

Світлофор із кнопкою

Продовжуємо урок 05-01: додаємо кнопку запиту пішохода. Без запиту горить зелений для машин; натискання запускає перехід до червоного.

- Arduino Uno та USB-кабель.
- Червоний, жовтий і зелений LED; три резистори по 220 Ом.
- Кнопка, макетна плата, дроти «тато-тато».

Спочатку пройди 05-01 — автоматичний світлофор. Навчимося зчитувати кнопку й відраховувати час за допомогою `millis()`.

Це модель сигналів для автомобілів. Збираємо три незалежні гілки LED і повторюємо кнопку з уроку 2.

Перед зміною схеми від'єднай живлення. У Wokwi зупини симуляцію. Справжню схему перевір разом з учителем перед підключенням USB.

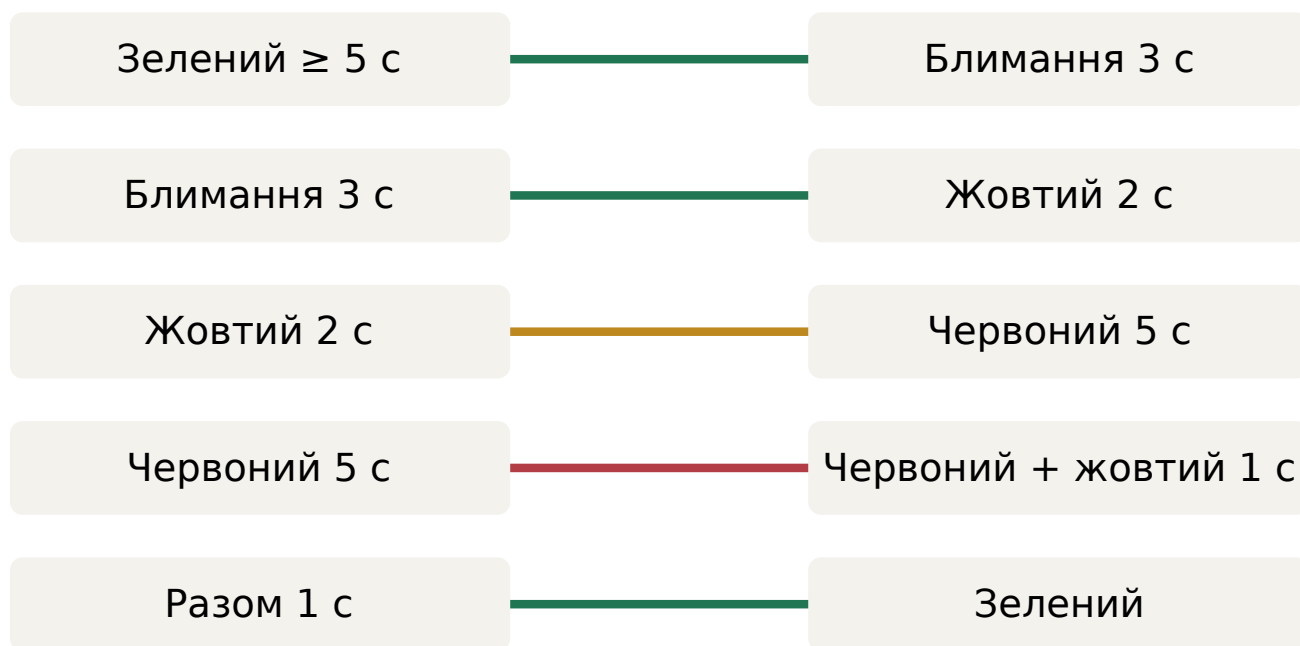
П'ять станів світлофора

1. **Зелений.** Горить щонайменше 5 секунд. Без натискання залишається зеленим.
2. **Зелений блимає.** Тричі: 0,5 с вимкнений і 0,5 с увімкнений. Разом 3 секунди.
3. **Жовтий.** Горить 2 секунди.
4. **Червоний.** Горить 5 секунд.
5. **Червоний і жовтий.** Горять разом 1 секунду, потім знову зелений.

Коротке натискання запам'ятовується. Якщо натиснути на другій секундi зеленого, блимання почнеться на п'ятій секундi, жовтий — на восьмій. Якщо натиснути після п'ятої — блимання почнеться майже одразу.

Натискання під час інших кольорів не додають нових циклів.

Послідовність після запиту



Електричні з'єднання · не масштаб плати

Окремий резистор для кожного LED

На макетній платі отвори а-е одного номера з'єднані. Різні номери — окремі контактні групи.

1. Червоний LED: анод А → **4a**, катод С → **3a**. Резистор 220 Ом → **4c–10c**.
2. Жовтий: А → **13a**, С → **12a**. Резистор → **13c–19c**.
3. Зелений: А → **22a**, С → **21a**. Резистор → **22c–28c**.

Довга ніжка нового LED зазвичай є анодом. Плаский край корпусу позначає катод. Резистор не має полярності.

У Wokwi деталі стоять у цих самих отворах. Резистори розташовані в рядку c; кожен з'єднує дві різні контактні групи.

Три виходи та спільний GND

1. **D8** → **10d** : струм іде через резистор до червоного LED.
2. **D9** → **19d** : живить жовту гілку.
3. **D10** → **28d** : живить зелену гілку.
4. **GND** Uno → нижня шина **–**. Від **3e**, **12e** та **21e** проведи по короткому чорному дроту до цієї шини.

Так усі три катоди мають спільний **GND**. Початок кожної гілки під'єднаний до окремого виходу Arduino. Не з'єднуй між собою **D8**, **D9** та **D10** : кожен вихід керує своїм кольором.

LED · через резистор 220 Ом



Електричні з'єднання · не масштаб плати

Натискання з'єднує D2 з GND

Постав кнопку ліворуч через центральну канавку: ніжки в **6e**, **8e**, **6j** та **8j**. У Wokwi вона повернута на 90°. Перевір, які контакти з'єднані без натискання, за уроком 2 або мультиметром на знеструмленій схемі.

1. **D2** → **8j**: ця група з'єднана з контактом кнопки 1.r у **8j**.
2. **6j** → нижня шина **—**: це контакт 2.r іншої внутрішньої групи.

У цій схемі використовуємо **1.r** і **2.r**. **1.l** та **1.r** уже з'єднані всередині, тому між ними натискання нічого не змінить.

З **INPUT_PULLUP** відпущена кнопка дає HIGH, натиснута — LOW. Додатковий резистор для кнопки не потрібний.

Кнопка · дві різні групи



Електричні з'єднання · не масштаб плати

Коли змінюється сигнал

```
void updatePhase(unsigned long now) {
    unsigned long elapsed = now - phaseStartedAt; // Тривалість поточного стану.

    if (phase == GREEN && crossingRequested && elapsed >= GREEN_MS) { // Можна
        завершити зелений.
        phase = BLINK; // Спершу три блимання зеленого.

    } else if (phase == BLINK && elapsed >= BLINK_MS) { // Три блимання
        завершилися.
        phase = YELLOW; // Після них вмикаємо жовтий.

    } else if (phase == YELLOW && elapsed >= YELLOW_MS) { // Жовтий завершився.
        phase = RED; // Вмикаємо червоний.

    } else if (phase == RED && elapsed >= RED_MS) { // Час переходу завершився.
        phase = READY; // Червоний разом із жовтим.

    } else if (phase == READY && elapsed >= READY_MS) { // Повертаємо рух.
        phase = GREEN; // Знову зелений.
        crossingRequested = false; // Старий запит виконано.

    } else { // Час зміни ще не настав.
        return; // Залишаємо поточний стан.
    }

    phaseStartedAt = now; // Запам'ятовуємо початок нового стану.
}
```

`phase` зберігає один із п'яти станів: GREEN, BLINK, YELLOW, RED або READY. `crossingRequested` означає, що пішохід натиснув кнопку. `elapsed` — час від початку поточного стану.

У кожному рядку `if` перевіряємо стан і час. Якщо переходити ще рано, `return` завершує тільки цю функцію. Після переходу запам'ятовуємо новий час у `phaseStartedAt`.

Три дії в loop()

```
void loop() {  
    unsigned long now = millis(); // Один показ часу для всіх дій цього оберту.  
    readButton(now); // Читаємо кнопку та запам'ятовуємо новий запит.  
    updatePhase(now); // Перемикаємо стан, якщо настав його час.  
    showLights(now - phaseStartedAt); // Показуємо кольори та оновлюємо  
    блимання.  
}
```

1. `readButton(now)` приймає зміну кнопки лише після 30 мс стабільного читання. Нове натискання на зеленому зберігає запит; утримання не створює нових запитів.
2. `updatePhase(now)` визначає, чи час перейти до наступного сигналу.
3. `showLights(now - phaseStartedAt)` встановлює виходи відповідно до поточного стану.

У `showLights` змінна `flashOn` чергується між `false` і `true` кожні 500 мс. Вираз `(elapsed / HALF_BLINK_MS) % 2` дає 0, 1, 0, 1...: 0 — темно, 1 — світло. Стан BLINK триває 3 секунди, тому зелений блимає тричі.

Довгих `delay` немає: усі три функції швидко завершуються, і `loop` знову читає кнопку. Повний код кожної функції є в блоці «Код і файли».

Перевір порядок кольорів

Запуск у Wokwi

Натисни кнопку запуску. Для самостійного завдання збережи власну копію проєкту.

Готовий проєкт: <https://wokwi.com/projects/475801822614727681>. Схему можна зібрати вручну за кроками уроку. Повний код — на останній сторінці.

1. Запусти й зачекай понад 5 секунд без натискання: має залишатися зелений.
2. Натисни й відпусти кнопку. Очікуй три блимання зеленого за 3 с, жовтий 2 с, червоний 5 с, два кольори разом 1 с, зелений.
3. Перезапусти. Натисни після 1–2 секунд зеленого: зміна має зачекати до п'ятої секунди.
4. Утримуй кнопку весь цикл: другий цикл сам не починається.

Перевір окремі гілки

Кнопка не запускає цикл

Перевір **D2** і дві різні контактні групи. Натискай на зеленому; він має тривати мінімум 5 секунд.

Колір не відповідає стану

Знеструм схему й збір **D8** — червоний, **D9** — жовтий, **D10** — зелений. Перевір полярність LED і його окремий резистор.

Чому червоний із жовтим горять одночасно?

Це передбачений стан READY тривалістю 1 секунду, після якого вмикається зелений.

Збільш тривалість червоного до 8 секунд

Зміни лише `RED_MS` так, щоб червоний горів 8 секунд.

1. Запиши потрібне число мілісекунд.
2. Зміни код, перезапусти та натисни кнопку.
3. Перевір секундоміром червоний і переконайся, що жовтий усе ще триває 2 секунди.

Відповідь після перевірки

`RED_MS = 8000`. Інші тривалості не змінюємо. Від початку жовтого до повернення зеленого тепер проходить приблизно 11 секунд.

Світлофор із кнопкою

Запит із кнопки запускає послідовність сигналів без довгих delay.

```
#include <Arduino.h> // Команди плати Arduino.

const byte RED_PIN = 8; // Червоний LED.
const byte YELLOW_PIN = 9; // Жовтий LED.
const byte GREEN_PIN = 10; // Зелений LED.
const byte BUTTON_PIN = 2; // Кнопка замикає D2 на GND.
const unsigned long GREEN_MS = 5000; // Мінімум 5 секунд зеленого.
const unsigned long BLINK_MS = 3000; // Три блимання по одній секунді.
const unsigned long HALF_BLINK_MS = 500; // Пів секунди темно, пів секунди світло.
const unsigned long YELLOW_MS = 2000; // 2 секунди жовтого.
const unsigned long RED_MS = 5000; // 5 секунд для переходу.
const unsigned long READY_MS = 1000; // 1 секунда червоного з жовтим.
const unsigned long DEBOUNCE_MS = 30; // Чекаємо стабільного контакту.
enum Phase { GREEN, BLINK, YELLOW, RED, READY }; // Імена п'яти станів.
Phase phase = GREEN; // Починаємо із зеленого.
bool crossingRequested = false; // Чи є запит на перехід.
bool lastReading = HIGH; // Попереднє читання кнопки.
bool buttonState = HIGH; // Стабільний стан кнопки.
unsigned long buttonChangedAt = 0; // Коли змінився контакт кнопки.
unsigned long phaseStartedAt = 0; // Коли почався поточний стан.

void showLights(unsigned long elapsed) {
    digitalWrite(RED_PIN, phase == RED || phase == READY); // Червоний у двох станах.
    digitalWrite(YELLOW_PIN, phase == YELLOW || phase == READY); // Жовтий у двох станах.
    bool flashOn = (elapsed / HALF_BLINK_MS) % 2 == 1; // Чергуємо темну та світлу
    половини секунди.
    bool greenOn = phase == GREEN || (phase == BLINK && flashOn); // Зелений горить
    постійно або блимає.
    digitalWrite(GREEN_PIN, greenOn); // Передаємо обчислений стан на зелений LED.
}

void setup() { // Підготовка після ввімкнення.
    pinMode(RED_PIN, OUTPUT); // Вихід червоного LED.
    pinMode(YELLOW_PIN, OUTPUT); // Вихід жовтого LED.
    pinMode(GREEN_PIN, OUTPUT); // Вихід зеленого LED.
    pinMode(BUTTON_PIN, INPUT_PULLUP); // Відпущена кнопка дає HIGH.
    phaseStartedAt = millis(); // Запам'ятовуємо початок зеленого.
    showLights(0); // Показуємо початковий стан.
}

void readButton(unsigned long now) {
    bool reading = digitalRead(BUTTON_PIN); // Читаємо контакт.

    if (reading != lastReading) { // Контакт щойно змінився.
        lastReading = reading; // Зберігаємо нове читання.
        buttonChangedAt = now; // Починаємо відлік стабілізації.
    }
}
```

```

}

if (now - buttonChangedAt >= DEBOUNCE_MS && reading != buttonState) { // Стан
стабільний.
    buttonState = reading; // Приймаємо зміну кнопки.

    if (buttonState == LOW && phase == GREEN) { // Нове натискання на зеленому.
        crossingRequested = true; // Зберігаємо запит навіть після відпускання.
    }

}
}

void updatePhase(unsigned long now) {
    unsigned long elapsed = now - phaseStartedAt; // Тривалість поточного стану.

    if (phase == GREEN && crossingRequested && elapsed >= GREEN_MS) { // Можна завершити
зелений.
        phase = BLINK; // Спершу три блимання зеленого.

    } else if (phase == BLINK && elapsed >= BLINK_MS) { // Три блимання завершилися.
        phase = YELLOW; // Після них вмикаємо жовтий.

    } else if (phase == YELLOW && elapsed >= YELLOW_MS) { // Жовтий завершився.
        phase = RED; // Вмикаємо червоний.

    } else if (phase == RED && elapsed >= RED_MS) { // Час переходу завершився.
        phase = READY; // Червоний разом із жовтим.

    } else if (phase == READY && elapsed >= READY_MS) { // Повертаємо рух.
        phase = GREEN; // Знову зелений.
        crossingRequested = false; // Старий запит виконано.

    } else { // Час зміни ще не настав.
        return; // Залишаємо поточний стан.
    }

    phaseStartedAt = now; // Запам'ятовуємо початок нового стану.
}

void loop() {
    unsigned long now = millis(); // Один показ часу для всіх дій цього оберту.
    readButton(now); // Читаємо кнопку та запам'ятовуємо новий запит.
    updatePhase(now); // Перемикаємо стан, якщо настав його час.
    showLights(now - phaseStartedAt); // Показуємо кольори та оновлюємо блимання.
}

```