

Практичне заняття № 1

Вимірювання відстані за допомогою ультразвукового дальноміра HCSR04

Мета роботи: дослідження функціоналу дальноміра HCSR04 за допомогою мікроконтролерного модуля Arduino.

Завдання: зібрати схему і написати програму для вимірювання відстані за допомогою ультразвукового дальноміра.

Обладнання: мікроконтролер Arduino; проводи; макетна плата; USB – кабель; ультразвуковий дальномір HC-SR04.

Загальні відомості

Дія ультразвукового дальноміра HC-SR04 заснована на принципі ехолокації. Він випромінює звукові імпульси в простір і приймає відбитий від перешкоди сигнал. По часу відбивання звукової хвилі до перешкоди і назад визначається відстань до об'єкта. Відбиття звукової хвилі починається з подачі імпульсу довжиною не менше 10 мікросекунд на вивід **TRIG** дальноміра. Як тільки імпульс закінчується, дальномір випромінює в простір перед собою пачку звукових імпульсів частотою 40 кГц. У цей же час на виводі **ECHO** дальноміра з'являється логічна одиниця. Як тільки датчик приймає відбитий сигнал, на виводі ECHO з'являється логічний нуль. По довжині логічної одиниці на виводі ECHO («Затримка ехо» на рис. 1) визначається відстань до перешкоди.

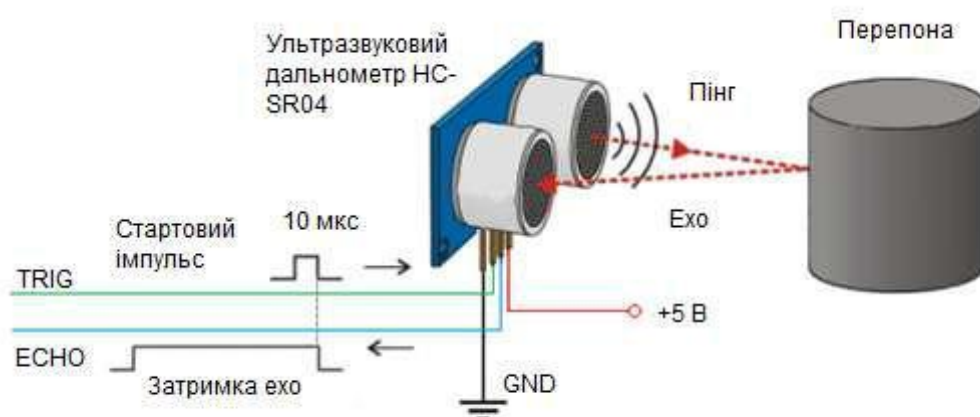


Рис. 1 – Принцип роботи ультразвукового дальноміра

Діапазон вимірів відстані дальноміра HC-SR04 – до 4 метрів з мінімальною градацією шкали в 0,3 см. Кут спостереження – 30°, ефективний кут – 15°. Струм живлення в режимі очікування 2 мА, при роботі – 15 мА.

Опис виводів

- 1 — **VCC** – живлення,
- 2 — **TRIG** – вхід для подачі стробу запуску,
- 3 — **ECHO** – вивід для зняття імпульсу у відповідь,
- 4 — **GND** – земля.

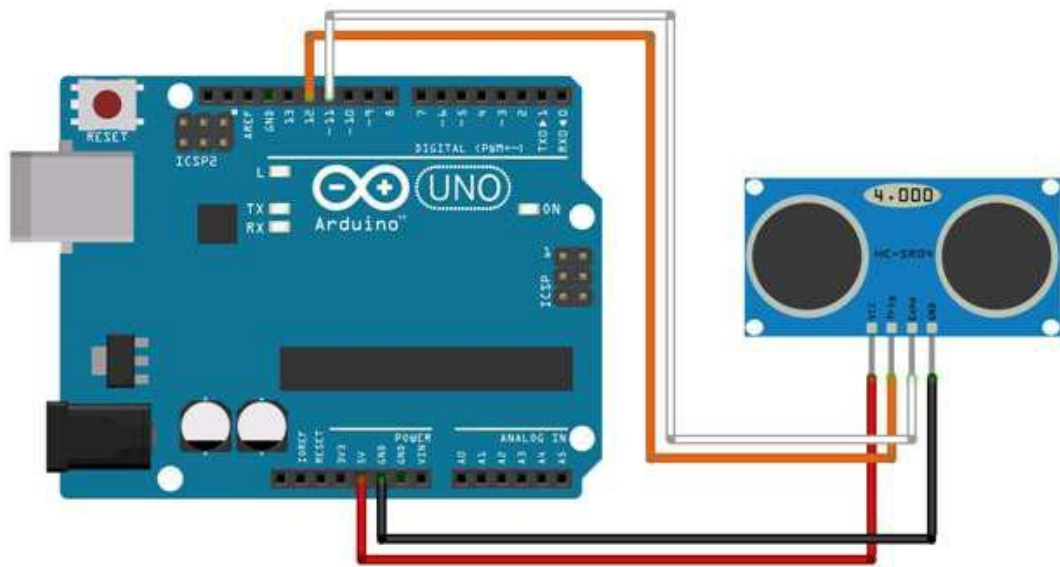


Рис. 2 – Схема підключення дальноміра до плати Arduino

Підключення дальноміру без використання бібліотеки NewPing

Скетч для визначення відстані до перешкоди і виводу визначеного значення в послідовний порт без використання бібліотек наведено нижче. Виводи дальноміра TRIG и ECHO – підключені до виводів 12 и 11 плати Arduino. В кожному повторенні циклу *loop()* обраховується дистанція і її значення виводиться в порт.

Функція *getEchoTiming()* генерує імпульс запуску. Вона створює 10-мікросекундний імпульс, який є тригером для початку випромінювання дальноміром звукового пакету у простір. Далі вона запам'ятовує час від початку передачі звукової хвилі до приходу відлуння (ехо).

Функція *getDistance()* розраховує дистанцію до об'єкта. Відстань дорівнює швидкості помноженої на час: $S = V \times t$.

Швидкість звуку в повітрі 340 м/сек, час в мікросекундах відомий (змінна *duration*). Щоб отримати час *duration* у секундах, необхідно розділити його на 1 000 000. Так як звук проходить подвійну відстань – до об'єкта і назад – треба ще розділити результат на два. Відстань до об'єкта складає:

$$S = 34000 \text{ см/сек} \times \text{duration} / 1\,000\,000 \text{ сек} / 2 = 1,7 \text{ см/сек} / 100.$$

Приклад програми:

```
const int trigPin = 12;
const int echoPin = 11;
void setup() {
    pinMode(trigPin, OUTPUT); // тригер - вихідний пін
    pinMode(echoPin, INPUT); // ехо - вхідний
    Serial.begin(9600); // ініціалізація послід. порту
}
void loop() {
    long distance = getDistance(); // отримуємо дистанцію з
    датчика
    Serial.println(distance); // виводимо в послідовний
    порт delay(100);
}
// Визначення дистанції до об'єкта в см
long getDistance() {
    long distacne_cm = getEchoTiming() * 1.7 *
    0.01; return distacne_cm;
}
// Визначення часу затримки
long getEchoTiming() {
```

```

digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH); // генеруємо імпульс запуску
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// визначення на пині echoPin довжині рівня HIGH, мксек:
long duration = pulseIn(echoPin,
HIGH); return duration;
}

```

Підключення дальноміру з використанням бібліотеки *NewPing*

Конструктор класу *NewPing*:

NewPing sonar (trigger_pin, echo_pin, max_distance_cm).

Ця функція визначає 12 пін як вихід тригера, 11 пін як ехо вхід, з максимальною відстанню вимірювання 200 см.

Основні функції бібліотеки *NewPing*:

- *sonar.ping()* – посилає пінг, повертає час ехо в мікросекундах,
- *sonar.ping_in()* – посилає пінг, повертає відстань в дюймах,
- *sonar.ping_cm()* – посилає пінг, повертає дистанцію в сантиметрах,
- *sonar.ping_median(iterations)* – посилає серію пінгів (за замовчуванням = 5), повертає середнє значення,
- *sonar.convert_in(echoTime)* – конвертує мікросекунди в дистанцію в дюймах
- *sonar.convert_cm(echoTime)* – конвертує мікросекунди в дистанцію в дюймах,
- *sonar.ping_timer(function)* – посилає пінг та викликає функцію, що тестує чи пінг працює,
- *sonar.check_timer()* – перевіряє чи пінг повернувся у межах вказаної максимальної відстані,
- *timer_us(frequency, function)* – викликає функцію з заданою частотою (в мікросекундах),

- *timer_ms(frequency, function)* – викликає функцію з заданою частотою (в мілісекундах),
- *timer_stop()* – зупиняє таймер.

Приклад програми ініціалізації:

```
#include <NewPing.h>
#define TRIGGER_PIN 12 // Arduino pin прив'язаний до тригера
на ультразвуковому датчику..
#define ECHO_PIN 11 // Вивід Arduino прив'язаний до echo-
виводу на ультразвуковому датчику.
#define MAX_DISTANCE 200 // Максимальна відстань, з якою ми
хочемо працювати, тобто пінгувати (у см).
//Максимальна відстань датчика оцінюється в 400-500 см.
// Налаштування контактів і максимальної відстані.
NewPing sonar(TRIGGER_PIN, ECHO_PIN,
MAX_DISTANCE); void setup() {
    Serial.begin(115200); // Відкриття послідовного порту 115200.
}
void loop() {
    delay(50); // Чекаємо 50 мс між пінгами (близько 20
пінг/с).
        //29мс має бути найкоротшою затримкою між пінгами.
    .....
}
```

Хід виконання роботи

1. Зібрати макет відповідно завдання.
2. Підключити схему до живлення(5В).
3. Завантажити програму в мікроконтролер Arduino.
4. Перевірити правильність роботи програми.

Завдання

Створити скетч програми, що буде використовувати функції бібліотеки *NewPing* задані згідно з номером варіанту у табл. 1.

Таблиця 1 – Завдання на лабораторну роботу згідно з номером варіанту

Варіант 1,4,7,10,13, 16,19,22	Варіант 2,5,8,11,14, 17,20,23	Варіант 3,6,9,12,15,18, 21,24
<i>sonar.ping()</i>	<i>sonar.ping_in()</i>	<i>sonar.ping_cm()</i>
<i>sonar.convert_cm(echoTime)</i>	<i>sonar.check_timer()</i>	<i>sonar.ping_timer(function)</i>

Підготувати звіт згідно з ДСТУ 3008-95 (лістинг програми, висновки, перелік посилань).

Контрольні питання

1. Які основні характеристики та опишіть призначення дальноміра HC-SR04?
2. Опишіть виводи, що використовують для підключення дальноміру.
3. Які функції використовують при підключенні дальноміру без використання бібліотеки *NewPing* та сформулюйте алгоритм роботи програми?
4. Зазначте основні особливості бібліотеки *NewPing*?
5. Які функції бібліотеки *NewPing* використовуються для підключення та якісної роботи дальноміра?