

Автоматичний шлагбаум

Іграшкова машинка наближається — картонна стріла підіймається. Після звільнення зони вона опускається.

- Arduino Uno, макетна плата, HC-SR04, позиційний SG90 або MG90S 180° з важелем і гвинтом.
- Окреме стабілізоване джерело 5 В для серво, дроти, USB для Uno.
- Невелика коробка, тонкий картон або легка соломинка довжиною 10–15 см, скотч.
- Іграшкова машинка, лінійка, плаский картон для відлуння.

Сервопривід безперервного обертання 360° тут не підходить: потрібно задавати положення стріли.

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

Стріла з картону

1. Закріпи корпус серво на коробці так, щоб вал був збоку й стріла могла рухатися вгору.
2. Виріж вузьку легку смужку 10–15 см або візьми соломинку. Розфарбуй червоно-білими смугами.
3. Прикріпи її скотчем до пластикового важеля з комплекту серво.
4. Поки не встановлюй важіль на вал: спочатку задаємо початковий кут.

Стріла має рухатися вільно, не зачіпаючи коробку. Не використовуй важку дерев'яну планку й не прокручуй вал силою.



Схема складання · не масштаб

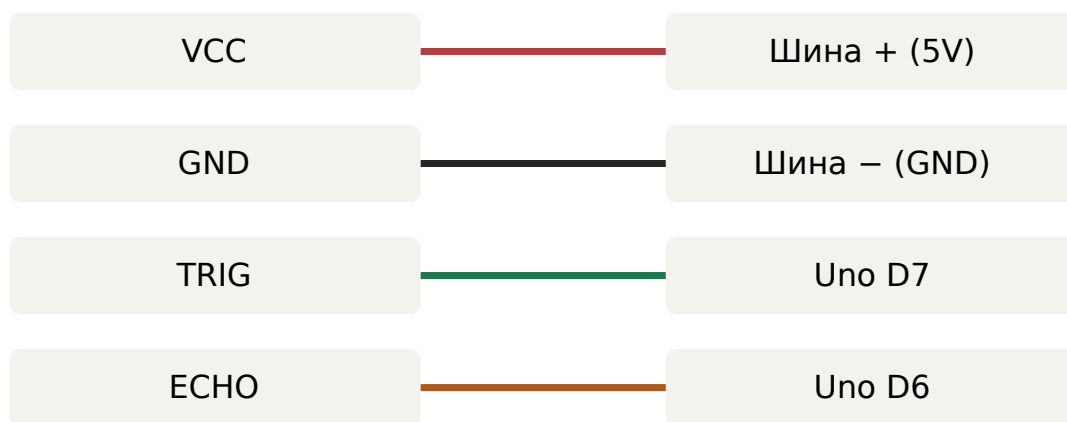
Датчик біля в'їзду

1. Від'єднай живлення. Прочитай написи біля чотирьох контактів HC-SR04.
2. З'єднай **5V** Uno з верхньою шиною **+**, а **GND** Uno — з верхньою шиною **–** макетної плати.
3. Проведи від **VCC** датчика червоний дріт до шини **+**, а від **GND** — чорний до шини **–**.
4. З'єднай **TRIG** з **D7**. Цим виходом Arduino запускає вимірювання.
5. З'єднай **ECHO** з **D6**. На цьому вході Arduino вимірює тривалість відповіді.

Шини **+** і **–** мають з'єднані між собою отвори вздовж плати. Через них підводимо живлення до датчика. На фізичній платі перевір, чи немає розриву посередині шини; за потреби з'єднай її половини перемичкою.

Звіряй саме написи: напрямок роз'єму може відрізнитися від малюнка. Для модуля з ніжками й гнізд Uno зручні дроти «мама-тато».

HC-SR04 → Arduino Uno



Електричні з'єднання · не масштаб плати

Закріпи датчик біля під'їзду так, щоб він бачив передню частину машинки, а не стрілу. Для досліду постав за зоною проїзду плаский фон на 40–60 см: без машинки датчик має стабільно бачити його.

Три дроти сервопривода

1. Сигнал PWM → D9 .
2. V+ → верхня шина + (5V) лише у цій симуляції.
3. GND → верхня шина —, спільна з датчиком.

Бібліотека Servo формує керувальні імпульси. Додай її у Library Manager Wokwi або перенеси `libraries.txt` із ZIP.

У Wokwi замість картонної стріли дивимося на положення важеля серво. На справжній платі підключи живлення за наступним кроком.



Електричні з'єднання · не масштаб плати

Серво живиться окремо

1. Знеструм обидві частини схеми.
2. Плюс окремого стабілізованого джерела 5 В → живлення серво.
3. Мінус джерела → **GND** серво та **GND** Uno.
4. Сигнальний дріт серво → **D9**. Uno й HC-SR04 отримують живлення через USB.

Зазвичай у SG90 червоний — живлення, коричневий — **GND**, помаранчевий — сигнал; звір маркування вашого виробу.

Спільний **GND** потрібен для правильного рівня керувальних імпульсів. Плюс зовнішнього джерела з **5V** Uno не з'єднуй. Джерело з достатнім струмом для конкретного серво підбирає вчитель.

Серво · фізична модель



Електричні з'єднання · не масштаб плати

Встановлюємо стрілу на вал

1. Залиш важіль зі стрілою знятим. Завантаж повний код і ввімкни схему, тримаючи предмети далі 25 см.
2. Серво перейде в `CLOSED_ANGLE = 10`.
3. Вимкни живлення. Установи важіль на вал так, щоб стріла була приблизно горизонтальною, закріпи комплектним гвинтом.
4. Увімкни, піднеси картон до датчика на 10 см. Стріла має піднятися.

Якщо рух іде вниз, зміни орієнтацію кріплення серво або разом з учителем підбери інше відкрите положення. Положення стріли залежить від кута вала й того, як встановлено важіль.

Починай перевірку без машинки під стрілою. Якщо механізм упирається або гуде, вимкни живлення й виправ кріплення чи кути.

Два пороги й таймер

```
const int CLOSED_ANGLE = 10; // Закрите положення без тиску на крайній упор.  
const int OPEN_ANGLE = 90; // Відкрите положення; звірити напрямок на своєму серво.  
const float OPEN_CM = 15; // Відкриваємо на відстані до 15 см включно.  
const float CLEAR_CM = 25; // Вважаємо зону вільною лише далі за 25 см.  
const unsigned long CLEAR_MS = 3000; // Потрібні 3 секунди вільної зони.  
Servo barrier; // Об'єкт керування стрілою.  
bool isOpen = false; // Початково стріла закрита.  
bool clearing = false; // Чи відраховуємо час вільної зони.  
unsigned long clearAt = 0; // Початок безперервного вільного інтервалу.
```

- До 15 см включно — відкрити й скасувати відлік.
- Понад 15 і до 25 см — залишити положення, скасувати відлік.
- Понад 25 см — якщо відкрито, почати відлік. Закрити після 3 секунд безперервно вільної зони.
- Немає відлуння — залишити положення й скасувати відлік.

Два різні пороги не дають стрілі смикатися біля однієї межі. `clearAt` зберігає початок інтервалу, `clearing` — чи він триває.

Один датчик спереду не бачить усього простору під стрілою. Це навчальна настільна модель; перед закриттям прибираємо машинку з-під стріли.

Перевір відкривання та затримку

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.

1. Перевір бібліотеку Servo. Почни з відстані 50 см: серво в закритому положенні.
2. Задай 10 см: серво переходить у відкрите.
3. Задай 20 см й зачекай 5 секунд: залишається відкритим.
4. Задай 50 см: через приблизно 3 секунди закривається.
5. Відкрий знову. Установи 50 см на 2 секунди, потім 10 см: відлік скасовується.

На фізичній моделі заberi машинку повністю й перевір стабільний показ фону 40–60 см. Якщо бачиш -1, стріла не повинна сама опускатися.

Окремо перевір датчик і механіку

Стріла не опускається

Перевір монітор порту 115200. Потрібні понад 25 см протягом 3 секунд. -1 або 20 см скидають таймер. Постав плоский фон на 40–60 см.

Серво смикається або Uno перезавантажується

Вимкни живлення. Перевір окреме джерело серво, спільний **GND** та відсутність механічного упору. Не намагайся виправити нестачу струму зміною затримок.

На старті стріла рухається

Після запуску програма задає закритий кут 10° . Тому перше налаштування виконуємо зі знятим важелем, а кожен запуск — із вільним місцем для руху.

Затримка закриття — 5 секунд

Збільш час безперервно вільної зони перед закриттям із 3 до 5 секунд.

1. Зміни лише `CLEAR_MS`.
2. Відкрий шлагбаум, потім прибери предмет за 25 см.
3. Через 4 секунди поверни предмет на 10 см: стріла має залишитися відкритою.
4. Прибери предмет знову й відрахуй повні 5 секунд.

Очікуваний результат

`CLEAR_MS = 5000`. Нове наближення скасовує попередній відлік. Після повторного звільнення зони відраховуються нові 5 секунд.

Автоматичний шлагбаум

HC-SR04 керує положенням SG90; закриття після 3 секунд вільної зони.

```
#include <Arduino.h> // Основні команди плати.
#include <Servo.h> // Керування позиційним сервоприводом.

const byte TRIG_PIN = 7; // Команда датчику почати вимірювання.
const byte ECHO_PIN = 6; // Вхід тривалості відлуння.
const byte SERVO_PIN = 9; // Сигнальний дріт серво.
const int CLOSED_ANGLE = 10; // Закрите положення без тиску на крайній упор.
const int OPEN_ANGLE = 90; // Відкрите положення; звірити напрямок на своєму серво.
const float OPEN_CM = 15; // Відкриваємо на відстані до 15 см включно.
const float CLEAR_CM = 25; // Вважаємо зону вільною лише далі за 25 см.
const unsigned long CLEAR_MS = 3000; // Потрібні 3 секунди вільної зони.
Servo barrier; // Об'єкт керування стрілою.
bool isOpen = false; // Початково стріла закрита.
bool clearing = false; // Чи відраховуємо час вільної зони.
unsigned long clearAt = 0; // Початок безперервного вільного інтервалу.

float readDistanceCm() { // Повертаємо сантиметри або -1 при помилці.
    digitalWrite(TRIG_PIN, LOW); // Початковий низький рівень.
    delayMicroseconds(2); // Даємо сигналу стабілізуватися.
    digitalWrite(TRIG_PIN, HIGH); // Починаємо запуск вимірювання.
    delayMicroseconds(10); // Тримаємо імпульс 10 мікросекунд.
    digitalWrite(TRIG_PIN, LOW); // Завершуємо імпульс.
    unsigned long duration = pulseIn(ECHO_PIN, HIGH, 25000UL); // Чекаємо не довше 25 мс.
    float cm = duration / 58.0; // Час шляху звуку туди й назад переводимо в см.

    if (duration == 0 || cm < 2 || cm > 400) { // Немає надійного вимірювання.
        return -1; // Окреме значення помилки, а не нуль сантиметрів.
    }

    return cm; // Віддаємо виміряну відстань.
}

void setup() { // Підготовка після ввімкнення.
    pinMode(TRIG_PIN, OUTPUT); // TRIG надсилає імпульс.
    pinMode(ECHO_PIN, INPUT); // ECHO приймає імпульс.
    digitalWrite(TRIG_PIN, LOW); // До першого вимірювання тримаємо LOW.
    Serial.begin(115200); // Монітор порту для перевірки.
    barrier.write(CLOSED_ANGLE); // Заздалегідь задаємо початковий кут.
    barrier.attach(SERVO_PIN); // Починаємо надсилати імпульси на D9.
}

void loop() { // Перевіряємо наближення та звільнення зони.
    float cm = readDistanceCm(); // Отримуємо відстань або -1.
    unsigned long now = millis(); // Час після вимірювання.
    Serial.println(cm); // Показ для налаштування датчика.
```

```
if (cm < 0) { // Невдале вимірювання не доводить, що проїзд вільний.  
    clearing = false; // Скидаємо відлік закривання; стріла залишається на місці.  
} else if (cm <= OPEN_CM) { // Предмет близько.  
    barrier.write(OPEN_ANGLE); // Підіймаємо стрілу.  
    isOpen = true; // Зберігаємо відкрите положення.  
    clearing = false; // Предмет знову поруч – відлік скасовано.  
} else if (isOpen && cm > CLEAR_CM) { // Відкритий шлагбаум і вільна зона.  
  
    if (!clearing) { // Перший вимір у новому вільному інтервалі.  
        clearAt = now; // Запам'ятовуємо його початок.  
        clearing = true; // Вмикаємо відлік.  
    }  
  
    if (now - clearAt >= CLEAR_MS) { // Зона була вільна щонайменше 3 секунди.  
        barrier.write(CLOSED_ANGLE); // Опускаємо легку стрілу.  
        isOpen = false; // Зберігаємо закритий стан.  
        clearing = false; // Завершуємо відлік.  
    }  
  
} else { // Предмет у проміжній зоні або шлагбаум уже закритий.  
    clearing = false; // Не накопичуємо час для закривання.  
}  
  
delay(100); // Пауза перед новим вимірюванням.  
}
```