

Кнопка та світлодіод

Завдання: LED світиться, поки кнопка натиснута, і гасне після відпускання.

Компоненти

- Arduino Uno та USB-кабель.
- Макетна плата.
- Червоний світлодіод і резистор 220 Ом.
- Тактова кнопка з чотирма ніжками.
- Три з'єднувальні дроти та одна коротка перемичка.

Використаємо коло зі світлодіодом із першого уроку. До нього додамо кнопку між піном 2 та **GND**.

Перед зміною підключень від'єднай USB. У Wokwi зупини симуляцію.

Підключення світлодіода

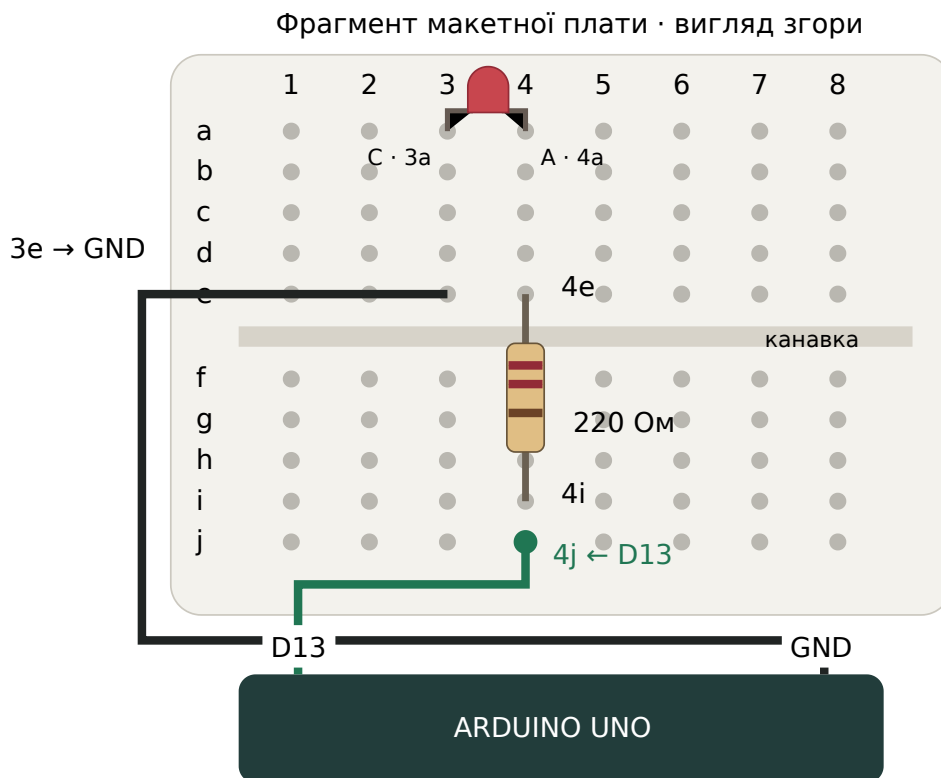
Збери коло з першого уроку або перевір уже зібране.

1. **D13** → **4j**.
2. Резистор 220 Ом → **4i** та **4e**.
3. Довга ніжка LED → **4a**, коротка → **3a**.
4. Чорний дріт → **3e** та **GND**.

Резистор обмежує струм через LED.

Перевір

Обидві ніжки LED мають бути в різних контактних групах: 3 та 4.



Чотири ніжки, дві пари

У цій схемі ніжки кнопки утворюють дві пари:

- **6e** та **6j** з'єднані постійно.
- **8e** та **8j** з'єднані постійно.

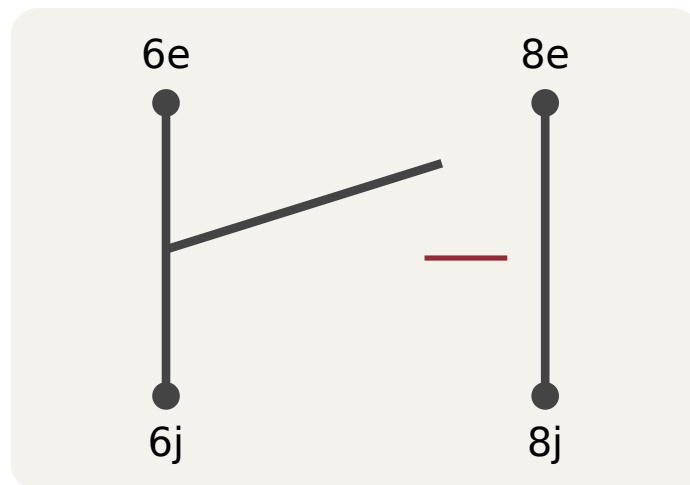
Коли натискаємо кнопку, вона з'єднує ці пари між собою. Після відпускання зв'язок між парами зникає.

У справжньої кнопки перевір пари мультиметром у режимі прозвонювання разом з учителем. Поворот корпусу змінює розташування пар на макетній платі.

Що буде, якщо обидва дроти під'єднати до однієї пари?

Вони будуть з'єднані й без натискання. Arduino постійно зчитуватиме стан «натиснуто».

З'єднання всередині кнопки



Натискання замикає проміжок між парами

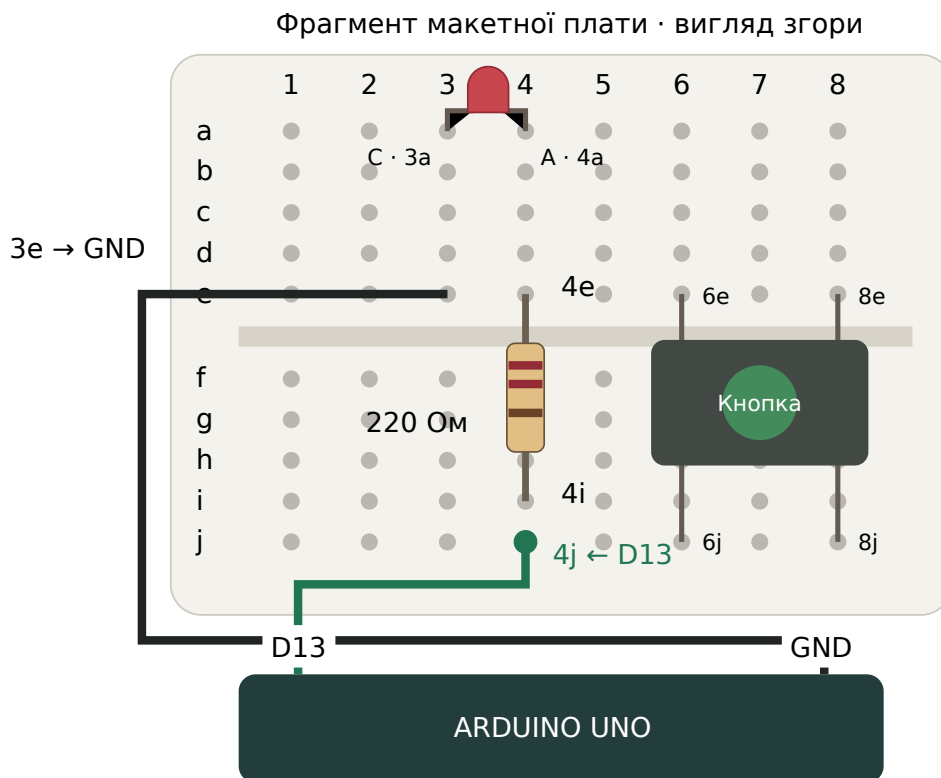
Кнопка на макетній платі

Встанови кнопку в отвори **6e**, **8e**, **6j** та **8j**, як на схемі.

Ці координати відповідають кнопці у нашому Wokwi-проекті. На справжній платі відстань між ніжками може відрізнятись. Не згинай їх силою: підбери отвори разом з учителем, зберігши ті самі електричні з'єднання.

Перевір

Один контакт кнопки має з'єднувати групи **6e** і **6j**, другий — **8e** і **8j**. Саме положення над канавкою не гарантує правильного підключення.



Перемичка до кнопки

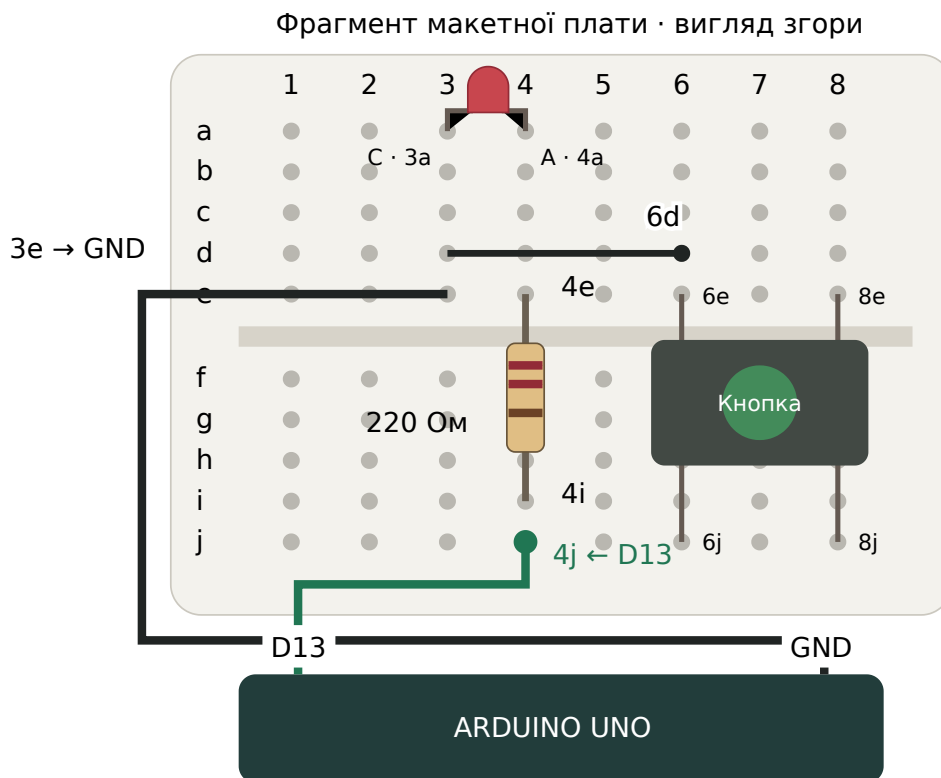
З'єднай **3d** і **6d** короткою чорною перемичкою.

3d вже з'єднаний із **GND** через групу **3a–3e** та чорний дрід у **3e**. Отвір **6d** з'єднаний із ніжкою кнопки в **6e**.

Так один контакт кнопки отримує з'єднання з **GND**.

Перевір

Перемичка веде з номера 3 до номера 6. Вона не повинна потрапити в групу 4, де під'єднаний LED.



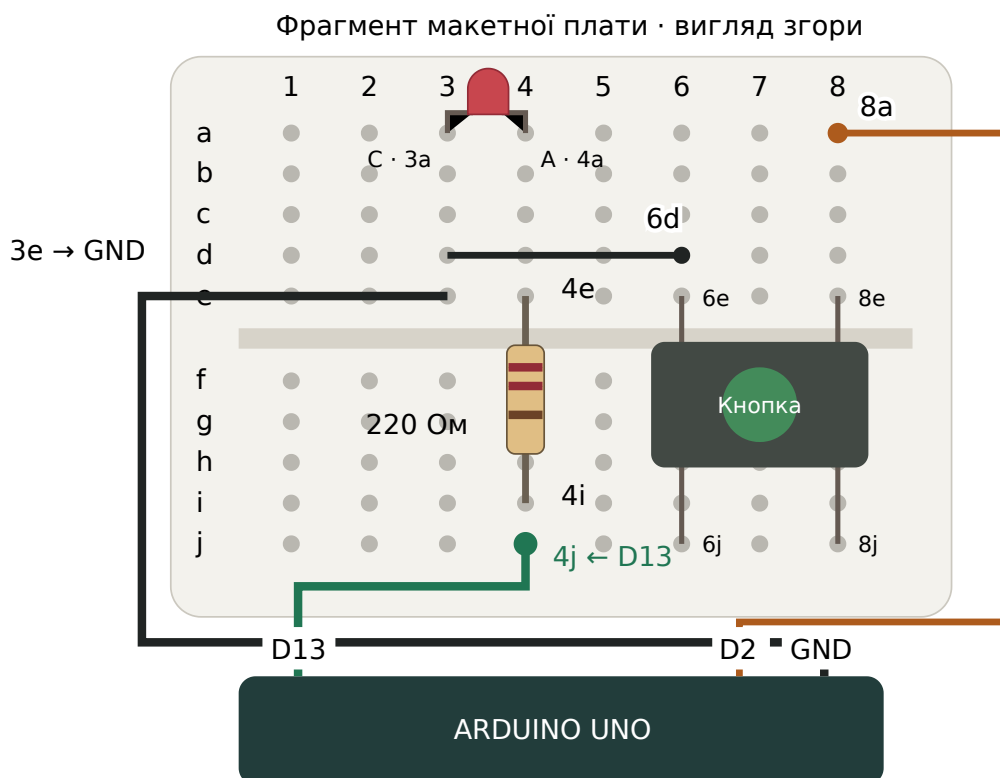
Пін 2 до кнопки

З'єднай пін **2** Arduino з отвором **8a** помаранчевим дротом.

8a з'єднаний із ніжкою кнопки в **8e**. Після натискання утворюється шлях:

D2 → **8a** → **8e** → **кнопка** → **6e** → **6d** → **3d** → **3e** → **GND**

Пін 2 працюватиме як вхід. Не налаштовуй його як **OUTPUT**: ця кнопка з'єднує його напряму з **GND**.



Як Arduino визначає натискання

`INPUT_PULLUP` вмикає внутрішній резистор між входом і живленням плати. Він задає визначений стан входу, коли кнопка відпущена.

Кнопка відпущена: пін `2` підтягнутий до живлення — `HIGH`.

Кнопка натиснута: пін `2` з'єднаний із `GND` — `LOW`.

Тому в цій схемі перевіряємо саме `LOW`. Окремий резистор для кнопки не потрібен. Резистор 220 Ом у колі LED залишається.

```
const int BUTTON_PIN = 2; // Номер входу кнопки.  
const int LED_PIN = 13; // Номер виходу світлодіода.  
  
void setup() {  
  pinMode(BUTTON_PIN, INPUT_PULLUP); // Вхід із підтягуванням.  
  pinMode(LED_PIN, OUTPUT); // Вихід для LED.  
}
```

Це фрагмент налаштування. Повна програма доступна в блоці «Код і файли».

Зчитування стану кнопки

```
void loop() {  
    bool pressed = digitalRead(BUTTON_PIN) == LOW; // Чи натиснута кнопка?  
    digitalWrite(LED_PIN, pressed); // Увімкнути або вимкнути LED.  
    delay(10); // Зачекати 10 мілісекунд перед наступним зчитуванням.  
}
```

`digitalRead()` читає рівень на вході. Порівняння `== LOW` дає `true`, коли кнопку натиснуто.

`bool` зберігає одне з двох значень: `true` або `false`. У команді `digitalWrite()` вони відповідають увімкненню та вимкненню LED.

Чому LED не блимає самостійно?

Програма щоразу встановлює його стан за кнопкою. Пауза 10 мс не перемикає вихід — вона лише затримує наступну перевірку.

Повідомлення про натискання

У повній програмі додано `Serial.begin(115200)` у `setup()`. Монітор порту відкрий на швидкості **115200**.

```
static bool wasPressed = false; // Попередній стан кнопки.  
  
if (pressed && !wasPressed) { // Натиснута зараз, раніше – ні.  
    Serial.println("Натиснули!"); // Повідомлення в монітор порту.  
}  
  
wasPressed = pressed; // Оновлюємо попередній стан.
```

Цей фрагмент у повному коді стоїть після керування LED. `static` зберігає змінну між викликами `loop()`, `&&` означає «і», а `!` — «не».

При утриманні кнопки повідомлення не повторюється на кожному проході циклу.

У нашій схемі Wokwi брязкіт контактів вимкнений. На справжній кнопці одне натискання іноді дає кілька повідомлень. Цей код ще не містить повного усунення брязкоту.

Перевірка роботи

Запуск

Відкрий <https://wokwi.com/projects/new/arduino-uno>. Додай компоненти зі списку й з'єднай їх за кроками 2–6. Введи повний код з останньої сторінки у sketch.ino й запусти симуляцію.

1. Натисни й утримуй кнопку. LED має світитися. Відпусти — LED має згаснути.
2. Відкрий монітор порту: на початку натискання має з'явитися «Натиснули!».

Для справжньої плати спочатку перевір коло з учителем, потім під'єднай USB і завантаж sketch.ino через Arduino IDE, вибравши Uno та порт.

LED світиться весь час

Зупини симуляцію або від'єднай USB. Перевір, чи **D2** та **GND** не потрапили на одну постійно з'єднану пару кнопки.

LED не світиться при натисканні

Перевір **D2–8a**, перемичку **3d–6d**, полярність LED та резистор. У коді має бути INPUT_PULLUP і порівняння з LOW.

Самостійне завдання

Зміни програму так, щоб кнопка **вимикала** LED: без натискання він світиться, під час натискання — ні. Підключення залиш ті самі.

Пояснення після спроби

Заміни команду керування LED на `digitalWrite(LED_PIN, !pressed);`. Оператор **!** змінює true на false і навпаки.

Покажи результат учителю й поясни, чому натискання дає LOW.

Кнопка та світлодіод

Кнопка натиснута — LED увімкнений; відпущена — вимкнений.

```
// Кнопка натиснута – світлодіод увімкнений; відпущена – вимкнений.  
#include <Arduino.h> // Підключаємо команди Arduino.  
  
const int BUTTON_PIN = 2; // Пін, до якого під'єднана кнопка.  
const int LED_PIN = 13; // Пін, до якого під'єднаний світлодіод.  
  
void setup() { // Виконується один раз після запуску плати.  
  pinMode(BUTTON_PIN, INPUT_PULLUP); // Вхід із внутрішнім резистором підтягування.  
  pinMode(LED_PIN, OUTPUT); // Налаштовуємо пін світлодіода як вихід.  
  Serial.begin(115200); // Відкриваємо послідовний порт зі швидкістю 115200 біт/с.  
}  
  
void loop() { // Повторюємо перевірку стану кнопки.  
  bool pressed = digitalRead(BUTTON_PIN) == LOW; // LOW означає, що кнопка натиснута.  
  digitalWrite(LED_PIN, pressed); // true вмикає LED, false вимикає.  
  
  static bool wasPressed = false; // Зберігаємо попередній стан між викликами loop.  
  
  if (pressed && !wasPressed) { // Кнопка натиснута зараз, але раніше була відпущена.  
    Serial.println("Натиснули!"); // Друкуємо повідомлення на початку натискання.  
  }  
  
  wasPressed = pressed; // Запам'ятовуємо стан для наступної перевірки.  
  delay(10); // Пауза 10 мс між перевітками; це не повне усунення брязкоту.  
}
```