Від положення до руху

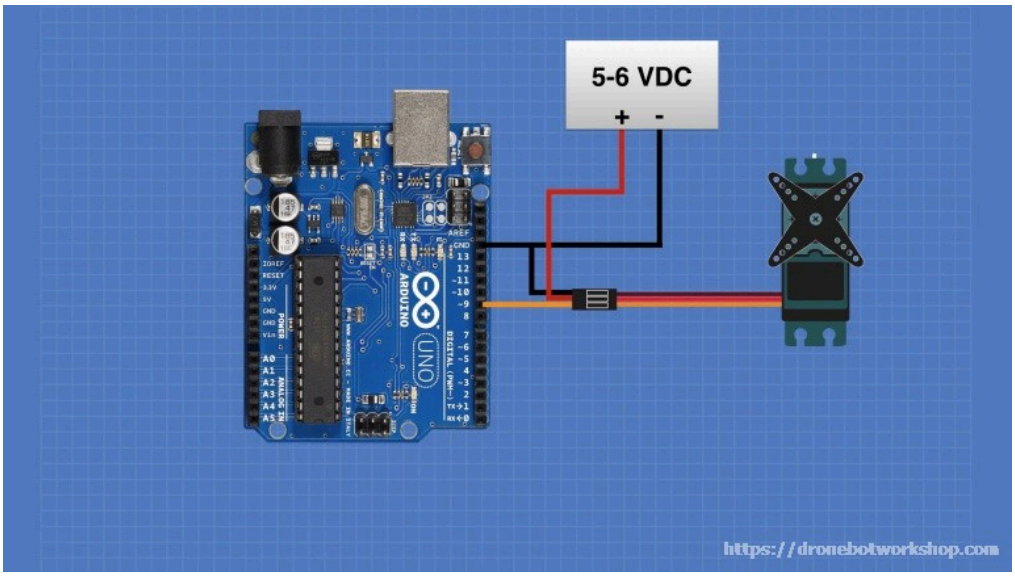
Як це працює

Великий стрибок і малі кроки

Порівняй дві послідовності: 60 → 120 та 60 → 61 → 62 → … → 120. Кінцеві положення однакові. У другому випадку Arduino багато разів змінює ціль, тому ми можемо розтягнути рух у часі.

Як це пов'язано зі Sweep

У статті приклад Sweep проходить діапазон двома циклами for. У нашій програмі один loop повторює малий крок, а змінна direction зберігає напрямок. Починаємо з обмеженого діапазону 60–120. Це послідовність команд; реальний двигун може трохи відставати.



Та сама схема — нова послідовність команд · DroneBot Workshop

Зроби

1. Повтори підключення уроку 09.
2. Залиш вал без навантаження.
3. Запусти програму двох положень.

Очікуваний результат

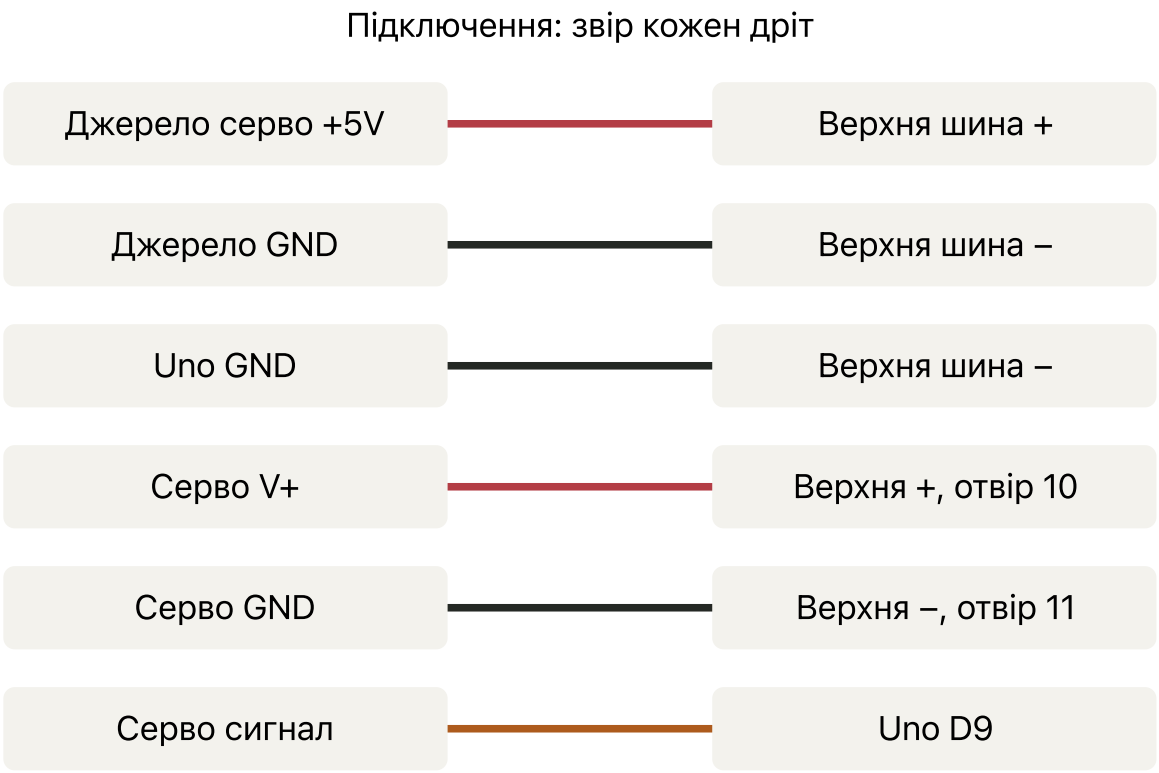
Маєш зразок для порівняння.

Що можна змінити

HOLD_MS залиш 2000.

У Wokwi: відкрий новий Arduino Uno, перенеси diagram.json, sketch/sketch.ino та libraries.txt з ZIP потрібного кроку у відповідні вкладки, натисни ▶. На справжній платі спочатку перевір підключення з учителем. Кожен код замінює весь попередній скетч.

Підключення справжнього комплекту



Два плюси розділені, GND спільний

На макетній платі верхня + — тільки джерело серво, нижня + — тільки Uno. Об’єднуємо мінусові шини. Перевір розриви шин; за потреби з’єднай половини тієї самої шини. У Wokwi верхня + живиться від Uno лише як спрощення. На справжній платі цей дріт заміни на +5 V окремого джерела; його мінус додай на верхню –.

Крок 1: код · 1

```
#include <Arduino.h>
#include <Servo.h>
const byte SERVO_PIN = 9;
const int ANGLE = 90; // Початковий кут без стріли
Servo arm;
const int FIRST_ANGLE = 60; // Перше положення
const int SECOND_ANGLE = 120; // Друге положення
const unsigned long HOLD_MS = 2000; // Час у кожному положенні

void setup() {
  arm.write(ANGLE); // Задаємо початкове положення
  arm.attach(SERVO_PIN); // Сигнал керування через D9
}

void loop() {
  arm.write(FIRST_ANGLE);
  delay(HOLD_MS);
  arm.write(SECOND_ANGLE);
  delay(HOLD_MS);
}
```

Рухайся по одному градусу

Як це працює

Пройди програму вручну

angle — остання задана ціль. Спочатку це 60. Програма надсилає її, чекає 30 мс і додає direction × STEP_DEG. За direction = 1 і STEP_DEG = 1 наступна ціль — 61, потім 62.

Звідки береться зворотний рух

Коли angle досягає MAX_ANGLE, програма встановлює direction = –1. Тепер додавання дає 119, 118 і далі. Біля MIN_ANGLE напрямок знову стає 1. Два if перевіряють межі після кожного кроку.

Зроби

1. Заміни код програмою цього кроку.
2. Простеж один повний рух туди й назад.
3. Знайди рядок, який змінює angle.

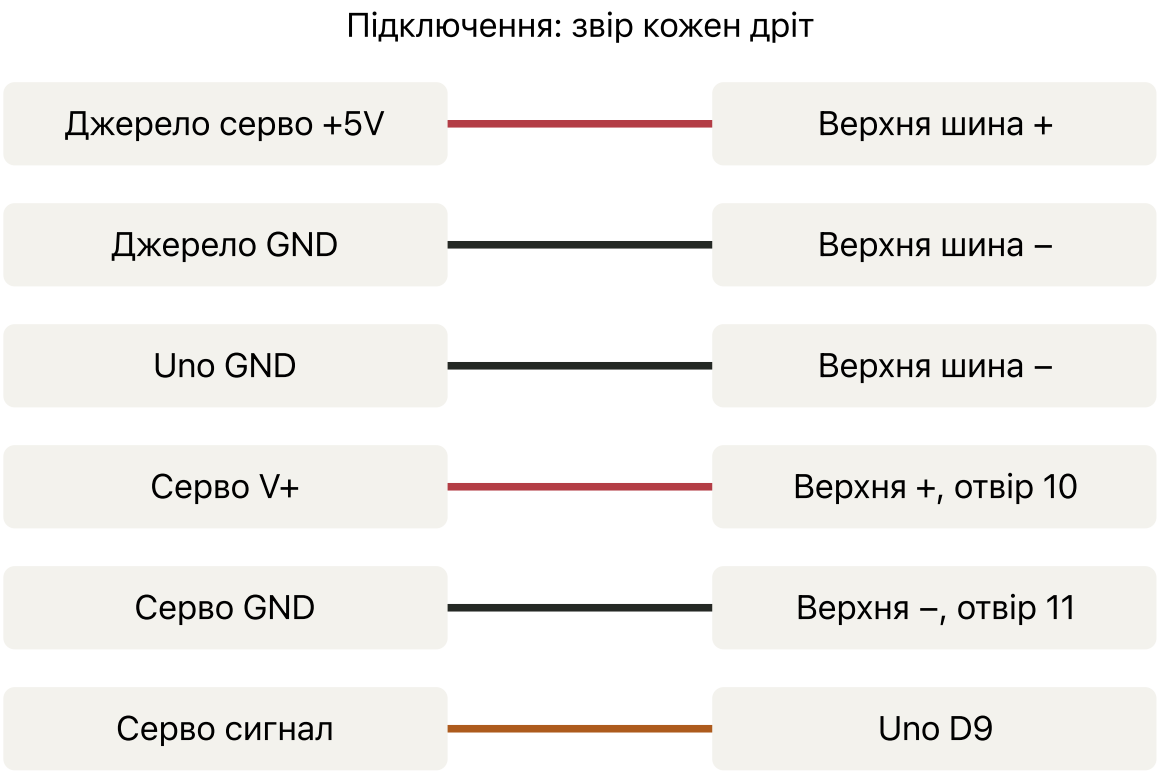
Очікуваний результат

Вал проходить проміжні положення від 60 до 120 і назад.

Що можна змінити

STEP_DEG = 1; STEP_MS = 30. Спочатку залиш ці значення.

Підключення справжнього комплекту



Два плюси розділені, GND спільний

На макетній платі верхня + — тільки джерело серво, нижня + — тільки Uno. Об’єднуємо мінусові шини. Перевір розриви шин; за потреби з’єднай половини тієї самої шини. У Wokwi верхня + живиться від Uno лише як спрощення. На справжній платі цей дріт заміни на +5 V окремого джерела; його мінус додай на верхню –.

Крок 2: код · 1

```
#include <Arduino.h>
#include <Servo.h>
const byte SERVO_PIN = 9;
const int MIN_ANGLE = 60;
const int MAX_ANGLE = 120;
const int STEP_DEG = 1;
const unsigned long STEP_MS = 30;
Servo arm;
int angle = MIN_ANGLE;
int direction = 1;

void setup() {
  arm.write(angle);
  arm.attach(SERVO_PIN);
}

void loop() {
  arm.write(angle);
  delay(STEP_MS);
  angle += direction * STEP_DEG;

  if (angle >= MAX_ANGLE) {
    angle = MAX_ANGLE;
    direction = -1;
  }

  if (angle <= MIN_ANGLE) {
    angle = MIN_ANGLE;
    direction = 1;
  }

}
```

Зміни паузу

Як це працює

Пауза повторюється після кожної команди

Для переходу від 60 до 120 із кроком 1 потрібні 60 змін кута. За 30 мс між ними це приблизно 60 × 30 = 1800 мс, тобто 1,8 с. За 60 мс — приблизно 3,6 с.

Що саме вимірюємо

Це час зміни команд між крайніми цілями, а не гарантований час руху механіки. Під час порівняння залиш STEP_DEG незмінним. Інакше буде незрозуміло, який параметр спричинив різницю.

Зроби

1. Порівняй STEP_MS = 20 і 60.
2. Запиши у зошиті, у якому досліді рух повільніший.
3. Поверни STEP_MS = 30.

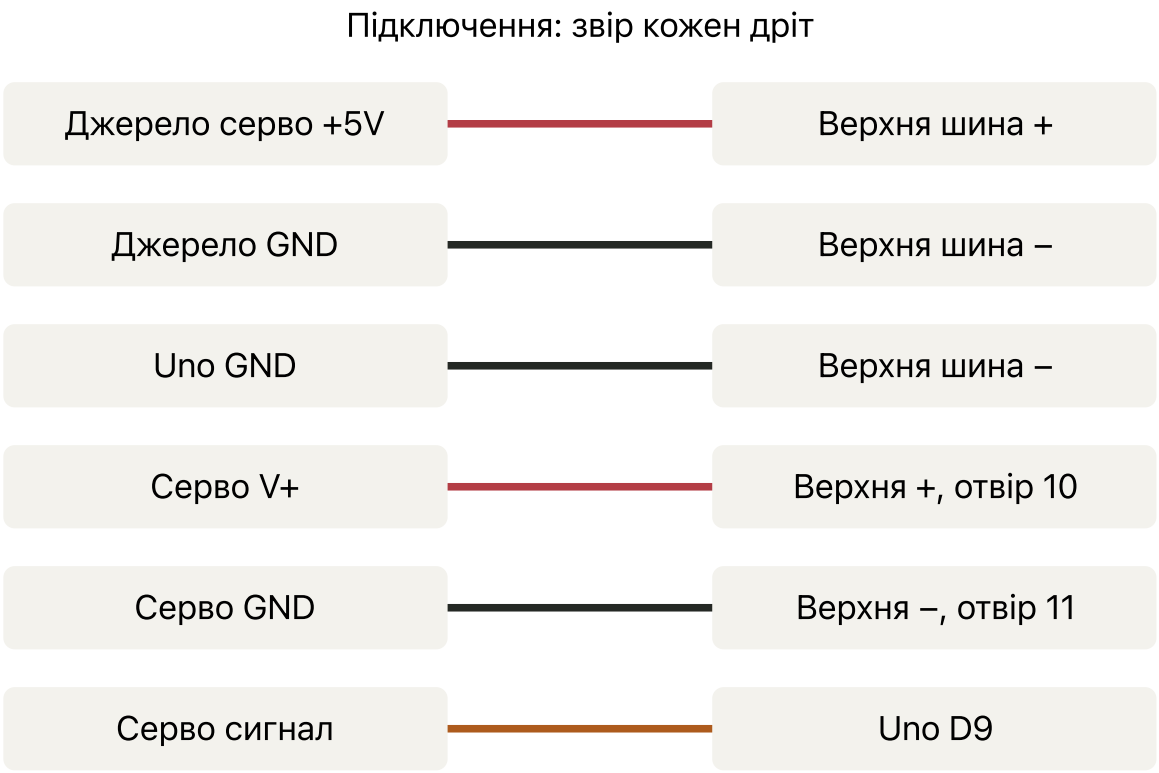
Очікуваний результат

Можеш пояснити вплив паузи, не змінюючи межі руху.

Що можна змінити

STEP_MS: 20–60 мс.

Підключення справжнього комплекту



Два плюси розділені, GND спільний

На макетній платі верхня + — тільки джерело серво, нижня + — тільки Uno. Об’єднуємо мінусові шини. Перевір розриви шин; за потреби з’єднай половини тієї самої шини. У Wokwi верхня + живиться від Uno лише як спрощення. На справжній платі цей дріт заміни на +5 V окремого джерела; його мінус додай на верхню –.

Крок 3: код · 1

```
#include <Arduino.h>
#include <Servo.h>
const byte SERVO_PIN = 9;
const int MIN_ANGLE = 60;
const int MAX_ANGLE = 120;
const int STEP_DEG = 1;
const unsigned long STEP_MS = 30;
Servo arm;
int angle = MIN_ANGLE;
int direction = 1;

void setup() {
  arm.write(angle);
  arm.attach(SERVO_PIN);
}

void loop() {
  arm.write(angle);
  delay(STEP_MS);
  angle += direction * STEP_DEG;

  if (angle >= MAX_ANGLE) {
    angle = MAX_ANGLE;
    direction = -1;
  }

  if (angle <= MIN_ANGLE) {
    angle = MIN_ANGLE;
    direction = 1;
  }

}
```

Зміни крок кута

Як це працює

Крок визначає кількість проміжних цілей

За STEP_DEG = 5 команди йдуть 60, 65, 70 і далі. До 120 буде 12 переходів замість 60. При тій самій паузі весь прохід команд скорочується, але проміжних положень стає менше.

Для чого перевіряти межі

Крок не завжди ділить діапазон без залишку. Перевірка >= MAX_ANGLE повертає команду точно на верхню межу, а <= MIN_ANGLE — на нижню. Так програма не продовжує збільшувати чи зменшувати кут без кінця.

Зроби

1. Порівняй STEP_DEG = 1 і 5 при однаковій паузі.
2. Опиши різницю плавності.
3. Збережи найплавніший варіант як Servo_sweep.

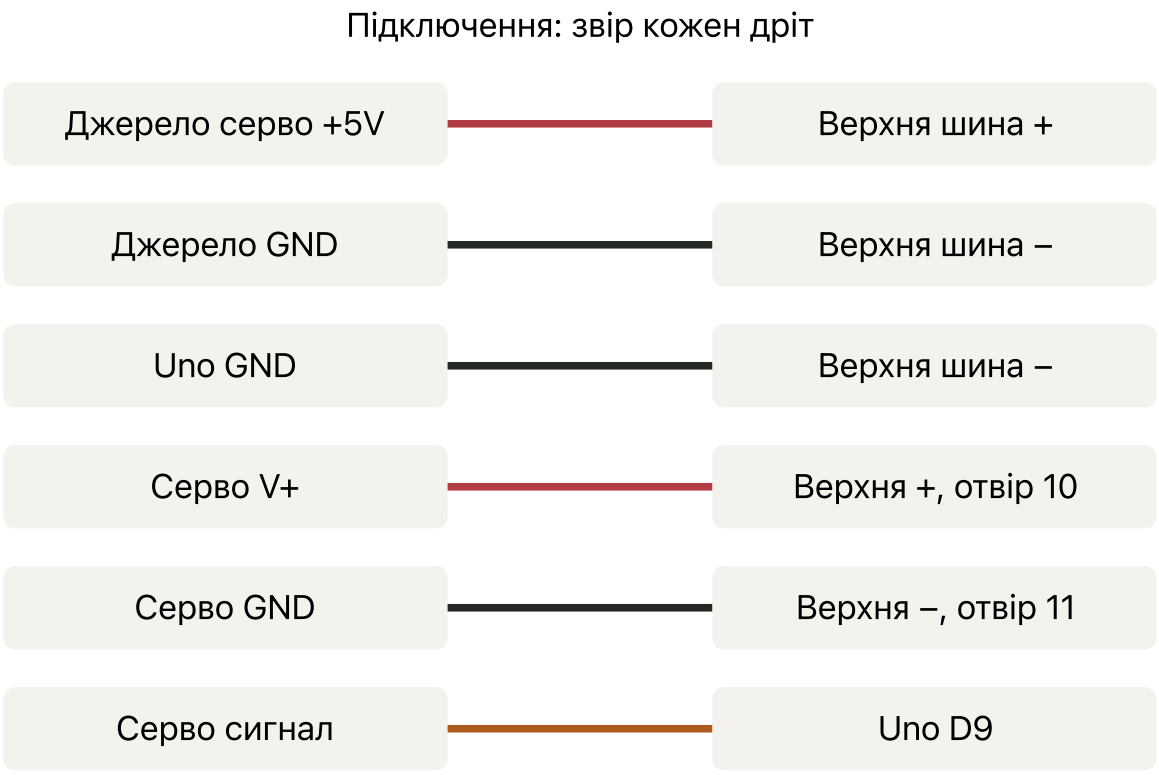
Очікуваний результат

У твоїй програмі є рух туди й назад і перевірені межі.

Що можна змінити

STEP_DEG: 1–5; MIN_ANGLE = 60, MAX_ANGLE = 120.

Підключення справжнього комплекту



Два плюси розділені, GND спільний

На макетній платі верхня + — тільки джерело серво, нижня + — тільки Uno. Об’єднуємо мінусові шини. Перевір розриви шин; за потреби з’єднай половини тієї самої шини. У Wokwi верхня + живиться від Uno лише як спрощення. На справжній платі цей дріт заміни на +5 V окремого джерела; його мінус додай на верхню –.

Крок 4: код · 1

```
#include <Arduino.h>
#include <Servo.h>
const byte SERVO_PIN = 9;
const int MIN_ANGLE = 60;
const int MAX_ANGLE = 120;
const int STEP_DEG = 1;
const unsigned long STEP_MS = 30;
Servo arm;
int angle = MIN_ANGLE;
int direction = 1;

void setup() {
  arm.write(angle);
  arm.attach(SERVO_PIN);
}

void loop() {
  arm.write(angle);
  delay(STEP_MS);
  angle += direction * STEP_DEG;

  if (angle >= MAX_ANGLE) {
    angle = MAX_ANGLE;
    direction = -1;
  }

  if (angle <= MIN_ANGLE) {
    angle = MIN_ANGLE;
    direction = 1;
  }

}
```

Перевір розуміння

Як це працює

Поясни результат через параметри

MIN_ANGLE і MAX_ANGLE визначають, між якими цілями рухаємося. STEP_DEG визначає величину одного переходу. STEP_MS визначає час до наступного переходу. direction визначає, додаємо ми кут чи віднімаємо.

Перевір себе

Потрібно зберегти крайні положення, але зробити прохід удвічі довшим. Який один параметр можна змінити? Спочатку сформулюй передбачення, потім перевір його на вільному серво.

Зроби

1. Запиши, як зробити рух повільнішим без зміни діапазону.
2. Поясни, навіщо direction змінює знак.
3. Покажи вчителю рух та свій збережений скетч.

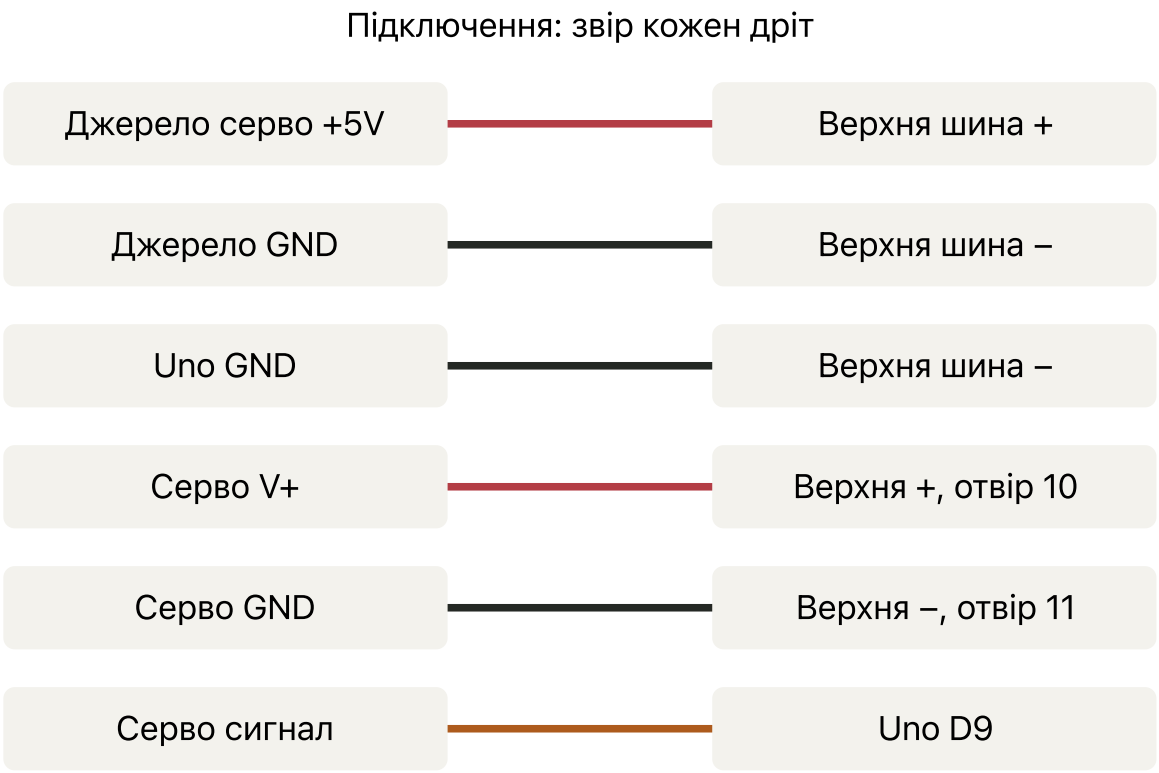
Очікуваний результат

Ти можеш передбачити наслідок зміни кроку і паузи.

Що можна змінити

Для останнього показу поверни STEP_DEG = 1 та STEP_MS = 30.

Підключення справжнього комплекту



Два плюси розділені, GND спільний

На макетній платі верхня + — тільки джерело серво, нижня + — тільки Uno. Об’єднуємо мінусові шини. Перевір розриви шин; за потреби з’єднай половини тієї самої шини. У Wokwi верхня + живиться від Uno лише як спрощення. На справжній платі цей дріт заміни на +5 V окремого джерела; його мінус додай на верхню –.

Крок 5: код · 1

```
#include <Arduino.h>
#include <Servo.h>
const byte SERVO_PIN = 9;
const int MIN_ANGLE = 60;
const int MAX_ANGLE = 120;
const int STEP_DEG = 1;
const unsigned long STEP_MS = 30;
Servo arm;
int angle = MIN_ANGLE;
int direction = 1;

void setup() {
  arm.write(angle);
  arm.attach(SERVO_PIN);
}

void loop() {
  arm.write(angle);
  delay(STEP_MS);
  angle += direction * STEP_DEG;

  if (angle >= MAX_ANGLE) {
    angle = MAX_ANGLE;
    direction = -1;
  }

  if (angle <= MIN_ANGLE) {
    angle = MIN_ANGLE;
    direction = 1;
  }

}
```

Джерела

Стаття й ілюстрації — DroneBot Workshop · Відео автора. Практичні завдання та програми адаптовано для цього курсу.

Далі: Серво слухається ручки →