Світлодіод повідомляє про наближення

Наближай предмет до датчика: світлодіод спочатку вимкнений, потім блимає повільно, швидко й нарешті світиться постійно.

- Arduino Uno та USB-кабель.
- Макетна плата, датчик HC-SR04, дроти.
- Червоний світлодіод і резистор 220 Ом.
- Лінійка та плаский картон.

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

Звук і відлуння

Датчик посилає короткий ультразвуковий сигнал і чекає відбиття від предмета. Людина цей звук зазвичай не чує.

1. Arduino тримає `TRIG` у HIGH протягом 10 мікросекунд.
2. `pulseIn` вимірює, скільки мікросекунд `ECHO` перебуває у HIGH.
3. Ділимо тривалість на 58 і отримуємо приблизну відстань у сантиметрах.

Звук проходить шлях до предмета й назад. Коефіцієнт 58 уже враховує обидва напрямки. Наприклад, 580 мкс відповідає приблизно 10 см.

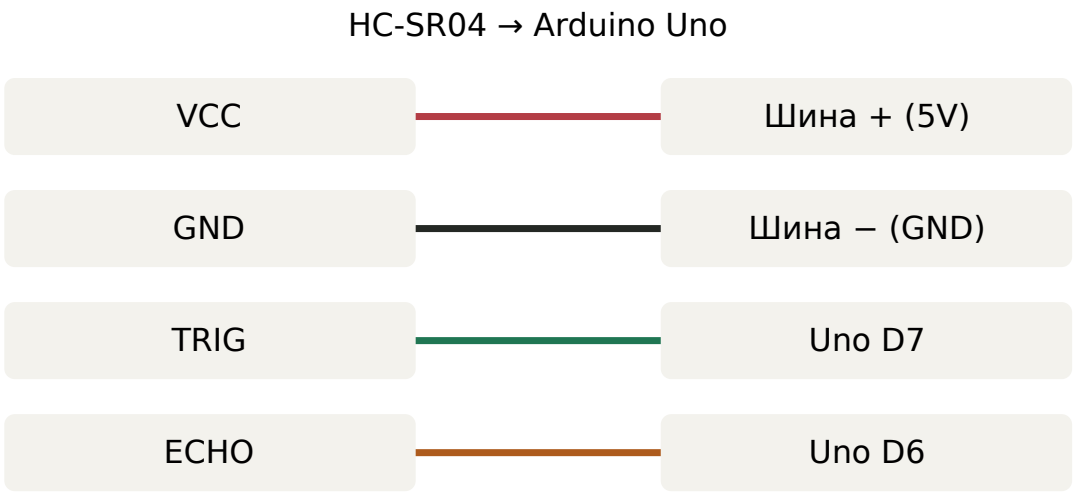
Чекаємо відповідь максимум 25 мс. Якщо її немає або результат поза 2–400 см, функція повертає `-1`. Це позначка невдалого вимірювання.

Датчик на макетній платі

1. Від’єднай живлення. Встав HC-SR04 у ряд `g`: `VCC → 12g`, `TRIG → 13g`, `ECHO → 14g`, `GND → 15g`. Ніжки мають бути в різних колонках. У межах кожної колонки отвори f–j з’єднані всередині плати.
2. З’єднай `5V` Uno з верхньою шиною `+`, а `GND` Uno — з верхньою шиною `−` макетної плати.
3. Проведи від `12h` (`VCC`) червоний дріт до шини `+`, а від `15h` (`GND`) — чорний до шини `−`.
4. З’єднай `13j` (TRIG) з `D7`. Цим виходом Arduino запускає вимірювання.
5. З’єднай `14i` (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 відрізнятися від малюнка. Якщо ніжки твого модуля не дозволяють вставити його в макетну плату, під’єднай ті самі контакти дротами «мама–тато».



Електричні з’єднання · не масштаб плати

Коло світлодіода на D8

1. Встав LED: анод A → `24a`, катод C → `23a`.
2. Встав резистор 220 Ом у `24d` та `30d`.
3. З’єднай `D8` з `30e`.
4. З’єднай `23b` з верхньою шиною −.

Отвори a–e одного номера з’єднані всередині макетної плати.

`D8` → `30e` → `30d` → резистор → `24d` → `24a` → LED → `23a` → `23b` → шина − → `GND`

Струм проходить через резистор, який обмежує його величину. Довга ніжка нового LED — анод; плаский край корпусу позначає катод.

Виміряти, показати, перевірити

```
void loop() {
  float distanceCm = readDistanceCm(); // 1. Вимірюємо відстань.
  Serial.println(distanceCm); // 2. Показуємо число в моніторі порту.
  showDistance(distanceCm); // 3. Вибираємо світловий сигнал.
}
```

1. Вимірюємо відстань і зберігаємо її у `distanceCm`.
2. Показуємо число в моніторі порту.
3. Функція `showDistance()` вибирає світловий сигнал і виконує його. Потім цикл повторюється.

Функція — група команд із назвою. Далі розглянемо дії всередині двох функцій.

Як отримати сантиметри

```
float readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW); // Готуємо вихід до нового імпульсу.
  delayMicroseconds(2); // Чекаємо 2 мікросекунди.
  digitalWrite(TRIG_PIN, HIGH); // Починаємо імпульс запуску.
  delayMicroseconds(10); // Тримаємо його 10 мікросекунд.
  digitalWrite(TRIG_PIN, LOW); // Завершуємо імпульс.

  unsigned long echoTime = pulseIn(ECHO_PIN, HIGH, 25000); // Чекаємо
відповідь до 25 мс.
  float distanceCm = echoTime / 58.0; // Перетворюємо тривалість відповіді на
сантиметри.

  if (echoTime == 0 || distanceCm < 2 || distanceCm > 400) {
    return -1; // Повідомляємо, що вимірювання не вдалося.
  }

  return distanceCm; // Повертаємо виміряну відстань.
}
```

Імпульс на TRIG запускає вимірювання. `pulseIn()` вимірює тривалість відповіді на ECHO у мікросекундах. Ділення на 58 перетворює її на сантиметри.

`return` повертає число до місця виклику функції. Значення `-1` позначає невдале вимірювання.

Чотири режими світлодіода

- Понад 100 см — вимкнений.
- Понад 30 і до 100 см — спалах 80 мс, пауза 600 мс.
- Понад 10 і до 30 см — спалах 80 мс, пауза 150 мс.
- Від 2 до 10 см — світиться постійно.

```
void showDistance(float distanceCm) {

  if (distanceCm < 0) { // Вимірювання не вдалося.
    digitalWrite(LED_PIN, LOW); // Вимикаємо світлодіод.
    delay(100); // Чекаємо перед повторною спробою.

  } else if (distanceCm > 100) { // Перешкода далі ніж 100 см.
    digitalWrite(LED_PIN, LOW); // Залишаємо світлодіод вимкненим.
    delay(100); // Чекаємо до наступного вимірювання.

  } else if (distanceCm > 30) { // Понад 30 і до 100 см включно.
    blinkOnce(600); // Короткий спалах, довга пауза.

  } else if (distanceCm > 10) { // Понад 10 і до 30 см включно.
    blinkOnce(150); // Короткий спалах, коротка пауза.

  } else { // Від 2 до 10 см включно.
    digitalWrite(LED_PIN, HIGH); // Світлодіод світиться постійно.
    delay(100); // Через 100 мс знову виміряємо відстань.
  }

}
```

Перевіряємо умови зверху вниз. Виконується перша гілка, умова якої істинна. Для 20 см це `distanceCm > 10`: викликаємо `blinkOnce(150)`.

Один спалах

```
void blinkOnce(int pauseMs) {
  digitalWrite(LED_PIN, HIGH); // Вмикаємо світлодіод.
  delay(80); // Світимо 80 мс.
  digitalWrite(LED_PIN, LOW); // Вимикаємо світлодіод.
  delay(pauseMs); // Чекаємо до наступного вимірювання.
}
```

Аргумент `pauseMs` задає паузу після спалаху. Під час паузи вимірювань немає: нове почнеться після 680 мс для повільного блимання або після 230 мс для швидкого.

Перевір чотири режими

Запуск у Wokwi

Натисни кнопку запуску. Для самостійного завдання збережи власну копію проєкту.

Готовий проєкт: https://wokwi.com/projects/475804502626573313. Схему можна зібрати вручну за кроками уроку. Повний код — на останній сторінці.

У Wokwi натисни на HC-SR04 та змінюй відстань повзунком. На справжній платі тримай перед датчиком плаский картон на відомій відстані, перевіряючи її лінійкою.

Не притискай предмет до датчика: ближче 2 см показ ненадійний. М’яка тканина, нахилена поверхня або порожній простір можуть не дати відлуння.

1. Відкрий монітор порту зі швидкістю 115200.
2. Установи 120 см — LED вимкнений.
3. Установи 60 см — повільно блимає.
4. Установи 20 см — швидко блимає.
5. Установи 8 см — світиться постійно.
6. Перевір межі: на 100 см блимає повільно, на 30 см — швидко, на 10 см світиться постійно.

Число і світло

У моніторі −1

Перевір `5V`, `GND`, TRIG–`D7`, ECHO–`D6`. Постав плаский картон на 20–50 см перед датчиком.

На 10 см LED не світиться

Від’єднай живлення. Перевір полярність LED, резистор `24d–30d` та `D8–30e`.

Якщо датчик не отримує надійної відповіді, LED вимикається. Щоб зрозуміти причину, перевір число в моніторі порту.

Швидке блимання з 40 см

> Зміни межу швидкого блимання з 30 на 40 см.

1. Знайди умову `distanceCm > 30` і заміни 30 на 40.
2. Перевір 35 см — тепер LED блимає швидко.
3. Перевір 60 см і 8 см — їхні режими залишилися незмінними.

Перевір відповідь

На 35 см виконується blinkOnce(150), на 60 см — blinkOnce(600), на 8 см LED світиться постійно.

Далі — 06-02: ті самі правила зі звуковим сигналом.

УРОК 06-01 · ПОВНИЙ КОД

Відстань і світлодіод

Наближаємо предмет: вимкнений LED → повільне блимання → швидке → постійне світло.

```
#include <Arduino.h> // Команди Arduino.

const byte TRIG_PIN = 7; // Вихід запуску датчика.
const byte ECHO_PIN = 6; // Вхід відповіді датчика.
const byte LED_PIN = 8; // Вихід світлодіода.

float readDistanceCm() {
  digitalWrite(TRIG_PIN, LOW); // Готуємо вихід до нового імпульсу.
  delayMicroseconds(2); // Чекаємо 2 мікросекунди.
  digitalWrite(TRIG_PIN, HIGH); // Починаємо імпульс запуску.
  delayMicroseconds(10); // Тримаємо його 10 мікросекунд.
  digitalWrite(TRIG_PIN, LOW); // Завершуємо імпульс.

  unsigned long echoTime = pulseIn(ECHO_PIN, HIGH, 25000); // Чекаємо відповідь до 25
мс.
  float distanceCm = echoTime / 58.0; // Перетворюємо тривалість відповіді на
сантиметри.

  if (echoTime == 0 || distanceCm < 2 || distanceCm > 400) {
    return -1; // Повідомляємо, що вимірювання не вдалося.
  }

  return distanceCm; // Повертаємо виміряну відстань.
}

void blinkOnce(int pauseMs) {
  digitalWrite(LED_PIN, HIGH); // Вмикаємо світлодіод.
  delay(80); // Світимо 80 мс.
  digitalWrite(LED_PIN, LOW); // Вимикаємо світлодіод.
  delay(pauseMs); // Чекаємо до наступного вимірювання.
}

void showDistance(float distanceCm) {

  if (distanceCm < 0) { // Вимірювання не вдалося.
    digitalWrite(LED_PIN, LOW); // Вимикаємо світлодіод.
    delay(100); // Чекаємо перед повторною спробою.

  } else if (distanceCm > 100) { // Перешкода далі ніж 100 см.
    digitalWrite(LED_PIN, LOW); // Залишаємо світлодіод вимкненим.
    delay(100); // Чекаємо до наступного вимірювання.

  } else if (distanceCm > 30) { // Понад 30 і до 100 см включно.
    blinkOnce(600); // Короткий спалах, довга пауза.
```

```
  } else if (distanceCm > 10) { // Понад 10 і до 30 см включно.
    blinkOnce(150); // Короткий спалах, коротка пауза.

  } else { // Від 2 до 10 см включно.
    digitalWrite(LED_PIN, HIGH); // Світлодіод світиться постійно.
    delay(100); // Через 100 мс знову виміряємо відстань.
  }

}

void setup() {
  pinMode(TRIG_PIN, OUTPUT); // Надсилаємо імпульс датчику.
  pinMode(ECHO_PIN, INPUT); // Приймаємо відповідь датчика.
  pinMode(LED_PIN, OUTPUT); // Керуємо світлодіодом.
  Serial.begin(115200); // Виводимо результати в монітор порту.
}

void loop() {
  float distanceCm = readDistanceCm(); // 1. Вимірюємо відстань.
  Serial.println(distanceCm); // 2. Показуємо число в моніторі порту.
  showDistance(distanceCm); // 3. Вибираємо світловий сигнал.
}
```