Що робить потенціометр

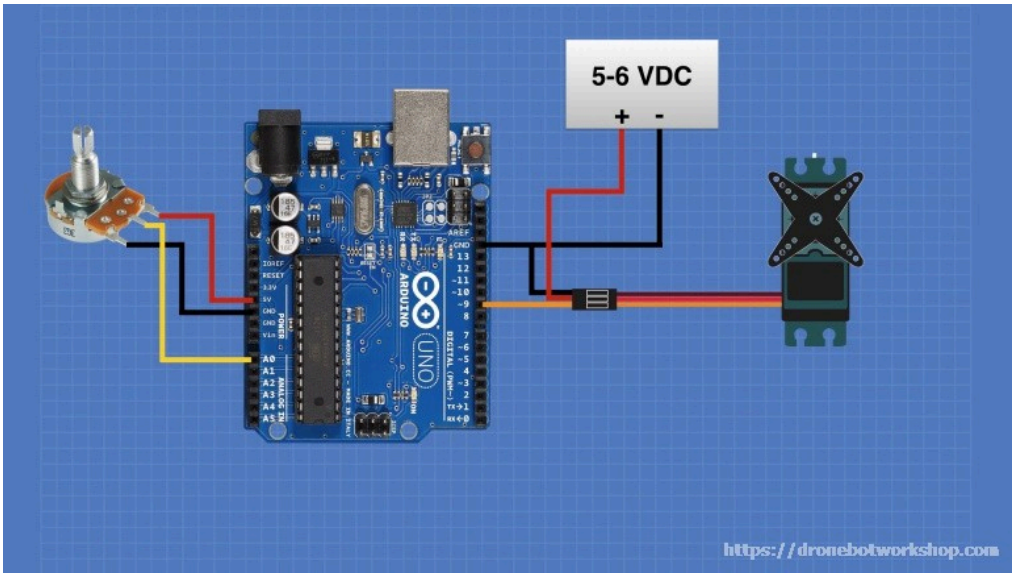
Як це працює

Що змінює ручка

Під ручкою є рухомий контакт, який ковзає вздовж резистивної доріжки. Коли її краї підключені до 5 V та `GND`, на рухомому контакті отримуємо проміжну напругу. Таке підключення називають подільником напруги.

Як напруга стає числом

`A0` на Uno вимірює напругу відносно `GND`. За звичайного налаштування analogRead повертає 0–1023: біля 0 V — близько 0, біля 5 V — близько 1023, посередині — близько 512. Це число ще не є кутом серво.



Потенціометр на `A0` та серво на `D9` · DroneBot Workshop

Зроби

1. Візьми лінійний потенціометр 10 кОм і три додаткові дроти.
2. Від'єднай обидва джерела живлення.
3. Знайди середній контакт; не орієнтуйся лише на напрямок ручки.

Очікуваний результат

Два краї задають діапазон напруги, середина передає показ.

Що можна змінити

Якщо потенціометра немає, виконай дослід у Wokwi.

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.

Прочитай положення ручки

Як це працює

Спочатку перевіряємо лише вхід

analogRead(POT_PIN) читає `A0`, а Serial.println(raw) виводить число. У цьому скетчі бібліотека Servo не використовується: ми спочатку переконуємося, що ручка дає зрозумілий показ.

Як знайти помилку

Якщо число не змінюється, звір середній контакт і `A0`. Якщо показ хаотичний, перевір `GND` та контакт дроту: від'єднаний аналоговий вхід може ловити завади. Невеликі коливання останніх одиниць можливі й у справному комплекті. Живлення ручки беремо з Uno, щоб її напруга відповідала діапазону входу.

Зроби

1. Підключи потенціометр за таблицею.
2. Завантаж код цього кроку.
3. Відкрий монітор порту на 115200 baud.
4. Повільно поверни ручку від одного краю до іншого.

Очікуваний результат

Показ змінюється приблизно від 0 до 1023. Серво в цьому кроці не отримує команд.

Що можна змінити

Не міняй POT_PIN без перепідключення дроту.

Підключення справжнього комплекту

Підключення: звір кожен дріт



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>
const byte POT_PIN = A0;

void setup() {
  Serial.begin(115200);
}

void loop() {
  int raw = analogRead(POT_PIN);
  Serial.println(raw);
  delay(100);
}
```

Перетвори показ на кут

Як це працює

Навіщо потрібен map

У ручки 1024 можливі покази, а наші команди серво обмежені 60–120. map(raw, 0, 1023, 60, 120) зберігає відносне положення: початок входу стає початком виходу, середина — серединою, кінець — кінцем.

Прочитай один рядок монітора

512 -> 90 означає: прочитали raw = 512, обчислили angle = 90, передали його в arm.write. За raw близько 256 команда буде близько 75. map працює з цілими числами, тому кілька сусідніх показів можуть давати той самий кут.

Ланцюжок керування

Рука повертає ручку → змінюється напруга `A0` → Arduino читає число → map обчислює ціль → Servo надсилає імпульси. Потенціометр не живить двигун і не повертає його безпосередньо.

Зроби

1. Завантаж повний код цього кроку.
2. Постав ручку в середину.
3. Звір рядок raw -> angle у моніторі.
4. Поверни ручку та простеж рух вала.

Очікуваний результат

Ручка керує кутом у межах 60–120.

Що можна змінити

MIN_ANGLE і MAX_ANGLE задають межі; поки залиш 60 та 120.

Підключення справжнього комплекту

Підключення: звір кожен дріт



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 byte POT_PIN = A0;
const int MIN_ANGLE = 60;
const int MAX_ANGLE = 120;
Servo arm;

void setup() {
  Serial.begin(115200);
  arm.write(90);
  arm.attach(SERVO_PIN);
}

void loop() {
  int raw = analogRead(POT_PIN);
  int angle = map(raw, 0, 1023, MIN_ANGLE, MAX_ANGLE);
  arm.write(angle);
  Serial.print(raw);
  Serial.print(" -> ");
  Serial.println(angle);
  delay(30);
}
```

Звузь діапазон

Як це працює

Змінюємо лише вихідну шкалу

Після заміни меж на 75 і 105 показ ручки залишається 0–1023. Початок тепер дає 75°, середина приблизно 90°, а кінець 105°. Повний хід ручки відповідає 30° команд замість 60°.

Для чого це потрібно

Так можна обмежити рух покажчика або механізму. Програмні межі обирають після перевірки реального ходу. Вони не визначають автоматично, де механічний упор: програма про нього не знає.

Зроби

1. Задай MIN_ANGLE = 75 і MAX_ANGLE = 105.
2. Завантаж код.
3. Порівняй повний хід ручки зі старим дослідом.

Очікуваний результат

Ручка повертається так само, а серво проходить менший кут.

Що можна змінити

Вибирай межі всередині 60–120; MIN_ANGLE має бути меншим за MAX_ANGLE.

Підключення справжнього комплекту

Підключення: звір кожен дріт



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 byte POT_PIN = A0;
const int MIN_ANGLE = 60;
const int MAX_ANGLE = 120;
Servo arm;

void setup() {
  Serial.begin(115200);
  arm.write(90);
  arm.attach(SERVO_PIN);
}

void loop() {
  int raw = analogRead(POT_PIN);
  int angle = map(raw, 0, 1023, MIN_ANGLE, MAX_ANGLE);
  arm.write(angle);
  Serial.print(raw);
  Serial.print(" -> ");
  Serial.println(angle);
  delay(30);
}
```

Зроби власний покажчик

Як це працює

Чому покажчик має бути легким

Довгий або важкий важіль сильніше навантажує вал. Обертальний момент залежить від сили й відстані до осі: та сама сила на вдвічі довшому плечі створює вдвічі більший момент. Тому починаємо з легкої паперової стрілки.

Що означає паспортний момент

Момент часто вказують у кгс·см. Це не маса, яку серво може безпечно підіймати за будь-якої довжини важеля. Граничний момент біля зупиненого вала також не є режимом для тривалої роботи. Не перевіряємо його, затискаючи вал пальцями.

Зроби

1. Попроси вчителя закріпити легкий покажчик після перевірки центру.
2. Познач на папері три положення.
3. Збережи скетч як Servo_knob.
4. Запиши, що станеться при від'єднаному `A0`.

Очікуваний результат

Маєш ручне керування покажчиком; плаваючий вхід може давати нестійкі команди.

Що можна змінити

Спочатку перевір електроніку без покажчика. Не використовуй важкі деталі.

Підключення справжнього комплекту

Підключення: звір кожен дріт



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 byte POT_PIN = A0;
const int MIN_ANGLE = 60;
const int MAX_ANGLE = 120;
Servo arm;

void setup() {
  Serial.begin(115200);
  arm.write(90);
  arm.attach(SERVO_PIN);
}

void loop() {
  int raw = analogRead(POT_PIN);
  int angle = map(raw, 0, 1023, MIN_ANGLE, MAX_ANGLE);
  arm.write(angle);
  Serial.print(raw);
  Serial.print(" -> ");
  Serial.println(angle);
  delay(30);
}
```

Джерела

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

Далі: Автоматична кришка →