УРОК 09-05 · КРОК 1 / 5

Відстань має напрямок

Як це працює

Однієї відстані недостатньо для карти

Показ 30 см повідомляє, як далеко відбиття, але не пояснює, у який бік дивиться датчик. Якщо повертати датчик і записувати команду кута поруч із відстанню, отримаємо кілька напрямків огляду.

Що ми справді знаємо

Arduino знає надісланий кут, але не отримує виміряний кут із нашого серво. Тому це приблизна карта. HC-SR04 використовує звук, а не радіохвилі; назва «радар» для такої моделі умовна.

Зроби

1. Повтори окреме живлення серво та спільний `GND`.
2. Підключи датчик за таблицею.
3. Залиш датчик нерухомим, поки перевіряєш електроніку.

Очікуваний результат

Серво та датчик підключені; механічне кріплення встановлює вчитель після перевірки.

Що можна змінити

Без готового кріплення проходь програмні кроки в симуляторі.

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 –.

Поверни, почекай, виміряй

Як це працює

Три дії мають свій порядок

Спочатку write задає кут. Потім SETTLE_MS залишає час на поворот і заспокоєння кріплення. Після цього вимірюємо відлуння. Якщо вимірювати відразу, датчик може ще рухатися, і підписаний кут не відповідатиме напрямку вимірювання.

Прочитай пару даних

Рядок 80,35 означає: команда серво 80°, відстань 35 см. Рядок 80,NA означає, що надійного показу немає. NA не є нульовою відстанню — таку точку пропускаємо.

Зроби

1. Завантаж код.
2. Відкрий монітор порту на 115200 baud.
3. Прочитай рядки angle,distance_cm.
4. Знайди повторення кута 90 під час руху назад.

Очікуваний результат

Кути йдуть 60, 70, 80, 90, 100, 110, 120 та у зворотному порядку.

Що можна змінити

STEP_DEG = 10; SETTLE_MS = 300.

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 byte TRIG_PIN = 7;
const byte ECHO_PIN = 6;
const int MIN_ANGLE = 60;
const int MAX_ANGLE = 120;
const int STEP_DEG = 10;
const unsigned long SETTLE_MS = 300;
Servo arm;
int angle = MIN_ANGLE;
int direction = 1;

float readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW); // Готуємо короткий імпульс
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); // Запускаємо вимірювання
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000UL);
  float cm = echoUs / 58.0; // Шлях туди й назад уже врахований

  if (echoUs == 0 || cm < 2 || cm > 400) {
    return -1; // Немає надійного вимірювання
  }

  return cm;
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT); // TRIG — команда датчику
  pinMode(ECHO_PIN, INPUT); // ECHO — відповідь датчика
  digitalWrite(TRIG_PIN, LOW);
```

Крок 2: код · 2

```
  Serial.begin(115200); // Така сама швидкість у моніторі
  arm.write(angle);
  arm.attach(SERVO_PIN);
  Serial.println("angle,distance_cm");
}

void loop() {
  arm.write(angle);
  delay(SETTLE_MS);
  float cm = readDistanceCm();
  Serial.print(angle);
  Serial.print(",");

  if (cm < 0) {
    Serial.println("NA");
  } else {
    Serial.println(cm);
  }

  angle += direction * STEP_DEG;

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

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

}
```

Перевір симуляцію чесно

Як це працює

Що моделює Wokwi

Симулятор виконує програму й дозволяє задати відстань для HC-SR04. Але два компоненти не утворюють у ньому справжній поворотний вузол: зміна кута серво не змінює напрямок датчика.

Який результат очікувати

При Distance = 50 усі кути можуть показувати 50 см — це очікувано. Так перевіряємо порядок кутів, формат рядків і обробку помилки. Для карти реальних перешкод потрібні фізичне кріплення й дослід у просторі.

Зроби

1. Постав у Wokwi Distance = 50.
2. Переконайся, що при різних кутах відстань лишається 50.
3. Зміни Distance на 20.
4. Постав 1 см і перевір NA.

Очікуваний результат

Ти перевірив формат і порядок даних, але ще не карту справжньої кімнати.

Що можна змінити

Для фізичного досліду потрібні кріплення датчика, вільні дроти й перевірений хід без зачіпання.

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 byte TRIG_PIN = 7;
const byte ECHO_PIN = 6;
const int MIN_ANGLE = 60;
const int MAX_ANGLE = 120;
const int STEP_DEG = 10;
const unsigned long SETTLE_MS = 300;
Servo arm;
int angle = MIN_ANGLE;
int direction = 1;

float readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW); // Готуємо короткий імпульс
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); // Запускаємо вимірювання
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000UL);
  float cm = echoUs / 58.0; // Шлях туди й назад уже врахований

  if (echoUs == 0 || cm < 2 || cm > 400) {
    return -1; // Немає надійного вимірювання
  }

  return cm;
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT); // TRIG — команда датчику
  pinMode(ECHO_PIN, INPUT); // ECHO — відповідь датчика
  digitalWrite(TRIG_PIN, LOW);
```

Крок 3: код · 2

```
  Serial.begin(115200); // Така сама швидкість у моніторі
  arm.write(angle);
  arm.attach(SERVO_PIN);
  Serial.println("angle,distance_cm");
}

void loop() {
  arm.write(angle);
  delay(SETTLE_MS);
  float cm = readDistanceCm();
  Serial.print(angle);
  Serial.print(",");

  if (cm < 0) {
    Serial.println("NA");
  } else {
    Serial.println(cm);
  }

  angle += direction * STEP_DEG;

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

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

}
```

Намалюй карту перешкод

Як це працює

Переносимо вимірювання на папір

Усі промені починаються в одній точці — умовному місці датчика. На кожному промені відкладаємо його відстань. За масштабу 1:10 показ 40 см стає 4 см на папері. Не з'єднуй точки так, ніби між ними вже точно відома форма предмета.

Чому карта приблизна

Датчик охоплює ділянку простору, а не одну тонку лінію. Похила поверхня може відбити звук убік, м'яка — послабити його. Різні покази поруч не завжди означають різкий край предмета: повтори вимірювання й порівняй.

Зроби

1. Після перевірки кріплення вчителем постав три картонні перешкоди на різних напрямках.
2. Запиши один прохід 60–120 у таблицю: кут | відстань.
3. Намалюй сім променів на папері.
4. Познач виміряні відстані в масштабі 1 см на папері = 10 см у досліді.

Очікуваний результат

Маєш просту карту; NA залишай пропуском, не малюй перешкоду на нульовій відстані.

Що можна змінити

Якщо працюєш тільки у Wokwi, підпиши таблицю «симуляція» та задай три різні відстані вручну.

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 byte TRIG_PIN = 7;
const byte ECHO_PIN = 6;
const int MIN_ANGLE = 60;
const int MAX_ANGLE = 120;
const int STEP_DEG = 10;
const unsigned long SETTLE_MS = 300;
Servo arm;
int angle = MIN_ANGLE;
int direction = 1;

float readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW); // Готуємо короткий імпульс
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); // Запускаємо вимірювання
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000UL);
  float cm = echoUs / 58.0; // Шлях туди й назад уже врахований

  if (echoUs == 0 || cm < 2 || cm > 400) {
    return -1; // Немає надійного вимірювання
  }

  return cm;
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT); // TRIG — команда датчику
  pinMode(ECHO_PIN, INPUT); // ECHO — відповідь датчика
  digitalWrite(TRIG_PIN, LOW);
```

Крок 4: код · 2

```
  Serial.begin(115200); // Така сама швидкість у моніторі
  arm.write(angle);
  arm.attach(SERVO_PIN);
  Serial.println("angle,distance_cm");
}

void loop() {
  arm.write(angle);
  delay(SETTLE_MS);
  float cm = readDistanceCm();
  Serial.print(angle);
  Serial.print(",");

  if (cm < 0) {
    Serial.println("NA");
  } else {
    Serial.println(cm);
  }

  angle += direction * STEP_DEG;

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

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

}
```

Порівняй деталізацію

Як це працює

Кількість положень і час

Від 60 до 120 із кроком 10 є 7 положень, бо враховуємо обидва краї. Із кроком 5 — 13. Кожне потребує окремої паузи та вимірювання, тому більше точок означає довший прохід.

Шукаємо корисний компроміс

Менший крок може показати деталі, але не робить промінь датчика вужчим. Коротша пауза прискорює сканування, проте механіка може не встигнути зупинитися. Змінюй один параметр за раз і порівнюй повторюваність показів.

Зроби

1. Зміни STEP_DEG з 10 на 5.
2. Порахуй кількість точок від 60 до 120 включно.
3. Порівняй тривалість проходу.
4. Збережи скетч як Servo_scanner і покажи таблицю вчителю.

Очікуваний результат

Можеш пояснити обмін між кількістю точок і часом сканування.

Що можна змінити

STEP_DEG: 5 або 10; SETTLE_MS: 300–600. Кути лишаються 60–120.

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 5: код · 1

```
#include <Arduino.h>
#include <Servo.h>
const byte SERVO_PIN = 9;
const byte TRIG_PIN = 7;
const byte ECHO_PIN = 6;
const int MIN_ANGLE = 60;
const int MAX_ANGLE = 120;
const int STEP_DEG = 10;
const unsigned long SETTLE_MS = 300;
Servo arm;
int angle = MIN_ANGLE;
int direction = 1;

float readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW); // Готуємо короткий імпульс
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); // Запускаємо вимірювання
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  unsigned long echoUs = pulseIn(ECHO_PIN, HIGH, 25000UL);
  float cm = echoUs / 58.0; // Шлях туди й назад уже врахований

  if (echoUs == 0 || cm < 2 || cm > 400) {
    return -1; // Немає надійного вимірювання
  }

  return cm;
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT); // TRIG — команда датчику
  pinMode(ECHO_PIN, INPUT); // ECHO — відповідь датчика
  digitalWrite(TRIG_PIN, LOW);
```

Крок 5: код · 2

```
  Serial.begin(115200); // Така сама швидкість у моніторі
  arm.write(angle);
  arm.attach(SERVO_PIN);
  Serial.println("angle,distance_cm");
}

void loop() {
  arm.write(angle);
  delay(SETTLE_MS);
  float cm = readDistanceCm();
  Serial.print(angle);
  Serial.print(",");

  if (cm < 0) {
    Serial.println("NA");
  } else {
    Serial.println(cm);
  }

  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.