetooth. Мікроконтролер може отримувати дані від зовнішніх пристроїв, передавати їх на інші системи або керувати підключеними актуаторами, виконуючи відповідні дії на основі обробленої інформації.