Лабораторна робота №4

**Створення алгоритму та управляючої програми для простого конвеєра на базі крокового двигуна, датчика відстані та мікроконтролера ARDUINO**

**Мета роботи:** Ознайомитись із особливостями взаємодії контролера сімейства Arduino з кроковим двигуном та датчиком відстані для побудови систем малої автоматизації.

## 4.1 Теоретичні відомості

### 4.1.1 Короткий опис ультразвукового дальноміра HC-SR04

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 перепони і назад визначається відстань до об'єкта. Відбиття звукової хвилі починається з подачі імпульсу довжиною не менше 10 мікросекунд на вивід TRIG дальнометра. Як тільки імпульс закінчується, дальномір випромінює в простір перед собою пачку звукових імпульсів частотою 40 кГц. В цей же час на виводі ECHO дальнометра 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. 4.1) визначається відстань до перешкоди.

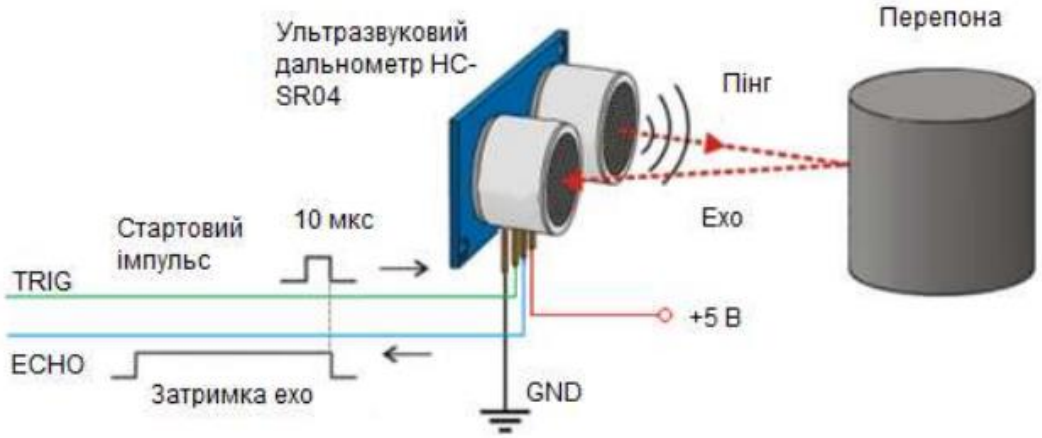


Рис. 4.1. Принцип роботи ультразвукового дальнометра

Діапазон вимірів відстані дальнометра 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 строба запуска, 3 — ECHO – вивід для зняття імпульса у відповідь, 4 — GND – земля.

### 4.1.2 Крокові двигуни

Кроковий двигун — електричний двигун, в якому імпульсне живлення електричним струмом призводить до того, що його ротор не обертається неперервно, а виконує щоразу обертальний рух на заданий кут. Завдяки цьому, кут повороту ротора залежить від числа поданих імпульсів струму, а кутова швидкість ротора точно рівна частоті імпульсів помноженій на кут повороту ротора за один цикл роботи двигуна.

Кут повороту двигуна під впливом одного імпульсу може мати різні значення, залежні від конструкції двигуна, — як правило це значення в діапазоні від декількох градусів до декілька десятків градусів. Крокові двигуни, залежно від призначення пристосовані до виконання від частки оберту на хвилину, до декількох тисяч обертів на хвилину.

Рис.4.2 Кроковий двигун

## 4.2 Виконання роботи

Зібраний мною лабораторний макет буде використовуватись для вимірювання розмірів заданих деталей чи об’єктів. Рух конвеєра буде відбуватись за допомогою крокового двигуна NEMA 17, а вимірювання висоти деталі – ультразвуковим дальноміром HC-SR04. Інформація з датчика буде передаватись на монітор порту, а оператор лінії, відповідно до отриманих розмірів, буде сортувати деталі.

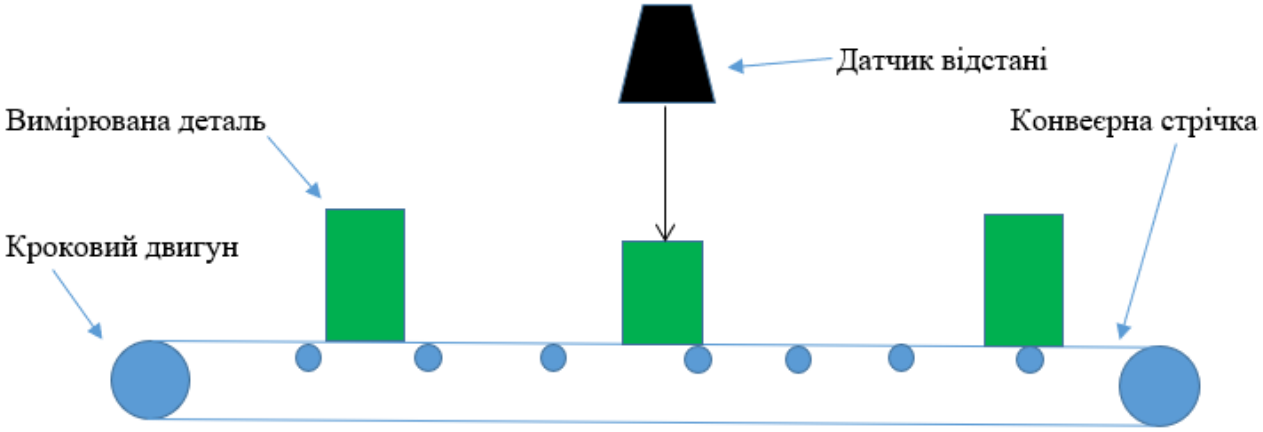


Рис. 4.3. Функціональна схема

Виводи дальнометра TRIG и ECHO – підключені до виводів 4 и 3 плати Arduino. В кожному повторенні цикла 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 дальнометром звукового пакета в простір. Далі вона запам’ятовує час від початку передачі звукової хвилі до прихода еха.

Функція getDistance() розраховує дистанцию до обєкта. Відстань дорівнює швидкість помножити на час: S = V×t. Швидкість звуку в повітрі 340 м/сек, час в мікросекундах ми знаемо (змінна duratuion). Щоб отримати час duratuion в секундах, необхідно розділити його на 1 000 000. Так як звук проходить подвійну відстань – до обєкта і назад – треба ще розділити результат на два.

Щоб керувати двигуном, скористаємося бібліотекою Stepper, яку не потрібно ні від куди качати – вона входить в поставку Arduino IDE – середовище розробки, в якій ми пишемо код для arduino. В якості параметрів передаємо кількість кроків повного обороту і піни, до яких підключаємо двигун.

Stepper myStepper(2048,2,3,4,5);

Методом setSpeed () можна керувати швидкістю, а step() відповідає за переміщення валу на задану кількість кроків (якщо значення негативне – вал крутиться у зворотний бік).

Наводимо схему підключення:

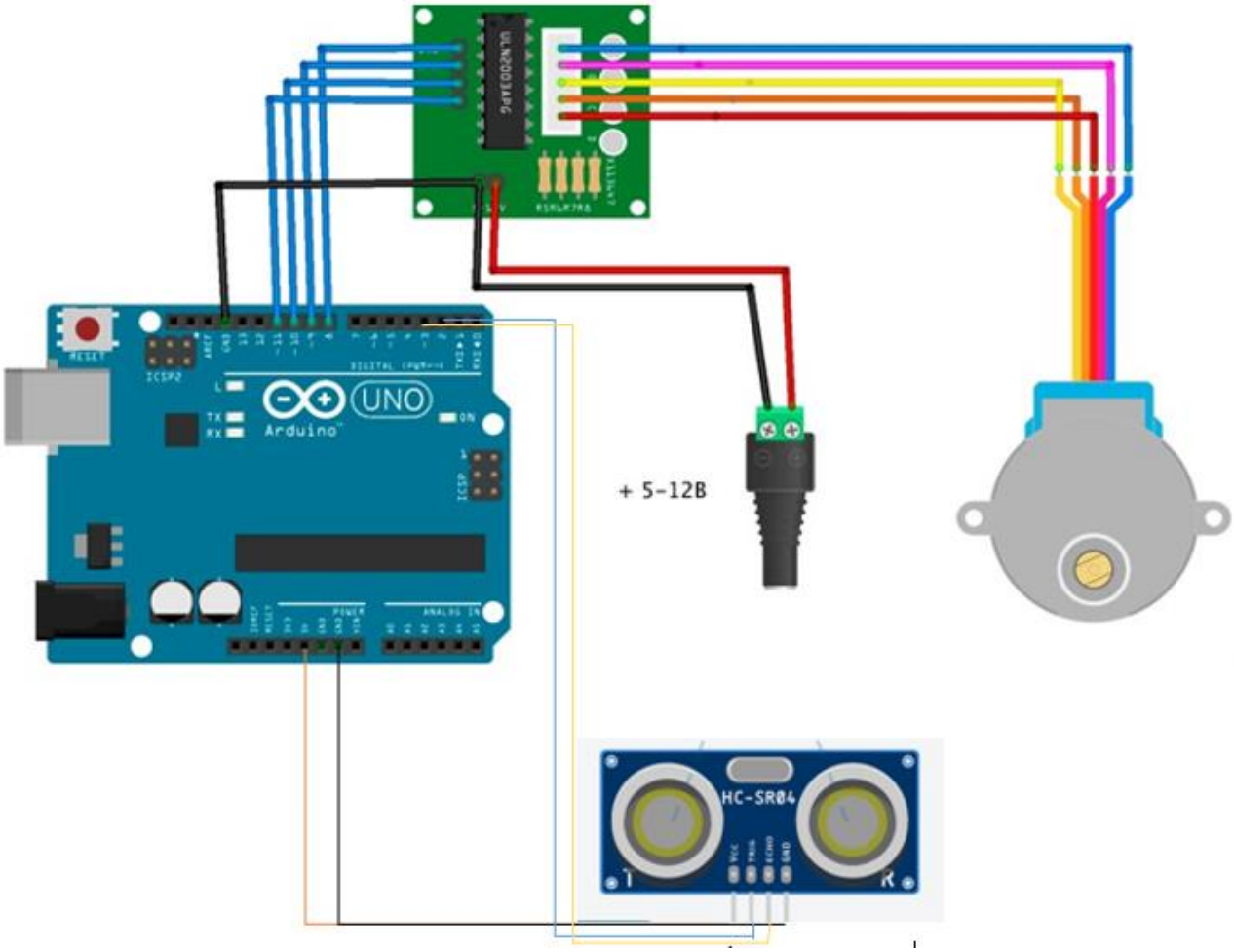


Рис. 4.4. Загальна схема підключення

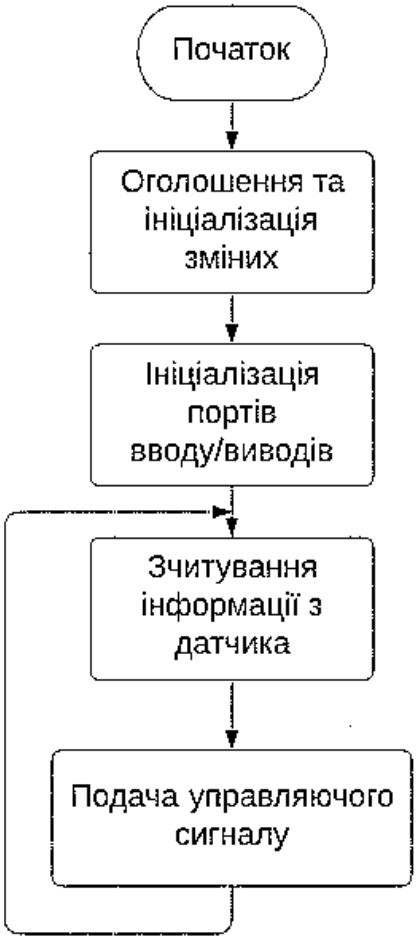


Рис. 4.5. Алгоритм роботи програми управління

Код програми управління:

```
const int trigPin = 4;
const int echoPin = 3;
#include <Stepper.h>
const int stepsPerRevolution = 200;
Stepper myStepper(stepsPerRevolution, 8,9,10,11);
void setup() {
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
 myStepper.setSpeed(60);
 Serial.begin(9600);
}
void loop() {
```

```
 Serial.println("Move right"); //за часовою стрілкою
 myStepper.step(stepsPerRevolution);
 long distance = getDistance();
 Serial.println(distance);
 delay(100);
}
long getDistance() {
 long distacne_cm = getEchoTiming() * 1.7 * 0.01;
 return distacne_cm;
}
long getEchoTiming() {
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 long duration = pulseIn(echoPin, HIGH);
 return duration;
}
```

**ЗАВДАННЯ**

**1. Реалізувати вище вказану систему в програмному середовищі TinkerCAD;**

**2. Модернізувати вище вказану систему на власний розсуд;**

**3. Оформити звіт та сформувати посилання для перевірки працездатності коду.**