

Блимання світлодіода

Завдання: підключити світлодіод до Arduino Uno й налаштувати блимання — 0,5 секунди світиться, 0,5 секунди вимкнений.

Компоненти

- Arduino Uno — 1 плата та USB-кабель для неї.
- Макетна плата — 1 шт.
- Червоний світлодіод — 1 шт.
- Резистор **220 Ом** — 1 шт.
- Два з'єднувальні дроти: зелений і чорний.

На схемі використано зелений і чорний дроти. Можна взяти інші кольори.

Змінюй підключення лише з від'єднанням USB. У симуляторі перед змінами натискай «Стоп».

Контакти макетної плати

Кожен отвір позначений номером і літерою, наприклад **4j**.

П'ять отворів **4a–4e** з'єднані всередині. Окремо з'єднані **4f–4j**. Між цими групами — канавка, електричного зв'язку немає.

Отже, **4j** і **4i** з'єднані, а **4i** і **4e** — ні. Сусідні номери, наприклад 3 і 4, теж не з'єднані.

На іншій макетній платі розмітка може бути повернута. Орієнтуйся на літери, номери й канавку, а не лише на «верх» та «низ».

Чи з'єднані отвори **3a** та **3e**?

Так, вони в одній групі а–е з номером 3. А **3e** та **4e** — ні.

Фрагмент макетної плати · вигляд згори



Підключення піна 13

З'єднай пін **13** Arduino із отвором **4j** зеленим дротом.

Пояснення

Програма вмикатиме й вимикатиме напругу на пині **13**. Дріт з'єднує цей пін із макетною платою. **D13** — позначення цифрового піна **13**.

Перевір

Початок дроту — саме 13, а не 3 чи **5V**. Другий кінець — **4j**.

Поки коло розімкнене, світлодіод ще не може світитися.



Підключення резистора

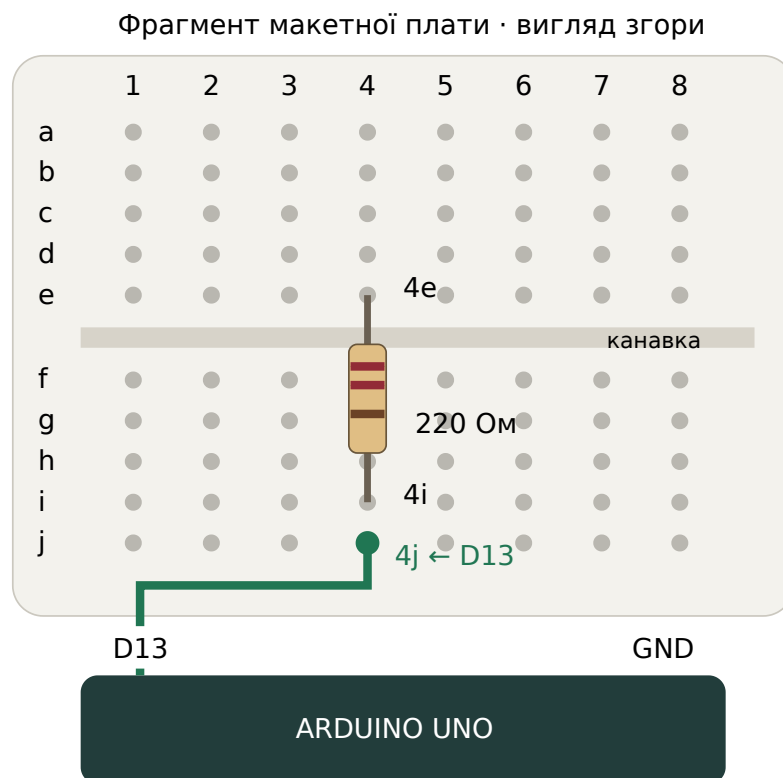
Встав резистор **220 Ом** між отворами **4i** та **4e**, через центральну канавку.

4i вже з'єднаний із зеленим дротом через **4j**. Резистор з'єднає дві половини плати й обмежить струм через LED.

Резистор можна повернути будь-яким кінцем. Для типового чотирисмугового 220 Ом: червона, червона, коричнева; четверта смуга позначає допуск.

Перевір

Ніжки стоять по різні боки канавки. Якщо обидві вставити в **4f–4j**, макетна плата з'єднає їх напряму й обійде резистор.



Підключення світлодіода

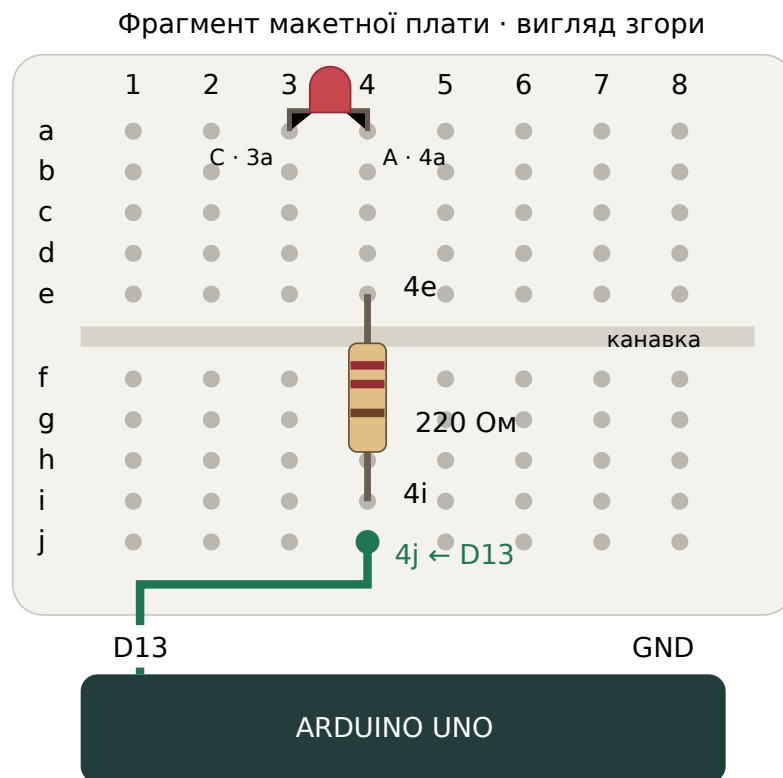
Встав довгу ніжку **A (анод)** у **4a**, а коротку **C (катод)** у **3a**.

У нового LED довша ніжка — анод. Плоска грань на корпусі зазвичай позначає катод. Якщо ніжки вже обрізані — перевір позначення з учителем.

4a з'єднаний із резистором через **4e**. Коротка ніжка у сусідньому номері 3 — це окрема група контактів.

Перевір

Ніжки у різних номерах: 4 та 3. На схемі A праворуч, C ліворуч.



Підключення GND

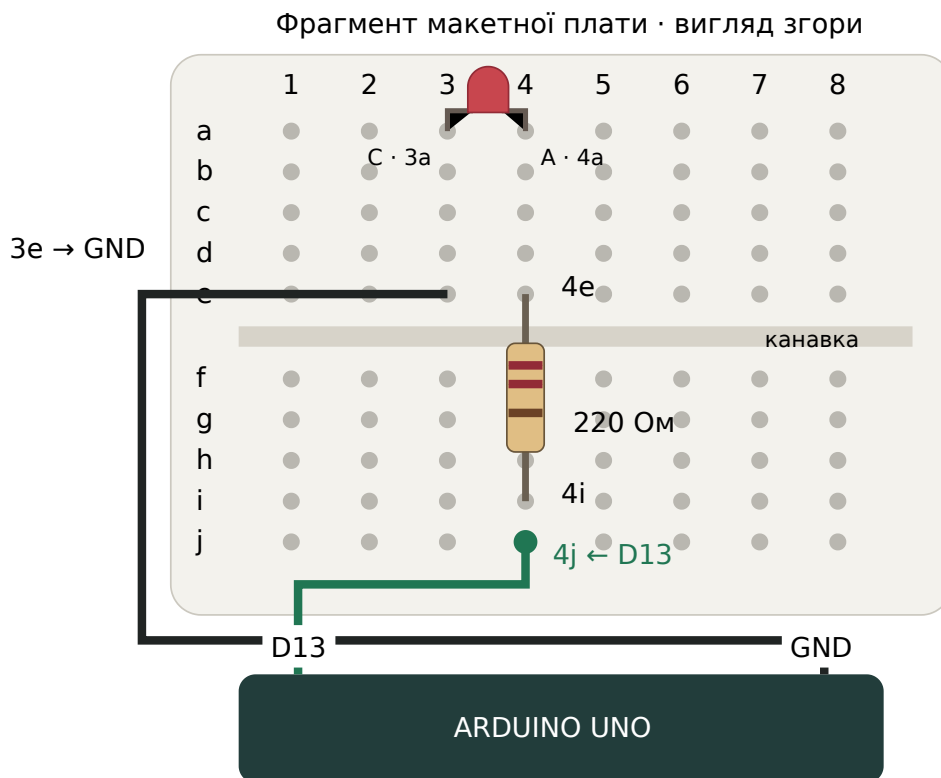
З'єднай **3e** з **GND** Arduino чорним дротом.

3e з'єднаний із короткою ніжкою LED у **3a**. **GND** — спільна точка відліку напруги й зворотний шлях струму в цьому колі.

D13 → **4j** → **4i** → **резистор** → **4e** → **4a** → **LED** → **3a** → **3e** → **GND**

Перед увімкненням

Перевір усі підключення за схемою. Струм має проходити через резистор і світлодіод, а потім повертатися до **GND**.



Налаштування піна в коді

Введи цей код у редакторі. Він налаштовує пін **13** як вихід. Команди блимання додамо у **loop()** на наступному кроці.

```
#include <Arduino.h> // Підключаємо команди Arduino.

const int LED_PIN = 13; // Зберігаємо номер піна світлодіода.

void setup() { // Один раз після запуску.
  pinMode(LED_PIN, OUTPUT); // Пін 13 працює як вихід.
}

void loop() { // Тут згодом будуть повторювані дії.
}
```

const int — ціле число, яке не змінюємо. **LED_PIN** — назва константи з номером піна. **OUTPUT** означає «вихід».

Фігурні дужки об'єднують команди. Крапка з комою завершує команду. Текст після **//** — пояснення для людини.

Увімкнення, пауза, вимкнення

Заміни порожню `loop()` цим фрагментом. `setup()` залиш.

```
void loop() { // Повторюємо від першої команди після останньої.  
  digitalWrite(LED_PIN, HIGH); // Увімкнути LED.  
  delay(500); // Пів секунди світла.  
  digitalWrite(LED_PIN, LOW); // Вимкнути LED.  
  delay(500); // Пів секунди без світла.  
}
```

`HIGH` дає високий рівень напруги, `LOW` — низький. `delay()` задає паузу в мілісекундах: **1000 мс = 1 с**. Під час паузи стан виходу зберігається.

Скільки триває один цикл блимання?

Приблизно 1 секунду: 500 мс світла + 500 мс темряви. Час виконання інших команд тут дуже малий.

Запуск програми

У Wokwi

Відкрий <https://wokwi.com/projects/new/arduino-uno>. Додай макетну плату, червоний LED і резистор 220 Ом кнопкою «+». Збери схему за кроками 3–6 цього документа. У вкладку `sketch.ino` введи повний код з останньої сторінки. Натисни зелену кнопку запуску.

На справжній платі

1. З від'єднаним USB звір коло зі схемою. Попроси вчителя перевірити підключення.
2. Під'єднай Uno через USB. Розпакуй архів коду. В Arduino IDE відкрий `sketch/sketch.ino`, вибери Arduino Uno та порт плати.
3. Натисни «Завантажити» й дочекайся завершення.

Перевірка роботи

Зовнішній LED світиться 0,5 с і гасне на 0,5 с. Вбудований індикатор L на Uno також може блимати: він під'єднаний до того самого піна **13**.

Блимає лише індикатор L на платі

Код працює. Від'єднай USB і перевір зовнішнє коло: А у **4a**, С у **3a**, резистор **4i–4e**, чорний дріт **3e** – **GND**.

Не запускається програма

У симуляторі перевір, чи натиснуто запуск. Якщо є помилка компіляції — звір код, дужки, великі літери й крапки з комою. На платі перевір вибір порту, плати та завершення завантаження.

Зміна тривалості блимання

Зроби так, щоб LED світився **2 секунди**, а не світився **0,5 секунди**.

1. Знайди паузу після команди ввімкнення LED.
2. Зміни тільки цю паузу. Запусти програму повторно.
3. Покажи результат учителю та поясни, чому друга пауза залишилася без змін.

Пояснення до завдання

Після HIGH потрібно написати `delay(2000);`. Після LOW залишається `delay(500);`. Повний цикл триватиме приблизно 2,5 секунди.

Що показати вчителю

- Зібране коло із резистором 220 Ом.
- Програму, що блимає за новим ритмом.
- Пояснення: навіщо резистор, де анод і яка команда гасить LED.

Збережи змінений проєкт як **Мій світлодіод**. Якщо працюєш у Wokwi — збережи його у своєму обліковому записі.

Світлодіод, що блимає

Світлодіод світиться 0,5 секунди й залишається вимкненим 0,5 секунди.

```
// Світлодіод світиться 0,5 секунди й не світиться 0,5 секунди.  
#include <Arduino.h> // Підключаємо команди керування Arduino.  
  
const int LED_PIN = 13; // Даємо піну 13 зрозуміле ім'я; його номер не змінюється.  
  
void setup() { // Виконується один раз після увімкнення або перезапуску плати.  
  pinMode(LED_PIN, OUTPUT); // Налаштовуємо пін світлодіода як вихід.  
}  
  
void loop() { // Arduino повторює ці дії, поки працює програма.  
  digitalWrite(LED_PIN, HIGH); // Подаємо високий рівень на пін 13: LED світиться.  
  delay(500); // Чекаємо 500 мілісекунд (0,5 секунди), зберігаючи світло.  
  digitalWrite(LED_PIN, LOW); // Подаємо низький рівень на пін 13: LED гасне.  
  delay(500); // Чекаємо 0,5 секунди перед наступним увімкненням.  
}
```