

Потенціометр і сервопривід

Повертаємо ручку потенціометра — змінюється положення вала сервопривода.

- Arduino Uno, USB-кабель і макетна плата.
- Потенціометр із трьома контактами, наприклад B5K.
- Позиційний сервопривід, наприклад SG90.
- П'ять дротів для потенціометра та три з'єднання сервопривода.
- Для справжнього сервопривода — окреме стабілізоване джерело 5 В із достатнім струмом, підібране вчителем, і спільне з'єднання **GND**.

Спочатку збираємо схему Wokwi. Особливості живлення справжнього сервопривода розглянемо окремо.

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

Що змінює потенціометр

Потенціометр — змінний резистор із трьома контактами. У Wokwi вони підписані **VCC**, **SIG** та **GND**.

- **VCC** під'єднаємо до **5V**.
- **GND** — до **GND** Arduino.
- **SIG** — до аналогового входу **A0**.

Ручка переміщує внутрішній контакт. Напруга на **SIG** змінюється між напругами двох крайніх контактів. Таке з'єднання називають **подільником напруги**.

У справжнього потенціометра середня ніжка зазвичай є рухомим контактом. Звір його будову з учителем; написи й орієнтація можуть відрізнятися.



Електричні з'єднання · не масштаб плати

Спільні шини + та –

1. З'єднай **5V** Arduino з верхньою шиною **+**.
2. З'єднай **GND** Arduino з верхньою шиною **–**.

Верхня шина + має 5 В, а верхня шина – з'єднана з **GND**. Потенціометр і серво у Wokwi отримуватимуть живлення від цих шин окремими дротами, паралельно.

Не замикай + і – між собою. На фізичній платі перевір, чи немає розриву посередині шини; за потреби додай перемичку між її половинами.

Живлення на макетній платі



Електричні з'єднання · не масштаб плати

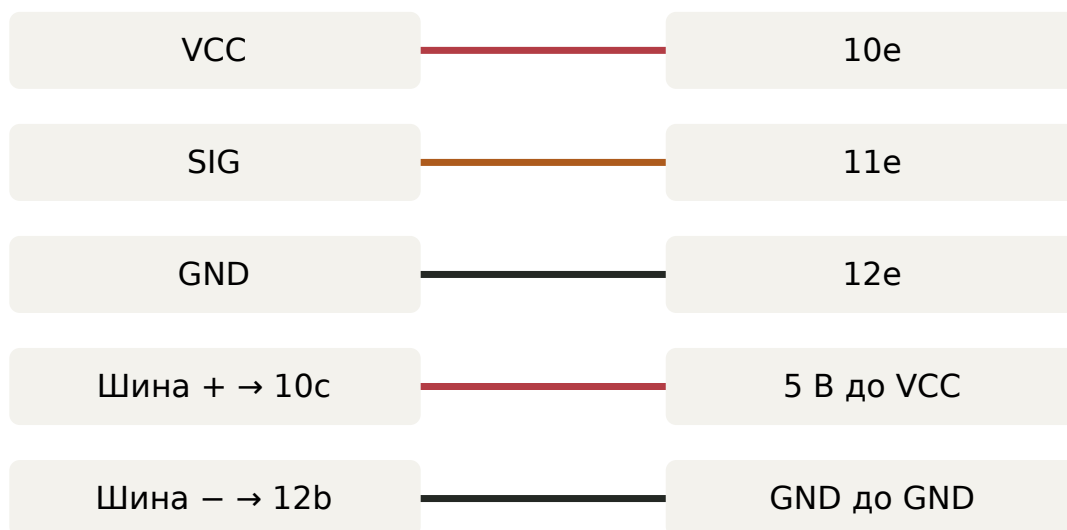
Встановлення та два дроти

1. Установи потенціометр: **VCC** → **10e**, **SIG** → **11e**, **GND** → **12e**. У Wokwi поверни його на 180°, як у готовій схемі.
2. З'єднай верхню шину + з **10c** червоним дротом.
3. З'єднай верхню шину – з **12b** чорним дротом.

10c та **10e** належать одній групі, як і **12b** та **12e**. Через ці групи потенціометр отримує живлення.

Усі три ніжки мають бути у різних номерах. Колір дроту допомагає перевірці, але не змінює з'єднання.

Потенціометр · координати



Електричні з'єднання · не масштаб плати

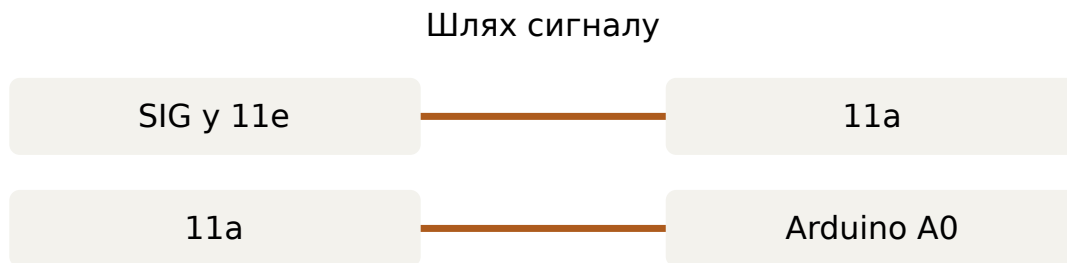
Сигнал на A0

З'єднай **11a** з **A0** Arduino.

11a з'єднаний із **SIG** у **11e**. Arduino вимірює напругу на цьому вході й перетворює її на число.

На Uno зі стандартними налаштуваннями `analogRead(A0)` повертає ціле число від **0 до 1023**. Для цієї схеми 0 відповідає напрузі близько 0 В, а 1023 — близько 5 В.

Середнє положення ручки дає приблизно 512. Далі перетворимо це число на кут повороту сервопривода.



Електричні з'єднання · не масштаб плати

Живлення та керування

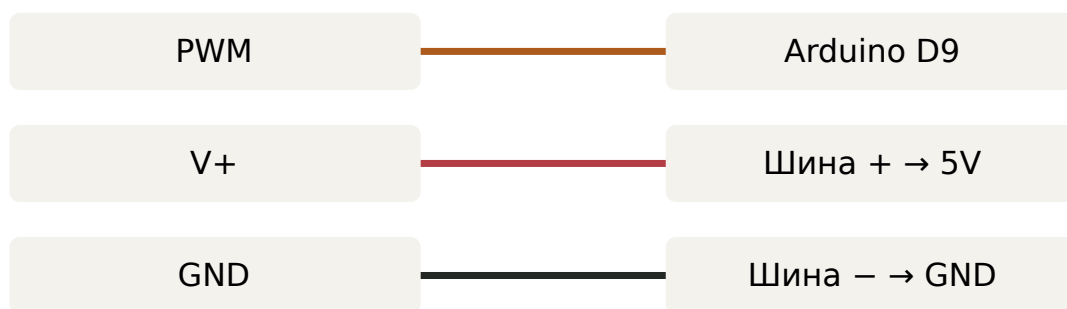
Сервопривід має два контакти живлення й один сигнальний.

1. Сигнал **PWM** → **D9** Arduino.
2. Живлення **V+** → верхня шина +.
3. **GND** → верхня шина —.

Це підключення для симулятора: шина + з'єднана з **5V**, а шина — — із **GND** Uno.

Через **D9** передаються керувальні імпульси. Живлення двигуна надходить через **V+**. Бібліотека **Servo** формує імпульси сама; позначка **~** біля піна для неї не обов'язкова.

Сервопривід · тільки схема Wokwi



Електричні з'єднання · не масштаб плати

Окреме живлення сервопривода

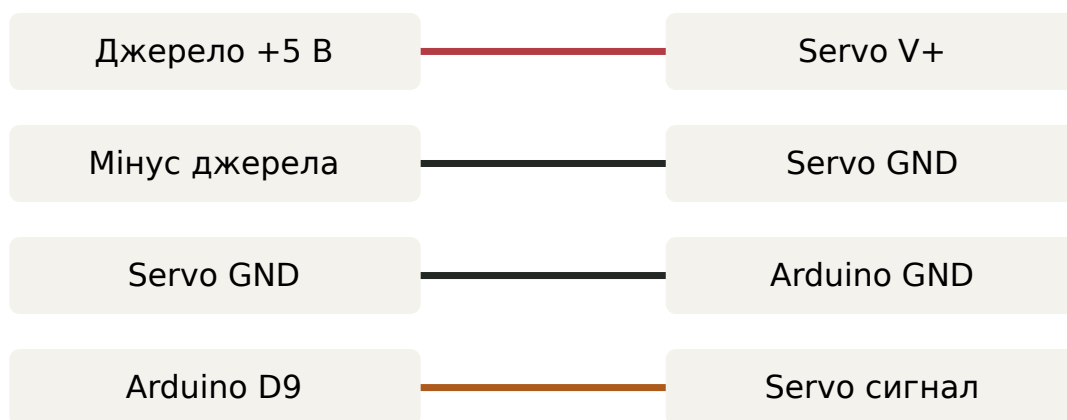
Сервопривід під час руху споживає більше струму, ніж вхід потенціометра. Нестача живлення може спричинити ривки та перезапуск Arduino.

1. Залиш потенціометр під'єднаним до **5V** Arduino.
2. Живлення **V+** сервопривода під'єднай до плюса окремого джерела 5 В, а не до шини + від Uno.
3. З'єднай **GND** сервопривода та мінус джерела з **GND** Arduino.
4. Сигнальний дрот залиш на **D9**.

Не з'єднуй плюс зовнішнього джерела з **5V** Arduino. Учитель має перевірити напругу, допустимий струм та призначення дротів саме вашого сервопривода.

Справжній механізм може не доходити до 0° та 180°. Якщо він упирається або гуде, зупини роботу й звузь діапазон.

Справжній сервопривід



Електричні з'єднання · не масштаб плати

Бібліотека та налаштування

```
#include <Arduino.h> // Команди Arduino.  
#include <Servo.h> // Керування сервоприводом.  
  
const int POT_PIN = A0; // Вхід потенціометра.  
const int SERVO_PIN = 9; // Сигнал сервопривода.  
Servo servo; // Об'єкт для керування валом.  
  
void setup() { // Один раз після запуску.  
  servo.attach(SERVO_PIN); // Сигнал через пін 9.  
  Serial.begin(115200); // Повідомлення в монітор порту.  
}
```

У Wokwi додай бібліотеку **Servo** через Library Manager. Інший спосіб: створи `libraries.txt` з одним рядком `Servo`. Цей файл уже є в ZIP уроку.

В Arduino IDE установи бібліотеку **Servo від Arduino** через менеджер бібліотек. `servo` — ім'я об'єкта; через нього викликаємо команди `attach()` і `write()`.

Від числа до градусів

```
int raw = analogRead(POT_PIN); // Вимірювання: 0–1023.  
int angle = map(raw, 0, 1023, 0, 180); // Кут: 0–180°.  
servo.write(angle); // Задаємо нове положення.  
delay(50); // Пауза перед наступним вимірюванням.
```

Ці команди виконуються всередині `loop()`. `map()` переносить число з одного діапазону в інший: $0 \rightarrow 0^\circ$, близько $512 \rightarrow$ близько 90° , $1023 \rightarrow 180^\circ$.

`write()` задає бажаний кут, але валу потрібен час, щоб повернутися. Під час `delay(50)` бібліотека продовжує надсилати керувальні імпульси.

У повному коді між поворотом і паузою стоять команди `Serial.print()` та `Serial.println()`: вони показують виміряне число і заданий кут.

Поворот та покази монітора

Запуск у Wokwi

Відкрий новий проєкт Arduino Uno. Скопіюй повний код із блока «Код і файли» у `sketch.ino`, а вміст завантаженої схеми — у `diagram.json`. Натисни кнопку запуску.

Адреса нового проєкту: <https://wokwi.com/projects/new/arduino-uno>. Схему можна зібрати вручну за кроками уроку. Повний код — на останній сторінці.

Перед запуском додай бібліотеку Servo, як у кроці 8. Відкрий монітор порту на швидкості 115200.

1. Поверни ручку в один край: очікуємо `pot=0` і `angle=0`.
2. У середньому положенні — приблизно `pot=512` і `angle=90`.
3. В іншому крайньому положенні — `pot=1023` і `angle=180`.

Покази змінюються, а вал стоїть

Перевір сигнал `D9`, три контакти сервопривода та Servo у списку бібліотек. На справжній платі додатково перевір живлення й спільний `GND`.

Самостійне завдання

Обмеж поворот діапазоном **30-150°**. Зміни тільки два числа в `map()`, перевір обидва краї та середину.

Пояснення після спроби

`map(raw, 0, 1023, 30, 150)`. Краї тепер 30° і 150°, середина — близько 90°. Вхідний діапазон 0-1023 залишаємо.

Потенціометр і сервопривід

Положення ручки задає кут сервопривода від 0° до 180°.

```
#include <Arduino.h> // Команди входів, виходів і монітора порту.
#include <Servo.h> // Бібліотека керування сервоприводом.

const int POT_PIN = A0; // Аналоговий вхід потенціометра.
const int SERVO_PIN = 9; // Сигнальний пін сервопривода.
Servo servo; // Об'єкт, через який задаємо положення вала.

void setup() { // Виконується один раз після запуску.
    servo.attach(SERVO_PIN); // Починаємо керувати сервоприводом через пін 9.
    Serial.begin(115200); // Відкриваємо монітор порту зі швидкістю 115200.
}

void loop() { // Повторюємо зчитування та зміну кута.
    int raw = analogRead(POT_PIN); // Читаємо значення від 0 до 1023.
    int angle = map(raw, 0, 1023, 0, 180); // Перетворюємо його на кут 0–180°.
    servo.write(angle); // Задаємо потрібне положення вала.
    Serial.print("pot="); // Підписуємо показ потенціометра.
    Serial.print(raw); // Виводимо прочитане число.
    Serial.print("  angle="); // Підписуємо заданий кут.
    Serial.println(angle); // Виводимо кут і переходимо на новий рядок.
    delay(50); // Чекаємо 50 мс перед наступним зчитуванням.
}
```