

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Факультет електроніки  
(повна назва інституту/факультету)

Кафедра акустичних та мультимедійних електронних систем  
(повна назва кафедри)

"На правах рукопису"  
УДК 681.8

"До захисту допущено"  
Завідувач кафедри



Найда С.А.  
(ініціали, прізвище)

07 грудня 2020 р.

## Магістерська дисертація

спеціальність 171 Електроніка  
(код та назва спеціальності)

на тему: "Розробка макету акустичної системи виявлення перешкод за допомогою ультразвукового далекоміра HC-SR04 Arduino"

Виконав: студент II курсу, групи ДГ-91мп  
(шифр групи)

Сінчук Ігор Олександрович  
(прізвище, ім'я, по батькові)



(підпис)

Науковий керівник ас., к.т.н., Старовойт Ярослав Іванович  
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

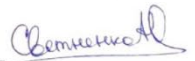


(підпис)

Консультант \_\_\_\_\_  
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент Головний інженер ДП "КНДІ Гп", к.т.н., Святненко А.О.  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)



(підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент   
(підпис)

Київ – 2020 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет електроніки  
(повна назва)

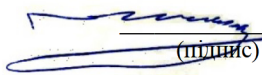
Кафедра акустичних та мультимедійних електронних систем  
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (освітня програма) 171 Електроніка (Акустичні електронні системи та технології обробки акустичної інформації)  
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

  
(підпис) Найда С.А.  
(ініціали, прізвище)

" 07" грудня 2020 р.

**ЗАВДАННЯ**

**на магістерську дисертацію студенту**

Сінчуку Ігорю Олександровичу  
(прізвище, ім'я, по батькові)

1. Тема дисертації: «Розробка макету акустичної системи виявлення перешкод за допомогою ультразвукового далекоміра HC-SR04 Arduino»

Науковий керівник дисертації Старовойт Ярослав Іванович,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від "05" листопада 2020 р. № 3241-с

2. Строк подання студентом дисертації \_\_\_\_\_ .12.2020 р.

3. Об'єкт дослідження: Система ультразвукового далекоміру.

4. Предмет дослідження (Вихідні дані – для магістерської дисертації за освітньо-професійною програмою) Макет ультразвукового далекоміру на мікроконтролері.

5. Перелік завдань, які потрібно розробити: Огляд мікроконтролерних систем; Програмні засоби проектування програмного коду; Проектування приладу та коду вимірювання відстані.

6. Перелік графічного (ілюстративного) матеріалу: 27 рис, 20 табл., 1 презентація, 10 слайдів


7. Орієнтовний перелік публікацій

8. Дата видачі завдання 10.09.2019 р.

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Написання першого розділу: " Загальна класифікація далекомірів "	15.03.2020	Виконано
2	Написання другого розділу: " Програмні засоби реалізації моделі далекоміру "	30.09.2020	Виконано
3	Написання третього розділу: "Реалізація ультразвукового далекоміру на мікроконтролері"	30.10.2020	Виконано
4	Написання четвертого розділу: "Розроблення стартап проекту"	25.11.2020	Виконано
5	Підготовка матеріалів до друку та оформлення пояснювальної записки	24.11.2020	Виконано
6	Підготовка та оформлення презентації для доповіді	03.12.2020	Виконано

Студент

  
\_\_\_\_\_  
(підпис)

Сінчук І.О.  
(ініціали, прізвище)

Керівник роботи

  
\_\_\_\_\_  
(підпис)

Старовойт Я.І.  
(ініціали, прізвище)

## РЕФЕРАТ

*Сінчук І.О. Розробка макету акустичної системи виявлення перешкод за допомогою ультразвукового далекоміра HC-SR 04 Arduino: магістерська дис.: 171 Електроніка. Київ, КПІ ім. Ігоря Сікорського, 2020, 62 с.*

**Актуальність теми:** робота присвячена дослідженню мікроконтролерних систем та способів управління ними. В результаті виконання роботи був створений діючий прототип моделі управління платою Arduino UNO для вимірювання відстані. Для демонстрації її роботи був створений макет, який за допомогою ультразвукового далекоміру вимірює відстань та виводить її на дисплей.

**Мета дослідження:** створення автономної системи для вимірювання відстані ультразвуковим методом на базі мікроконтролерів.

Для вирішення поставленого завдання необхідно вирішити такі питання:

- Провести огляд основних елементів;
- Провести огляд технічних параметрів та функціональних можливостей систем від різних виробників;
- Запропонувати власну систему для вимірювання відстані ультразвуковим методом на базі мікроконтроллера.

**Об'єкт дослідження:** системи ультразвукового вимірювання відстані.

**Предмет дослідження:** макет автономного пристрою для вимірювання відстані за допомогою ультразвуку на базі мікроконтроллера.

**Методи дослідження:** теоретичний огляд основних компонентів систем ультразвукових далекомірів, теоретичний огляд можливостей даних систем та аналіз світових рішень, розробка та впровадження власної системи.

**Практичне значення одержаних результатів** полягає у проектуванні та реалізації макету та порівняння його із існуючими аналогами.

**Ключові слова:** решітки ультразвукових перетворювачів, мікроконтролери, цифрові схеми, мікропрограмування, далекомірні системи

## ABSTRACT

*Sinchuk I. Development of model of acoustic system for detecting obstacles based on ultrasonic range finder HC-SR04 Arduino: master's thesis: 171 Electronics, Kyiv, Igor Sikorsky Kyiv Polytechnic Institute, 2020, 62 p.*

**Relevance of the topic:** the thesis is devoted to the study of microcontroller systems and methods of their control. As a result of the study, a working prototype of the Arduino UNO board control model for distance measurement was created. To demonstrate its work, a model was created that uses an ultrasonic range finder to measure distance and display it. It is recommended to use this work as a basis for further study of technologies and development of new systems.

**The aim of the study:** create autonomous system for measuring distances using ultrasonic methods and microcontrollers.

To obtain this aim we need to solve next problems:

- Make the review of the main elements;
- Make the review of technical parameters and functional possibilities of such systems from different manufacturers;
- Offer own system for measuring distance using ultrasonic methods and microcontrollers.

**Object of study:** ultrasonic system for measuring distance.

**Subject of study:** model of autonomous for measuring distance using ultrasonic transducers and microcontrollers.

**Research methods:** theoretical review of the main parts, characteristics of such systems. Analysis of industrial systems, and proposing of own system.

**Practical meaning of the obtained results** is in designing and implementation of the model and comparing it with industrial examples.

**Key words:** ultrasonic transducer arrays, microcontrollers, digital circuits, microprogramming, distance measurement systems

## ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. ЗАГАЛЬНА КЛАСИФІКАЦІЯ ДАЛЕКОМІРІВ	8
1.1. Відстань визначення та класифікація	8
1.2. Методи вимірювання дальності.	11
1.3. Принцип роботи ультразвукових далекомірів.	13
1.4. Мікроконтролерні системи.	14
1.5. Arduino UnoR3 – загальні відомості.	17
1.6. Класифікація далекомірних систем за типом вимірювання.	22
1.7. Загальні відомості датчика відстані HC-SR04.	24
1.8. Семисегментний індикатор 4-розрядний з точками.	29
1.9. Висновки	32
РОЗДІЛ 2. ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ МОДЕЛІ ДАЛЕКОМІРУ	33
2.1. Arduino IDE	33
2.2. Основи скетчу Arduino.	38
2.3. Структура програми в Arduino IDE.	40
2.4. Висновки	41
РОЗДІЛ 3. РЕАЛІЗАЦІЯ УЛЬТРАЗВУКОВОГО ДАЛЕКОМІРУ НА МІКРОКОНТРОЛЛЕРІ	42
3.1. Блок схема алгоритму роботи ультра звукового далекоміру.	42
3.2. Проектування функціональної, принципової та структурної схем ультразвукового далекоміру на мікроконтроллері.	43
3.3. Написання коду скетчу для ультразвукового далекоміра.	45
3.4. Написання коду скетчу для чотирьох секційного семисегментного індикатора.	46
3.5. Тестування та результати роботи макету	49
3.6. Висновки	49
4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ	50
4.1. Опис ідеї проекту	50
4.2. Технологічний аудит ідеї проекту	52
4.3. Аналіз ринкових можливостей запуску стартап-проекту	52
4.4. Розроблення ринкової стратегії проекту	56
4.5. Розроблення маркетингової програми стартап-проекту	58
4.6. Висновки	59
ВИСНОВОК	60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ	61
ДОДАТКИ	63
Додаток А. Лістинг скетчу далекоміру.	63

## ВСТУП

Ультразвукові далекоміри - це датчики, які можуть вимірювати відстань. Ви можете використовувати їх, щоб знайти відстань до об'єкта або виявити, коли щось знаходиться поруч із датчиком, як детектор руху.

Вони ідеально підходять для проектів, що стосуються навігації, уникнення об'єктів та домашньої безпеки. Оскільки вони використовують звук для вимірювання відстані, вони працюють так само добре в темряві, як і на світлі.

Ультразвуковий далекомір, який використано в даній роботі, є HC-SR04, який може вимірювати відстань від 2 см до 400 см з точністю до  $\pm 3$  мм.

В якості основного об'єкту для дослідження була обрана Arduino UNO на основі якої побудовано макет приладу для вимірювання відстані.

У рамках даної роботи показано, як розробити макет далекоміра для мікроконтролера. Далекомір не потрібно налаштувати і має досить хорошу точність.

## РОЗДІЛ 1.

### ЗАГАЛЬНА КЛАСИФІКАЦІЯ ДАЛЕКОМІРІВ

#### 1.1. Відстань визначення та класифікація

Відстань - це чисельне вимірювання відстані між собою предметів або точок. У фізиці чи повсякденному використанні відстань може стосуватися фізичної довжини або оцінки, заснованої на інших критеріях (наприклад, "на два округи більше"). Відстань від точки А до точки В іноді позначають як  $|AB|$  або  $|AB|$ . [1] У більшості випадків "відстань від А до В" взаємозамінна з "відстань від В до А". [2] У математиці функція відстані або метрика - це узагальнення поняття фізичної відстані; це спосіб описати, що означає для елементів деякого простору бути «близько» або «далеко від» один одного. У психології та соціальних науках відстань є нечисельним виміром; Психологічна дистанція визначається як "різні способи, якими об'єкт може бути вилучений з" Я за такими розмірами, як "час, простір, соціальна дистанція та гіпотетичність".

Термін "відстань" також використовується за аналогією для певного вимірювання нефізичних сутностей.

Фізична відстань може означати кілька різних речей:

- Пройдена відстань: довжина певного шляху, пройденого між двома точками, наприклад, пройдена відстань під час орієнтації в лабіринті

- Прямолінійна (евклідова) відстань: довжина найкоротшого можливого шляху через простір між двома точками, яку можна було б пройти, якби не було перешкод (зазвичай формалізована як евклідова відстань )

- Геодезична відстань: довжина найкоротшого шляху між двома точками, залишаючись на деякій поверхні, наприклад, відстань великого кола вздовж кривої Землі

- Довжина конкретного шляху, який повертається до вихідної точки,



наприклад, кулі, кинутої прямо вгору, або Землі, коли вона завершує одну орбіту.

Кругова відстань - це відстань, пройдена колесом, яка може бути корисною при проектуванні транспортних засобів або механічних передач. Окружність колеса дорівнює  $2 \times$  радіус, і якщо припустити, що радіус дорівнює 1, тоді кожен оберт колеса еквівалентний відстані 2 радіанів. У техніці часто використовується  $\lambda = 2 f$ , де  $f$  - частота .

Незвичні визначення відстані можуть бути корисними для моделювання певних фізичних ситуацій, але також використовуються в теоретичній математиці:

Манхеттенська відстань - це прямолінійна відстань, названа за кількістю кварталів (у північному, південному, східному чи західному напрямках), на яких повинен проїхати таксі, щоб дістатися до місця призначення на сітці вулиць в частині Нью-Йорка .

Шахова дошка, формалізована як відстань Чебишева, - це мінімальна кількість ходів, які король повинен зробити на шаховій дошці , щоб подорожувати між двома квадратами.

Міри відстані в космології ускладнюються розширенням Всесвіту та ефектами, описаними теорією відносності (наприклад, скорочення довжини рухомих об'єктів).

В інформатиці існує поняття " відстань редагування " між двома рядками. Наприклад, слова "собака" і "крапка", які змінюються лише на одну букву, ближчі, ніж "собака" і "кішка", які відрізняються трьома буквами. Ця ідея використовується в засобах перевірки правопису та в теорії кодування і математично оформлена кількома різними способами, такими як:

- Відстань Левенштейна
- Відстань Хеммінга
- Відстань Лі
- Відстань Яро – Вінклер

У математиці метричний простір - це множина, для якої визначаються

відстані між усіма членами множини. Таким чином можна розрахувати багато різних типів "відстаней", таких як обхід графіків, порівняння розподілів та кривих та використання незвичних визначень поняття "простір" (наприклад, за допомогою колектора чи відбиттів). Поняття відстані в теорії графів використовувалося для опису соціальних мереж, наприклад, за номером Ерда чи числом Бекона - кількість взаємовідносин у співпраці далеко від людини від плідного математика Пола Ердеса та актора Кевіна Бекона відповідно.

У психології, людській географії та соціальних науках відстань часто теоретизується не як об'єктивна метрика, а як суб'єктивний досвід.

Як відстань, так і переміщення вимірюють рух предмета. Відстань не може бути негативною і ніколи не зменшується. Відстань - це скалярна величина або величина, тоді як переміщення - це векторна величина як з величиною, так і з напрямком. Він може бути негативним, нульовим або позитивним. Направлена відстань не вимірює рух; він вимірює поділ двох точок і може бути позитивним, нульовим або негативним вектором.

Відстань, пройдена транспортним засобом (наприклад, в вигляді записаного з допомогою одометра), людини, тварини або об'єкта уздовж зігнутої траєкторії від точки А до точки В, слід відрізняти від прямолінійного відстані від А до В. Наприклад, якою б не була відстань, подолана під час поїздки в обидва кінці від А до В і назад до А, переміщення дорівнює нулю, оскільки початкові та кінцеві точки збігаються. Як правило, пряма відстань не дорівнює пройденій відстані, за винятком подорожей прямою лінією.

Направлені відстані можна визначити по прямих лініях та по кривих.

Направлені відстані вздовж прямих - це вектори, що дають відстань і напрямок між початковою і кінцевою точками. Напряmlена відстань точки С від точки А у напрямку В на прямій АВ у векторному просторі Евкліда - це відстань від А до С, якщо С падає на промінь АВ, але є від'ємником цієї відстані, якщо С падає на промінь ВА (Тобто, якщо С не знаходиться на тій же стороні А, що і В). Наприклад, спрямована відстань від стовпа прапора Головної бібліотеки

Нью-Йорка до стовпа прапора Статуї Свободи має:

- Початкова точка: прапор бібліотеки
- Кінцева точка: стовп прапора статуї
- Напрямок: -38
- Відстань: 8,72 км

Інший вид спрямованої відстані - це між двома різними частинками або точковою масою в даний момент часу. Наприклад, відстань від центру ваги Землі А та центру ваги Місяця В (що не суворо передбачає рух від А до В ) відноситься до цієї категорії.

Направлена відстань вздовж кривої лінії не є вектором і представлена відрізком цієї кривої лінії, визначеної кінцевими точками А і В , з деякою конкретною інформацією, що вказує на сенс (або напрямок) ідеального чи реального руху від однієї кінцевої точки сегментувати до іншого (див. рисунок). Наприклад, просто позначення двох кінцевих точок як А і В може вказувати на сенс, якщо передбачається упорядкована послідовність ( А , В ), що означає, що А є початковою точкою.

## **1.2. Методи вимірювання дальності.**

В основі вимірювання дальності лежать принципи локації. Виділяють два види локації:

- пасивна локація заснована на прийомі власного випромінювання об'єкта;
- при активній локації випромінюється свій власний зондуючий імпульс і приймається його відбиття від цілі. Залежно від параметрів прийнятого сигналу визначаються характеристики цілі.

Активна локація в свою чергу поділяється на:

- локацію безперервної дії;
- локацію імпульсної дії.

Перевагою системи такого типу є дешевизна і простота використання,

однак у таких системах сильно ускладнено вимірювання відстані до об'єкта.

В основі вимірювання дальності методом імпульсної локації лежить вимірювання часу затримки  $t_z$  сигналів. Залежно від параметрів сигналу, в яких міститься інформація про  $t_z$ , розрізняють амплітудний, частотний і фазовий метод вимірювання відстані.

Амплітудний метод характеризується використанням імпульсної модуляції випромінюваних коливань. У цьому випадку на передачу і прийом сигналів працює один приймач. Нижче наведено структурну схему імпульсної локаційної системи. Перемикач (П) під'єднує УЗ трансмітер (УЗТ) до генератора пачки УЗ імпульсів (ГІ) на час ти генерації УЗ імпульсу і до приймача (ПР) на все інше час, обумовлений періодом проходження імпульсів.

Зондуєчий сигнал являє послідовність імпульсів з періодом  $T_p$ , який вибирається з умови однозначності виміру дальності:  $R_{max} = V_{zv} \cdot T_p / 2$ , де  $V_{zv}$  швидкість звуку.

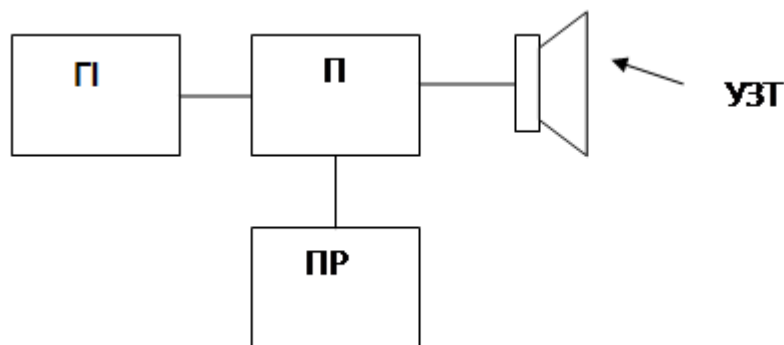


Рисунок 1.1 Структурна схема імпульсної ЛЗ.

Переваги імпульсних схем локації полягають у можливості використання одного оборотного УЗ випромінювача для випромінювання і прийому сигналів, зручність одночасного вимірювання дальностей багатьох об'єктів. Недоліки полягають в обмеженні імпульсної потужності передавача і в пов'язаній з цим обмеження  $R_{max}$ .

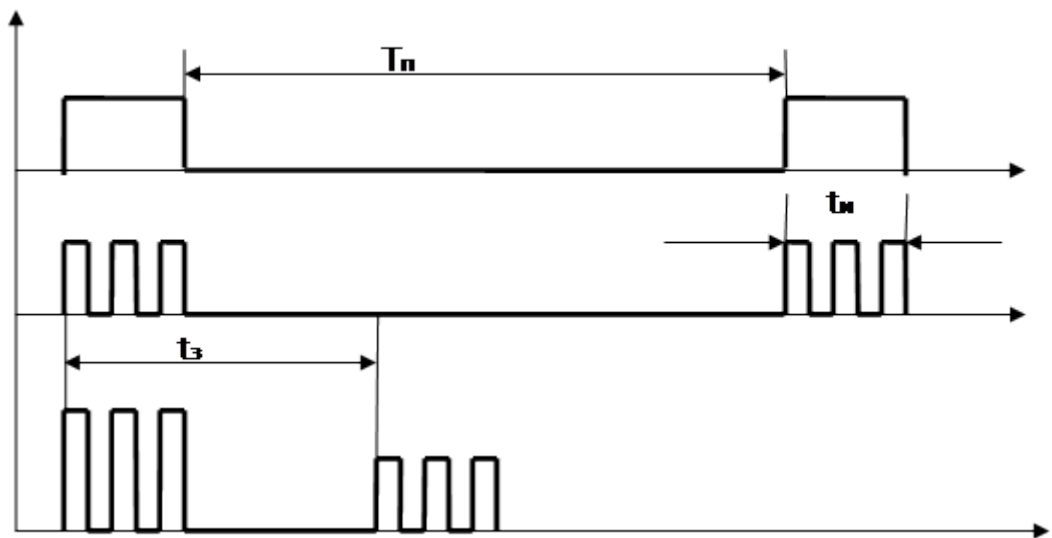


Рисунок 1.2 Часова діаграма імпульсної ЛЗ

### 1.3. Принцип роботи ультразвукових далекомірів.

Ультразвукові далекоміри діють за тим же принципом, що і ехолоти. Вбудоване джерело випромінює сигнал, а вмонтований у цей же корпус приймач приймає хвилі, відбиті від різних перешкод.

Далі прилад аналізує часовий інтервал між двома «подіями» і видає відомості про відстань і вигляді цифрового запису на дисплеї. Ультразвук відбивається в тому напрямку, звідки прийшов вихідний промінь. Розсіювання відносно мало залежить від напрямку падаючого променя і, отже, дозволяє краще візуалізувати відбиваючі поверхні. Відстань до відбивача одно  $1/2$  твори швидкості ультразвуку в середовищі на час між випромінюванням і прийомом відбитого сигналу (рис. 1.3). Твір швидкості на час ділиться навпіл, так як ультразвук проходить подвійний шлях (від випромінювача до відбивача і назад). При цьому досягається дозволу до 0,2 мм. Максимальна дальність дії даних приладів порівняно невелика - до 30 м.

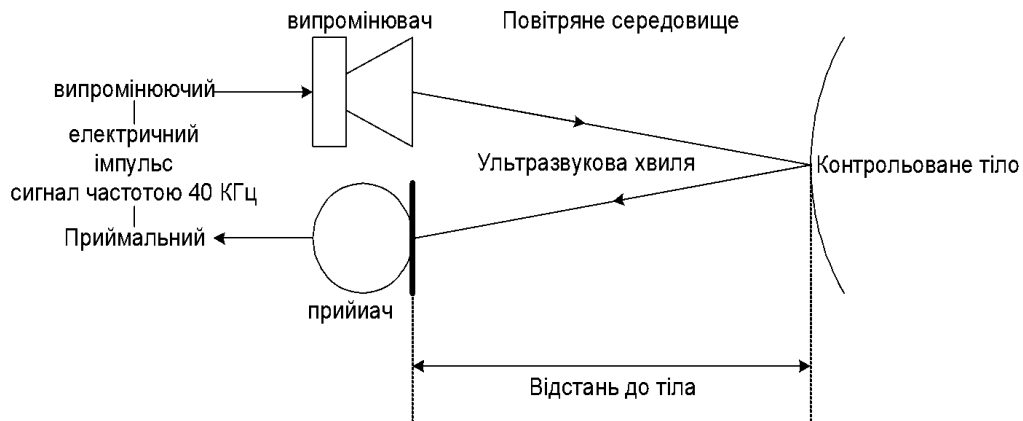


Рисунок 1.3 Загальна схема ультразвукового далекоміра

Для отримання ультразвуку використовуються спеціальні випромінювачі, які перетворюють електричну енергію в звукову, використовуючи зворотний п'єзоелектричний ефект. На додаток ультразвуковими датчиками можна вимірювати порівняно великі відстані, при цьому, зберігаючи їх невеликі розміри, що може бути істотно для ряду застосувань. [1]

#### 1.4. Мікроконтролерні системи

AVR це сім'я мікроконтролерів, розроблена компанією Atmel з 1996 року придбана компанією Microchip Technology у 2016 році. Це модифікована Гарвардська архітектура 8-бітових одноконтрольних мікроконтролерів RISC AVR була однією з перших сімейств мікроконтролерів, яка використовувала флеш-пам'ять на мікросхемі для зберігання програми, на відміну від одноразового програмованого ПЗУ EPROM або EEPROM, який використовували інші мікроконтролери в той час.

Мікроконтролери AVR знаходять багато застосувань як вбудовані системи Вони особливо поширені в аматорських і освітніх вбудованих додатках, популяризували їх включення в багатьох з Arduino лінії відкритої апаратної макетних плат.

AVR використовувались у різних автомобільних додатках, таких як системи

безпеки, безпеки, силових агрегатів та розваг. Аtmель нещодавно випустив нову публікацію "Atmel Automotive Compilation", щоб допомогти розробникам з автомобільними додатками. Деякі сучасні звичаї у BMW, Daimler-Chrysler та TRW.

Arduino фізична обчислювальна платформа заснована на ATmega328 мікроконтролера (ATmega168 або ATmega8 в настільних версіях старіших ніж Diecimila). ATmega1280 та ATmega2560, що мають більше можливостей для розширення та пам'яті, також були використані для розробки платформи Arduino Mega Плати Arduino можна використовувати з його мовою та IDE або з більш звичайними середовищами програмування C асемблер тощо) як просто стандартизовані та широко доступні платформи AVR.

AVR на основі USB використовувались у ручних контролерах Microsoft Xbox. Посилання між контролерами та Xbox USB.

Численні компанії випускають мікроконтролерні плати на основі AVR, призначені для використання любителями, роботобудівниками, експериментаторами та розробниками невеликих систем, включаючи: Cubloc, gnusb, BasicX Oak Micros, мікроконтролери ZX, myAVR. Також існує велика спільнота ардуїно-сумісних плат, які підтримують подібних користувачів.

Архітектура пристрою: Flash EEPROM та SRAM інтегровані в єдиний чіп, що усуває потребу у зовнішній пам'яті у більшості програм. Деякі пристрої мають паралельну зовнішню шину, що дозволяє додати додаткову пам'ять даних або пристрої, орієнтовані на пам'ять. Практично всі пристрої (крім найменших мікросхем TinyAVR) мають послідовний інтерфейс, який можна використовувати для підключення великих послідовних EEPROM або флеш-чіпів.

Пам'ять програми: інструкції програми зберігаються у енергонезалежній флеш-пам'яті Хоча MCU 8-бітні, кожна інструкція займає одне або два 16-бітні слова.

Розмір пам'яті програми зазвичай вказується в іменуванні самого пристрою

(наприклад, лінія ATmega64x має 64 Кб спалаху, а лінія ATmega32x 32 КБ).

Не передбачено пам'ять програм без мікросхеми; весь код, виконаний ядром AVR, повинен знаходитись у мікросхемі. Однак це обмеження не поширюється на мікросхеми AT94 FPSLIC AVR FPGA.

Внутрішня пам'ять даних: адресний простір даних складається з файлів регістрів, регістрів вводу виводу та SRAM Деякі невеликі моделі також відображають програму ROM в адресний простір даних, але більші моделі ні.

EEPROM - майже всі мікроконтролери AVR мають внутрішній EEPROM для напівпостійного зберігання даних. Як і флеш-пам'ять, EEPROM може підтримувати її вміст при відключенні електроенергії.

У більшості варіантів архітектури AVR ця внутрішня пам'ять EEPROM не відображається в адресному просторі пам'яті MCU. Доступ до нього можна лише таким же чином, як і зовнішній периферійний пристрій, використовуючи спеціальні регістри покажчиків та інструкції з читання запису, що робить доступ до EEPROM набагато повільніше, ніж інші внутрішні ОЗУ.

Лінія AVR може підтримувати тактову частоту від 0 до 20 МГц, а деякі пристрої досягають 32 МГц. Робота з меншим живленням зазвичай вимагає зменшення тактової частоти. Усі останні (Tiny, Mega та Xmega, але не 90S) AVR мають вбудований осцилятор, що усуває необхідність у зовнішніх синхронізованих або резонаторних схемах. Деякі AVR також мають системний програмовий годинник, який може розділити синхросигнал на 1024. Цей доскалер може бути налаштований програмним забезпеченням під час роботи, що дозволяє оптимізувати тактову частоту.

Оскільки всі операції (за винятком множення та 16-бітного додавання віднімання) на регістри R0 R31 одноциклічними, AVR може досягти до 1 MIPS на МГц, тобто процесор 8 МГц може досягти до 8 MIPS. Навантаження та зберігання в з пам'яті займають два цикли, розгалуження займає два цикли. Гілки в останніх "3-байтових ПК" частинах, таких як ATmega2560, на один цикл повільніше, ніж на попередніх пристроях.

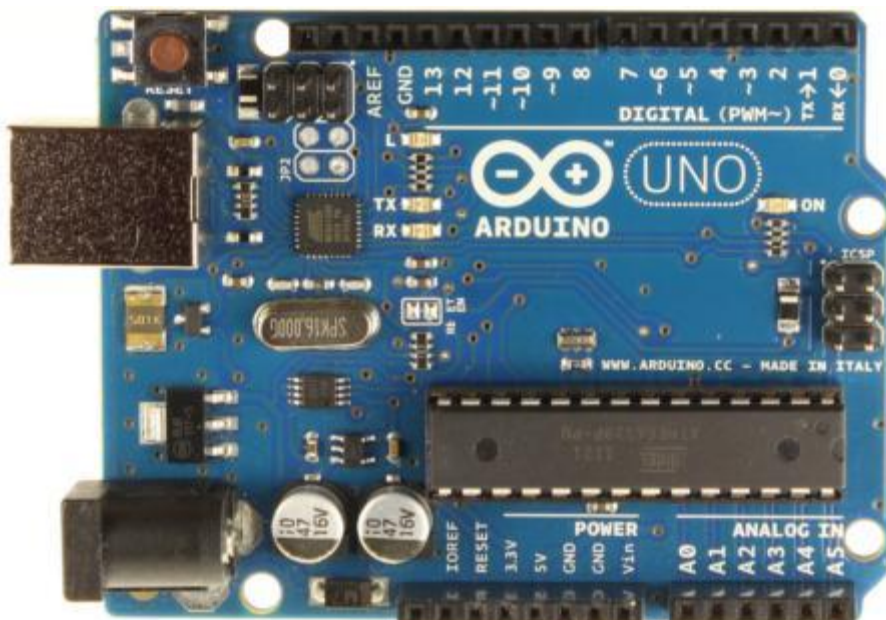


Завдяки доступним безкоштовним та недорогим інструментам розробки AVR такими великими можливостями, зокрема, доступними дошками розробок та безкоштовним програмним забезпеченням для розробки. AVR продаються під різними іменами, які мають одне і те ж основне ядро, але з різними периферійними та комбінаційними пам'яттями. Сумісність між мікросхемами в кожній родині досить хорошою, хоча функції контролера вводу виводу можуть відрізнятися.

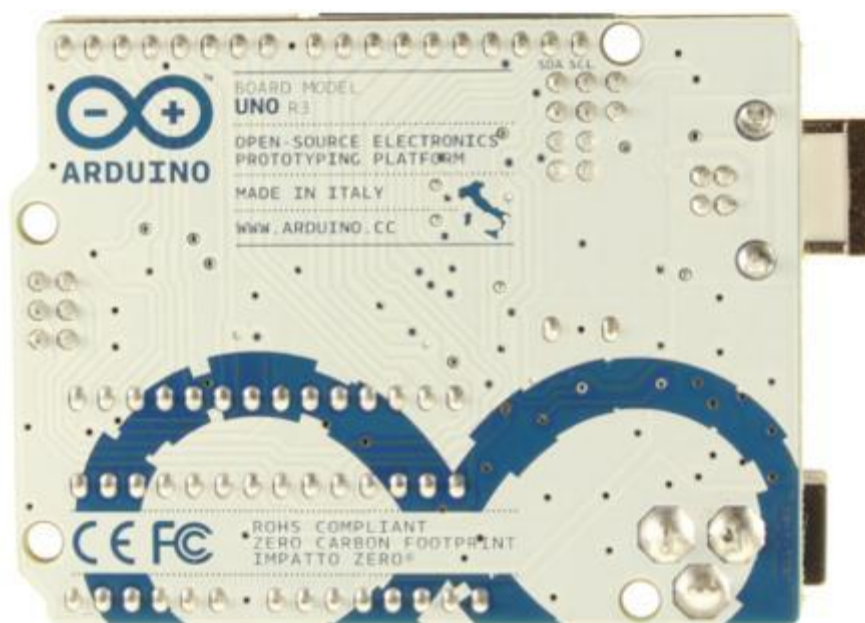
### **1.5. Arduino UnoR3 – загальні відомості**

Arduino Uno - це мікроконтролерна система на базі ATmega328P. Він має 14 пінів вводу / виводу (з них 6 можна використовувати як вихід ШІМ), 6 аналогових входів / виходів, керамічний резонатор 16 МГц (CSTCE16M0V53-R0), USB-з'єднання та роз'єм живлення, заголовок ICSP та кнопку скидання. Він містить все необхідне для підтримки мікроконтролера його лише потрібно підключити до комп'ютера за допомогою кабелю USB або підключити його за допомогою адаптера змінного струму до акумулятора постійного струму або батареї, щоб почати роботу. Ви можете поводитися зі своїм Uno, не турбуючись надто про те, що робити щось не так, у гіршому випадку ви можете замінити чіп за кілька доларів і почніть заново[8].

"Uno" означає італійську мову, і вона була обрана для ознайомлення з випуском Arduino Software (IDE) 1.0. Дошка Uno та версія 1.0 програмного забезпечення Arduino (IDE) були референтними версіями Arduino, тепер еволюціонували до нових версій. Дошка Uno - перша в серії плати USB Arduino та еталонна модель для платформи Arduino; докладний перелік поточних, минулих або застарілих дощок див. в індексі Arduino[8].



*Рис. 1.4 Arduino Uno R3 - видгяд спереду.*



*Рис. 1.5 Arduino Uno R3 - видгядззаду.*

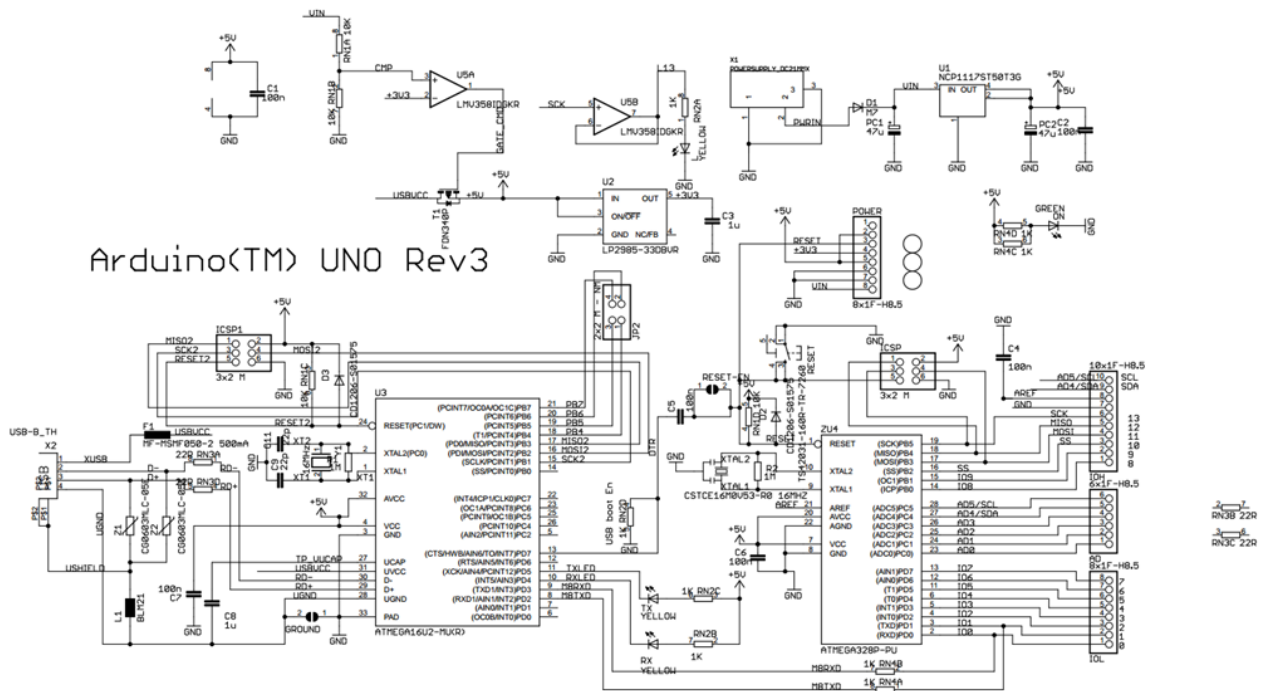


Рис. 1.6 Принципова електрична схема Arduino Uno R3.

Нижче перераховані виходи живлення, розташовані на платі:

- VIN. Напряга, що надходить в Arduino безпосередньо від зовнішнього джерела живлення. Через цей вивід можна як подавати зовнішнє живлення, так і споживати струм, коли пристрій живиться від зовнішнього адаптера.
- 5V. На виході надходить напруга 5V від стабілізатора напруги на платі, поза незалежності від того, як живиться пристрій: від адаптера (7 - 12В), від USB (5В) або через вивід VIN (7 - 12В).
- 3V3. 3.3В, що надходять від стабілізатора напруги на платі. Максимальний струм, споживаний від цього висновку, становить 50 мА.
- GND. Виходи землі.
- IOREF. Цей вихід надає платам розширення інформацію про робочій напрузі мікроконтролера Ардуіно. Залежно від напруги, ліченого з виведення IOREF, плата розширення може переключитися на відповідний джерело живлення або задіяти перетворювачі рівнів, що дозволить їй працювати як з 5В, так і з 3.3В-пристроями[8].

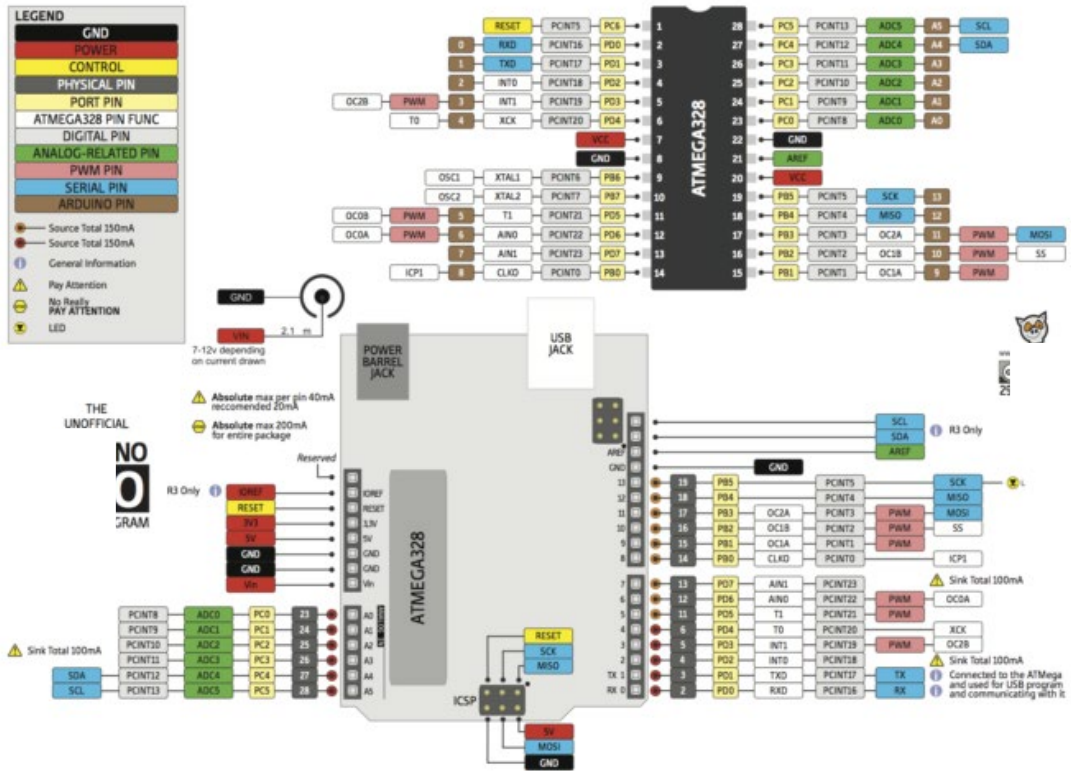


Рис. 1.7 Розпіновка плати Arduino Uno R3 на базі ATMEGA 328.

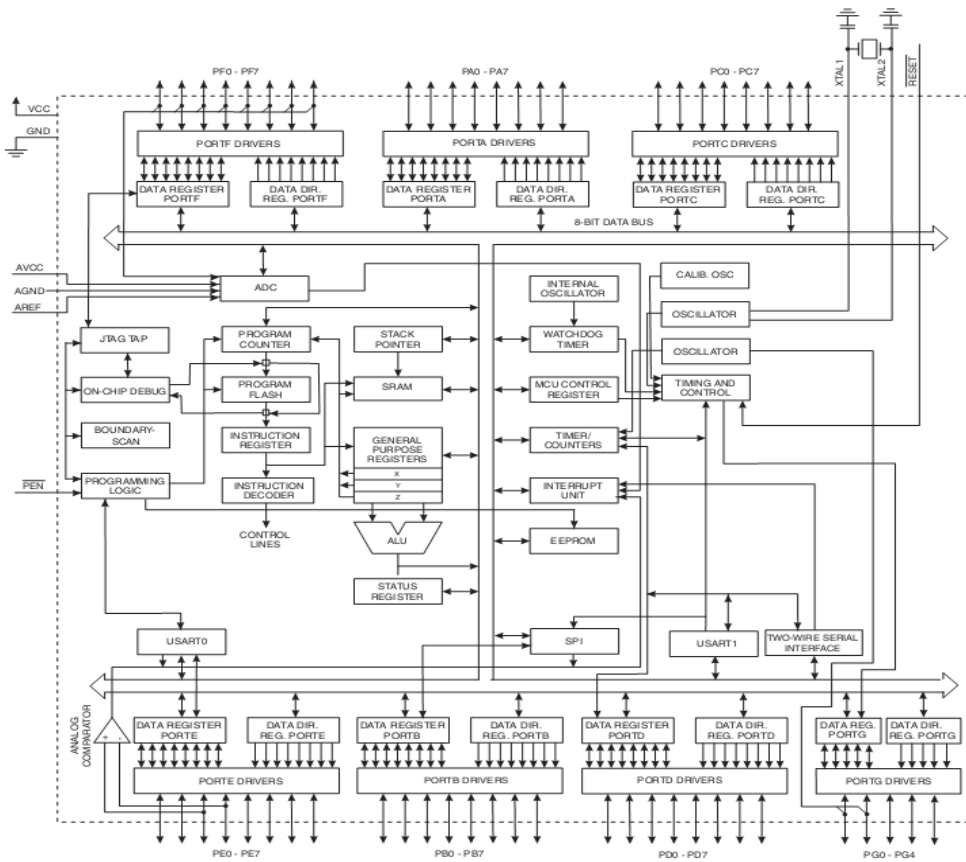


Рис. 1.8 Структурна схема Arduino Uno R3.

В Arduino Uno є 6 аналогових входів (A0 - A5), кожен з яких може уявити аналогову напругу у вигляді 10-бітного числа (1024 різних значення). За замовчуванням, вимір напруги здійснюється щодо діапазону від 0 до 5 В. Проте, верхню межу цього діапазону можна змінити, використовуючи вихід AREF і функцію `analogReference()`. Крім цього, деякі з аналогових входів мають додаткові функції:

Arduino Uno надає ряд можливостей для здійснення зв'язку з комп'ютером, ще одним Ардуіно або іншими мікроконтроллерами. У ATmega328 є приймач UART, що дозволяє здійснювати послідовну зв'язок за допомогою цифрових висновків 0 (RX) і 1 (TX). Мікроконтролер ATmega16U2 на платі забезпечує зв'язок цього приймача з USB-портом комп'ютера, і при підключенні до ПК дозволяє Ардуіно визначатися як віртуальний COM-порт. Прошивка мікросхеми 16U2 використовує стандартні драйвера USB-COM, тому установка зовнішніх драйверів не потрібно.

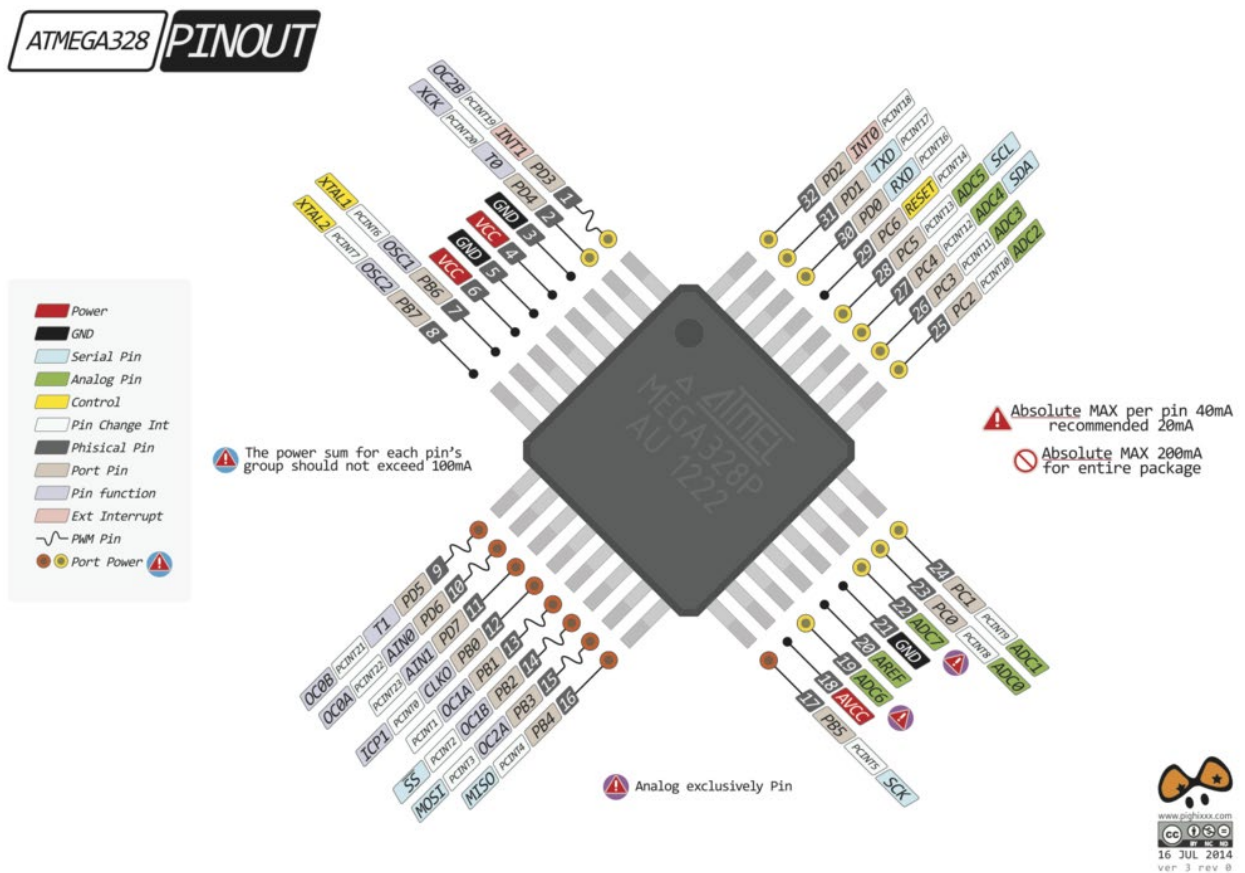


Рис. 1.9 Розпіновка мікропроцесора ATMEGA 328.

ATmega-328 в основному є мікроконтролером Advanced Virtual RISC (AVR). Він підтримує дані до восьми (8) бітів. ATmega-328 має вбудовану пам'ять 32 КБ. Цей мікроконтролер має масу інших характеристик. Ви також повинні поглянути на Вступ до PIC16F877a (це мікроконтролер PIC), а потім порівняти функції цих двох мікроконтролерів[9].

ATmega 328 має 1 КБ програмованої для читання пам'яті (EEPROM) з електричним стиранням. Ця властивість показує, якщо електропостачання, що подається на мікроконтролер, відключено, навіть тоді він може зберігати дані та може надавати результати після надання йому електропостачання. Більше того, ATmega-328 має 2 КБ статичної оперативної пам'яті (SRAM). ATmega 328 має кілька різних функцій, що робить його найпопулярнішим пристроєм на сучасному ринку. Ці функції складаються з вдосконаленої архітектури RISC, хорошої продуктивності, низького енергоспоживання, лічильника реального таймера з окремим генератором, 6 ШІМ- контактів, програмованого послідовного USART, блокування програмування для забезпечення програмного забезпечення, пропускної здатності до 20 MIPS тощо. ATmega-328 в основному використовується в Arduino[9].

### **1.6 Класифікація далекомірних систем за типом вимірювання.**

Далекомір - це вимірювальний пристрій, що служить для точного визначення відстані від спостерігача до об'єкта вимірювання. Далекіми по типу дії діляться на пасивні і активні, а ці два типи, в свою чергу діляться на наступні види.

У наш розвинений час, у величезному потоці інформації сучасній людині часто важко встежити за всіма нововведеннями і винаходами, тому про деякі корисні приладах багато людей навіть і не чули, не кажучи вже про володіння ними. Наприклад, тривалий час, щоб виміряти відстань до потрібної мети, застосовувалися дальномірні шкали на біноклях і оптичні приціли. Такий спосіб

був далеко не ідеальний, при своїй простоті, він давав не надійні і не точні відомості, до того ж спостерігач повинен був знати точні розміри орієнтира, що, наприклад в чистому полі, було далеко не легким умовою. Виходом з таких ситуацій став далекомір, портативний, точний і надзвичайно простий в управлінні прилад.

Відмінність пасивного далекоміра від активного полягає в принципі його роботи. Робота далекоміра це вимір часу, який витрачає відправлений далекоміром ультразвуковий сигнал для проходження дистанції до об'єкту і назад. Швидкість поширення сигналів вважається відомою. Вимірювання відстаней дальномерами пасивного типу ґрунтується на визначенні висоти  $h$  рівнобедреного трикутника  $ABC$ , наприклад по відомій стороні  $AB = l$  (базі) і протилежного гострого кута  $b$ , так званому параллактическому кутку. При малих кутах  $b$  (виражених в радіанах)  $h = l / b$ . Одна з величин,  $l$  або  $b$ , зазвичай є постійною, а інша - змінною (ізмeряеммой)., Особливо широке застосування прилад знайшов в інженерній геодезії, далекомір не замінний помічник при будівництві шляхів сполучення, гідротехнічних споруджено, ліній електропередач, в військовій справі він незамінний для визначення відстані до цілей, в фотографії для точної і швидкої фокусування, в туризмі, в сільському господарстві, в системах бомбометання, в навігації, в астрономічних дослідженнях, у мисливсько-стрілецьких цілях і звичайно ж далекомір займає окрему нішу в будівельній діяльності.

Але, мабуть, найсуттєвіший недолік ультразвукового далекоміра криється в самій природі звуку, а точніше в особливостях поширення звукової хвилі. Перше, що слід знати про звуки, це те, що він має хвильову природу. Щоб проілюструвати, як саме хвильова природа звуку позначається на точності вимірювання ультразвукового далекоміра, давайте повернемося в минуле - в дитинство. Всі діти люблять кидати камені в воду і спостерігати за тим, як від місця падіння каменя по воді поширюються хвилі. Чим далі від епіцентру відходять хвилі, тим вони стають ширшими. Звук, це теж хвиля. І в міру

віддалення звукової хвилі від її джерела вона також розширюється. Сучасні ультразвукові далекоміри оснащені лазерними покажчиками. І багато думаю, що ультразвукова хвиля відбивається саме в тій точці, куди спрямований промінь лазера. Але це не так. Якщо на шляху хвилі випадково виявиться перешкода (наприклад меблі) то звук відбитий від неї може досягти приймача раніше звуку відбитого наприклад стіни. З цієї ж причини важко проводити виміри ультразвуковими дальномерами кутів. Також важко вимірювати діагоналі.

- Лазерний далекомір це прилад, що вимірює відстані використовуючи лазерний промінь. Застосовується в інженерній геодезії, при топографічній зйомці, в військовій справі, в навігації, в астрономічних дослідженнях, в фотографії.

- Світлодалекомір або світловіддалемір, прилад для виміру відстаней за принципом дії світлодалекоміра ґрунтується на вимірюванні часу проходження світла від світлодалекоміра до відбивача або предмету і назад.

### **1.7. Загальні відомості датчика відстані HC-SR04.**

Ультразвуковий датчик HC-SR04 включає передавач і приймач. Цей датчик використовується для з'ясування відстані від об'єкта. Тут кількість часу, необхідного для передачі та прийому хвиль, визначатиме відстань між датчиком та об'єктом. Цей датчик використовує звукові хвилі за допомогою безконтактної технології. За допомогою цього датчика можна виміряти відстань, необхідну для цілі, без пошкоджень і забезпечити точні деталі. Діапазон датчика доступний від 2 см до 400 см.

HC-SR04 - це тип ультразвукового датчика, який використовує ехолот, щоб визначити відстань об'єкта від датчика. Він забезпечує чудовий діапазон безконтактного виявлення з високою точністю та стабільними показаннями. Він включає два модулі, такі як ультразвуковий передавач і приймач. Цей датчик використовується в різних додатках, таких як вимірювання напрямку та



швидкості, охоронна сигналізація, медична допомога, ехолоти, зволожувачі повітря, бездротова зарядка, неруйнівний контроль та ультрасонографія. Характеристики датчика HC SR04:

- Джерело живлення цього датчика становить + 5 В постійного струму
- Розміри - 45 мм x 20 мм x 15 мм
- Струм спокою, який використовується для датчика, становить <math><2\text{ mA}</math>
- Ширина вхідного імпульсу тригера становить 10 мкс
- Робочий струм становить 15 мА
- Кут вимірювання становить 30 градусів
- Відстань від 2 см до 800 см
- Роздільна здатність 0,3 см
- Ефективний кут <math><15^\circ</math>
- Діапазон робочих частот становить 40 Гц
- Точність 3 мм



Рис. 1.10 Загальний вигляд модуля HC-SR04

## Принцип роботи датчика відстані HC-SR04:

Вимірювання точної відстані можна досягти шляхом взаємодії датчика HC-SR04 з різними типами плат Arduino .

Спочатку подайте джерело живлення на датчик, щоб він увімкнувся, і підключіть контакт GND цього датчика до контакту GND плати Arduino. І модуль датчика може житися від напруги живлення плати Arduino, коли струм, який проходить через датчик, нижче 15 мА. Тож номінальні струми Arduino не вплинуть на датчик.

Після встановлення первинного пристрою підключіть обидва висновки датчиків, таких як Trig & Echo, до входних / вихідних висновків плати Arduino. Як ми вже обговорювали раніше, для початку методу вимірювання штифт Trig в датчику повинен утримуватися 10 $\mu$ s на початку. Отже, цей сенсорний модуль генеруватиме звукові хвилі частотою 40000 Гц приблизно кожену секунду від джерела.

Коли звукові хвилі повертаються назад, штифт Echo активуватиметься, доки приймач не отримає ці хвилі. Час буде вимірюватися за допомогою дошки Arduino.



Рис. 1.11 Розпіновка HC-SR04.

Коли джерело живлення подається на цей модуль, він генерує звукові хвилі, які рухаються по всьому повітрю, щоб потрапити на необхідний предмет. Ці хвилі б'ються і повертаються від об'єкта, а потім збираються модулем приймача.

Тут і відстань, і час, який ми взяли, прямо пропорційні, оскільки час, необхідний для більшої відстані, великий. Якщо тригерний штифт підтримувати високим протягом 10 мкс, тоді утворюватимуться ультразвукові хвилі, які рухатимуться зі швидкістю звуку. Таким чином, він створює вісім циклів звукового сплеску, які будуть зібрані в штифті Echo. Цей ультразвуковий датчик з'єднаний з Arduino для вимірювання необхідної відстані між датчиком та об'єктом. Відстань можна розрахувати за такою формулою.

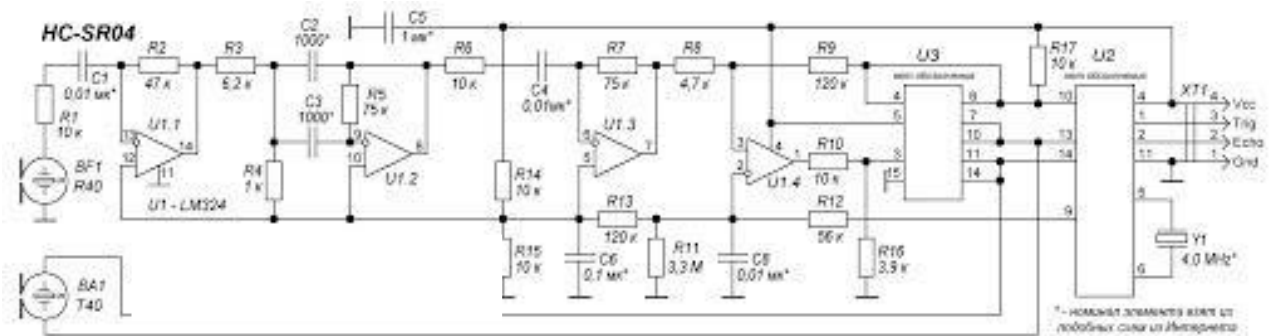


Рис. 1.12 Принципова електрична схема HC-SR04.

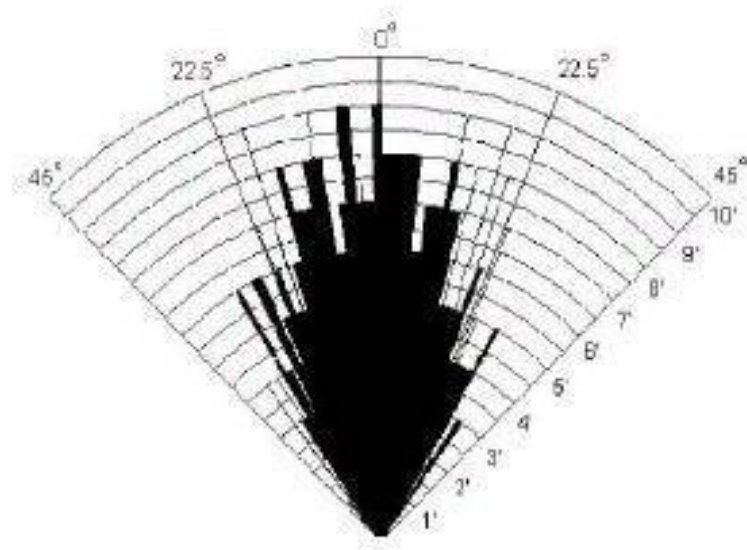


Рис. 1.13 Діаграма направленості HC SR04.

Також для підвищення точності треба правильно направити датчик: зробити так, щоб предмет був в рамках конуса діаграми направленості. Простіше кажучи, «очі» HC-SR04 повинні дивитися прямо на предмет. Діаграма спрямованості HC-SR04

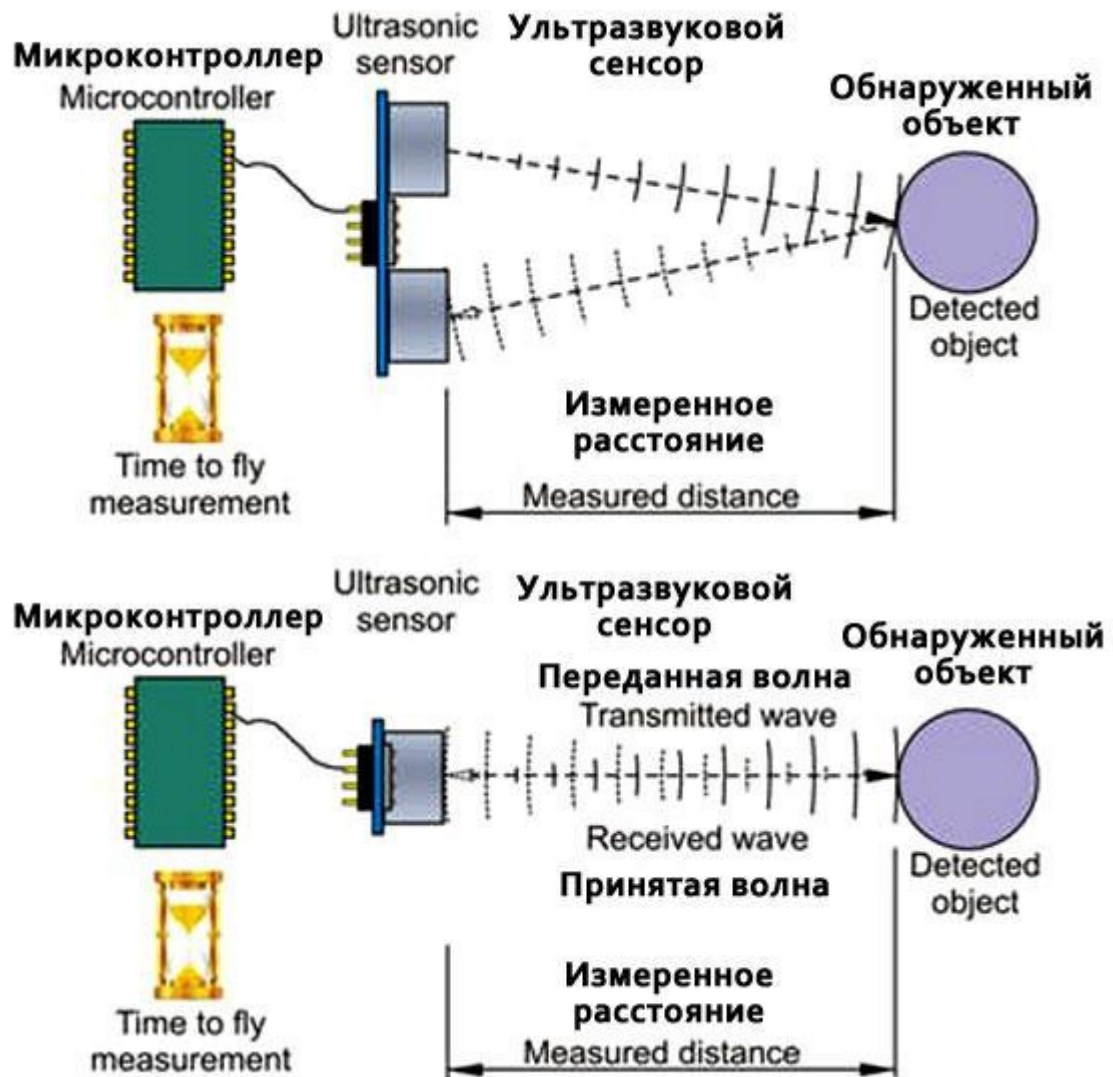


Рис. 1.14 Схемароботи HC-SR04.

Для зменшення помилок і похибки вимірювань зазвичай виконуються наступні дії:

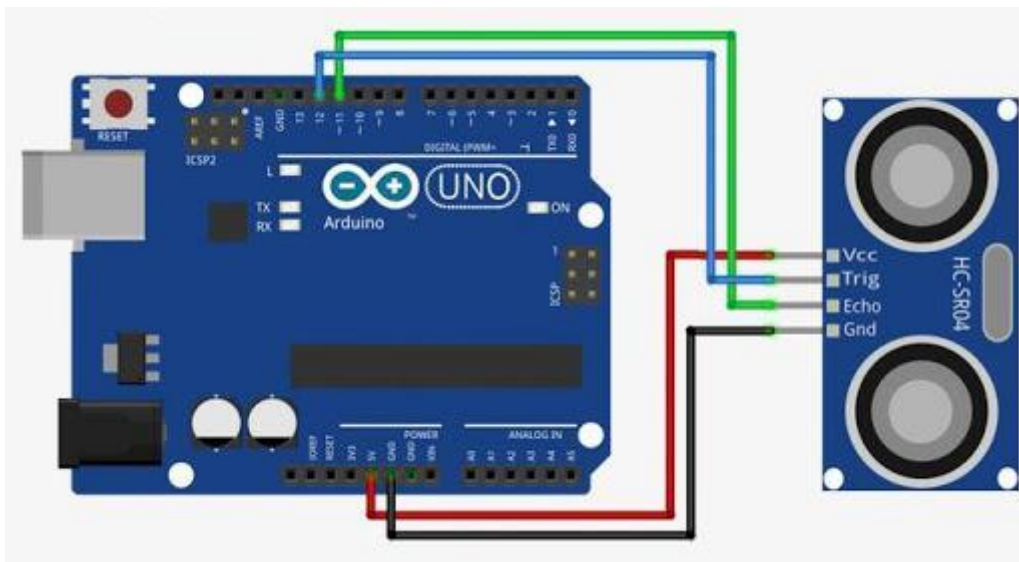
- усереднюються значення (кілька разів заміряємо, прибираємо сплески, потім знаходимо середнє);

- за допомогою датчиків (наприклад, DHT11 або DHT22) визначається температура і вносяться поправочні коефіцієнти;

- датчик встановлюється на серводвигун, за допомогою якого ми «повертаємо голову», переміщаючи діаграму спрямованості вліво або вправо.

Ультразвуковий датчик HC-SR04 постачається з чотирма контактами, а саме штифтом Vcc, тригером, ехо-штифтом і штифтом заземлення. Цей датчик використовується для вимірювання точної відстані між ціллю та датчиком. Цей датчик в основному працює на звукових хвилях. Пристрої, які потрібні для підключення:

- ультразвуковий далекомір HC-SR04
- Arduino Uno (Mega, Leonardo, Nano, тощо)
- з'єднувальні дроти (4 шт.)



*Рис. 1.15 Підключення модуля HC SR04 до Arduino.*

### **1.8. Семисегментний індикатор 4-розрядний з точками.**

Семисегментний індикатор - багаторозрядний семисегментний індикатор на базі мікросхеми MAX - кількість розрядів це і є кількість цифр, що може відобразити модуль.

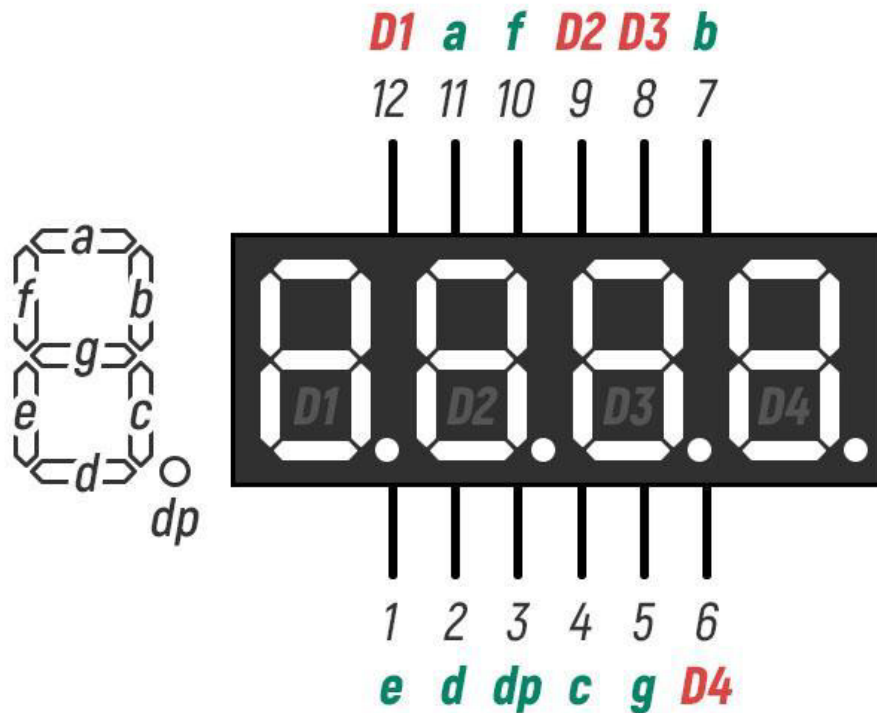


Рис. 1.16 Розпіновка чотирьох розрядного семисегментного індикатора з точками на мікросхемі MAX.

MAX - мікросхема є драйвером для світлодіодних семисегментних індикаторів, а також LED матриць 8x8, і ми не будемо розглядати принципові схеми підключення цього драйвера. Просто за основу взято готовий модуль, та наведено приклад підключення до плати Arduino UNO.

Отже багаторозрядні було дано хороший опис, семисегментний- тому що для формування символу або відображення цифри використовується сім світлодіодів, проіндексованих буквами A, B, C, D, E, F, DP, рисунок нижче показує як це позначено.

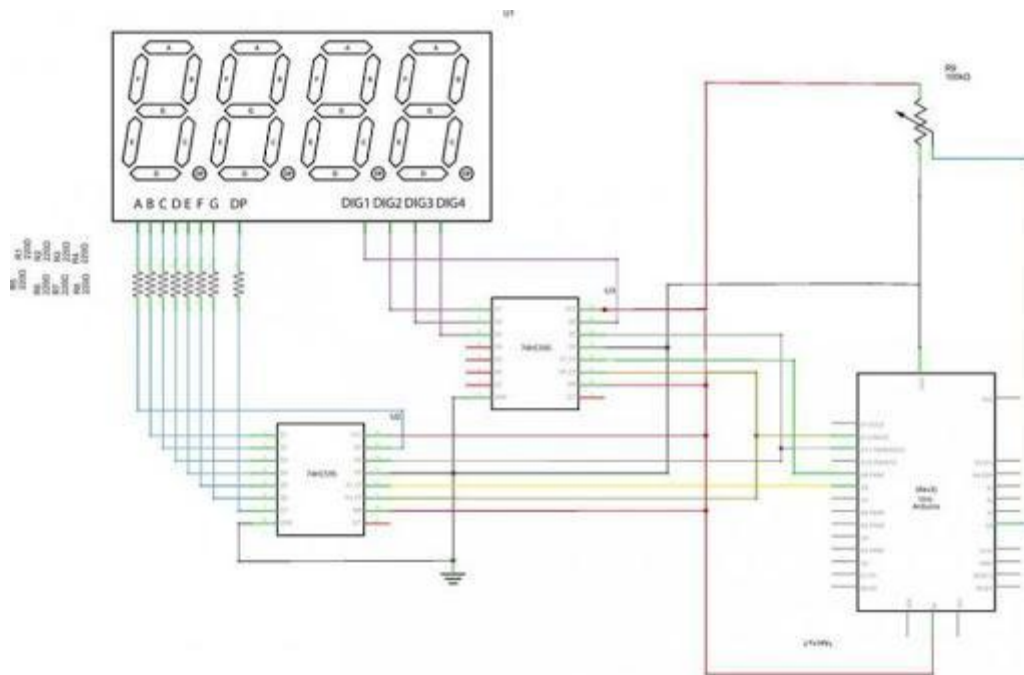


Рис. 1.17 Принципова схема чотирьох розрядного семисегментного індикатора з точками на мікросхемі MAX.

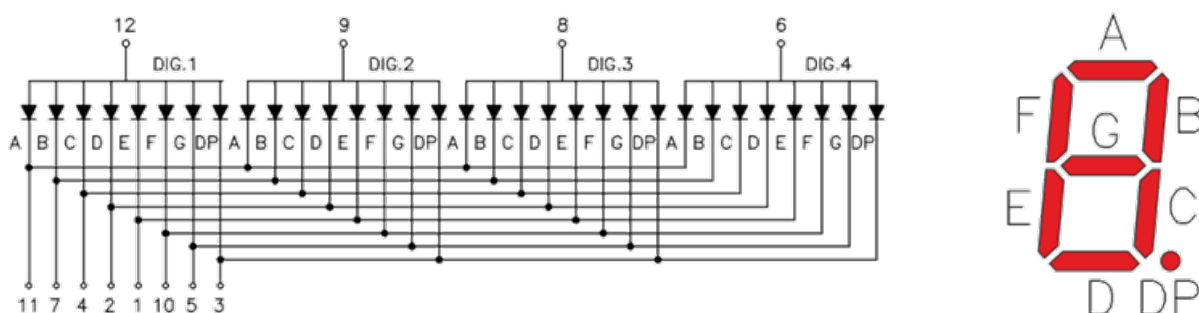


Рис. 1.18 Принципова схема та нумерація контактів одного розряду..

Як видно на рисунку, є також і восьмий світлодіод - DP. Повністю закодувати символ або цифру можна в 1 байті, встановлюючи або скидаючи певний біт, як це показано з прикладом кодування символу J.

Для відображення символів використовується більше функцій, підключається бібліотеки LedControl.

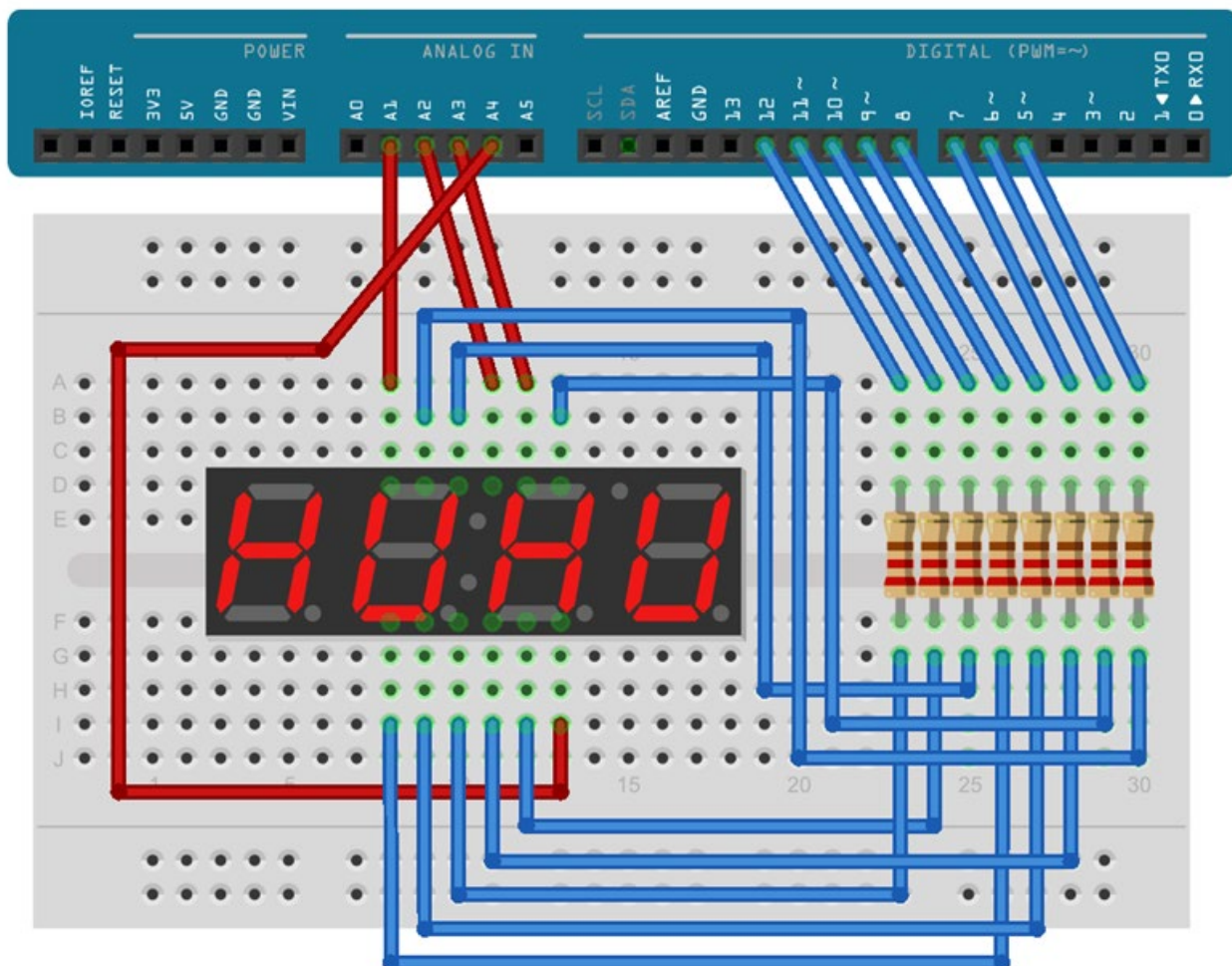


Рис. 1.19 Схеми підключення матриці сегментних індикаторів.

## 1.9. Висновки

В даному розділі була розглянута мікроконтролерна частина дипломного проекту. А саме була платформа, що служать основою для дальноміру. Також були розглянуті периферійні пристрої для використання їх у практичній частині роботи.



## РОЗДІЛ 2.

### ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ МОДЕЛІ ДАЛЕКОМІРУ

#### 2.1. Arduino IDE.

Arduino IDE (інтегроване середовище розробки) - це програмне забезпечення для Arduino. Це текстовий редактор, як блокнот з різними функціями. Він використовується для написання коду, складання коду для перевірки наявності помилок та завантаження коду в Arduino.

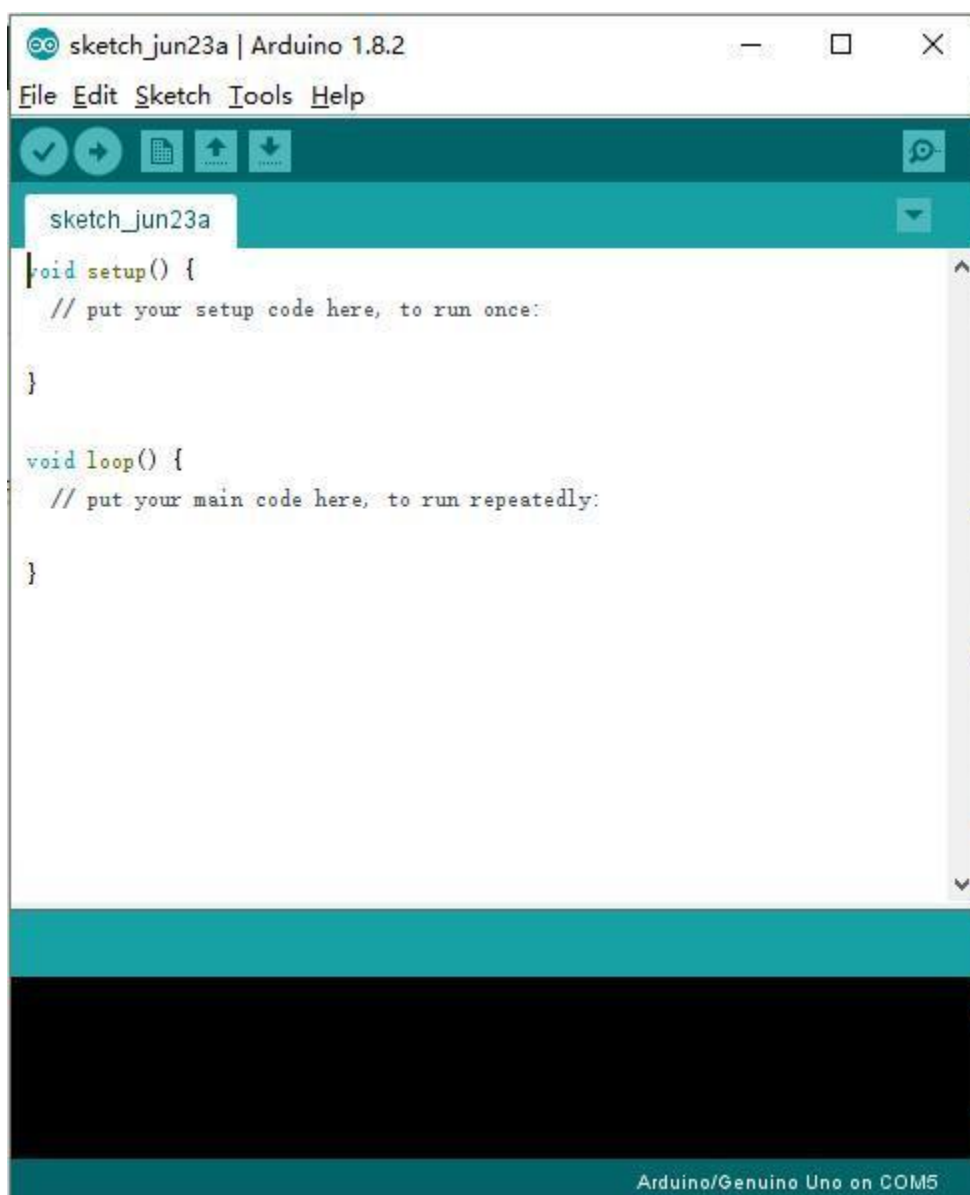


Рис. 2.20 Вікно програми.

Це крос-платформне програмне забезпечення, яке доступне для будь-якої операційної системи, як-от Windows, Linux, macOS.

Він підтримує мову C / C ++

Це програмне забезпечення з відкритим кодом, де користувач може використовувати програмне забезпечення так, як хоче. Вони також можуть створювати власні модулі / функції та додавати їх до програмного забезпечення

Він підтримує всі доступні плати Arduino, включаючи Arduino mega, Arduino Leonardo, Arduino Ethernet та багато іншого

Файл Word називається Документом так само, файл Arduino називається Ескізом, де користувач пише код.

Формат Arduino зберігається як .ino

Коли користувач пише код і компілює, IDE генерує шістнадцятковий файл для коду. (Шістнадцятковий файл - це десяткові файли Неха, які розуміє Arduino), а потім надсилаються на плату за допомогою кабелю USB. Кожна плата Arduino інтегрована з мікроконтролером, мікроконтролер отримає шістнадцятковий файл і працює відповідно до написаного коду.

Функції Arduino IDE:

IDE Arduino складається з різних розділів

1. WindowBar
2. MenuBar
3. Клавіші швидкого доступу
4. Текстовий редактор
5. Панель виводу

Віконна панель:

Рядок вікна містить назву Файл та версію програмного забезпечення Arduino IDE

Рядок меню:

Рядок меню складається з

- Файл
- Редагувати
- Ескіз
- Інструменти
- Допомога

Файл:

- **Новий**  
Створює новий файл. (**Ctrl + N**)
- **Відкрити**  
Використовується для відкриття файлу, який було збережено раніше. (**Ctrl + O**)
- **Open Recent (Відкрити останні)**  
Показує список нещодавно відкритих програм.
- **Sketchbook**  
Показує поточні ескізи, які ви використовували для свого проекту
- **Приклади**  
Приклади кількох основних проблем для довідки.
- **Закрити**  
Закриває головне вікно екрана. (**Ctrl + W**)
- **Зберегти**  
Використовується для збереження поточного ескізу. (**Ctrl + S**)
- **Зберегти як ...**  
Дозволяє зберегти поточний ескіз з іншою назвою. (**Ctrl + Shift + S**)
- **Налаштування сторінки Налаштування**  
сторінки для зміни сторінки (Текст). (**Ctrl + Shift + P**)
- **Друк**  
Використовується для друку поточної програми. (**Ctrl + P**)
- **Налаштування**  
Налаштування програмного забезпечення IDE можна змінити тут. (**Ctrl +,**)

- **Вийти**

Закриває всі вікна IDE. (Ctrl + Q)

**Редагувати :**

- **Скасувати / Повторити**

Повертається до одного або кількох кроків, зроблених вами під час редагування.

- **Вирізати**

Вирізає виділений текст із редактора.

- **Копіювати**

Копіює вибраний текст із редактора

- **Копіювати для форуму Копіює**

та змінює стиль коду, придатного для форуму.

- **Копіювати як HTML**

Він копіює та змінює стиль коду, придатного для HTML.

- **Вставити**

- Він вставляє текст із скопійованого тексту.

- **Виділити все**

Виберіть весь вміст із редактора.

- **Коментар / Розкоментация**

Використовується для коментування та коментування вибраних рядків коду.

- **Збільшити / зменшити відступ**

Додає або видаляє пробіл на початку кожного виділеного рядка

- **Знайти Знайде**

набраний текст у редакторі

- **Знайти далі Знайде**

наступну позицію шуканого слова.

- **Знайти попереднє Знайде**

попереднє положення шуканого слова.

## Ескіз

- **Verify / Compile**

Перевіряє або перевіряє вашу програму, якщо є помилки, і відображається на панелі виводу.

- **Upload**

It компілює, а також завантажує код на плату Arduino.

- **Завантажити за допомогою програміста**

Завантажити код за допомогою програміста, який доступний на вкладці Інструменти.

- **Експортувати скомпільований двійковий**

файл Зберігає файл .hex у системі

- **Показати папку ескізу**

Відкриває поточну папку ескізу.

- **Включити бібліотеку**

Додає бібліотеку до ескізу, вставляючи оператори `#include` на початку коду

- **Додати файл ...**

Додає файл до ескізу, і новий файл з'являється на новій вкладці у вікні.

## Інструменти

- **Автоматичне форматування**

Цей параметр форматує ваш код у приємний формат, щоб усі могли зрозуміти.

- **Архівний ескіз**

Копіює код у формат winrar (.zip)

- **Виправити кодування та перезавантаження** **Виправляє**

можливі розбіжності між кодуванням карт редактора символів та іншими картами символів операційних систем.

- **Послідовний монітор**

Послідовний монітор відображає візуальне спілкування шляхом

надсилання та отримання даних

- **Дошка**

Для вибору типу дошки Arduino

- **Порт**

Для вибору порту, куди ви підключили Arduino

- **Програміст**

Для вибору апаратного програміста під час програмування плати або мікросхеми, а не за допомогою типу зв'язку USB.

- **Запис завантажувача**

- Він використовується для запису завантажувача на плату Arduino

**Панель виводу:**

Ця панель виводу використовується для коментарів щодо коду

- якщо код успішно скомпільовано або виникає помилка.
- Якщо код було успішно завантажено на дошку.

Скільки місця зайняла дошка.

## 2.2. Основи скетчу Arduino

Змінна є місцем для зберігання порції даних. Він має ім'я, тип і значення. Наприклад, рядок із наведеного вище ескізу Blink декларує змінну з іменем `ledPin`, типом `int`та початковим значенням `13`. Він використовується для позначення, до якого штифта Arduino підключений світлодіод. Кожного разу, коли ім'я `ledPin` з'являється в коді, його значення отримуватиметься. У цьому випадку особа, яка пише програму, могла б вирішити не турбуватись створенням `ledPin` змінної, а натомість просто написати `13` скрізь, де їм потрібно було вказати PIN-код. Перевага використання змінної полягає в тому, що простіше перенести світлодіод на інший штифт: вам потрібно відредагувати лише один рядок, який присвоює змінну початкове значення.

Однак часто значення змінної змінюватиметься під час запуску ескізу.

Наприклад, ви можете зберегти значення, прочитане з введення, у змінну. У посібнику зі змінних є додаткова інформація .

Функція (інакше відома як процедури або підпрограма) є іменованих фрагментом коду, який може бути використаний в інших місцях в ескізі.

Перший рядок містить інформацію про функцію, як-от її назву, "налаштування". Текст до і після імені вказує тип і параметри повернення: це буде пояснено пізніше. Код між {} називається тілом функції: що робить функція.

Ви можете викликати функцію, яка вже була визначена (або у вашому ескізі, або як частина мови Arduino). Наприклад, рядок `pinMode(ledPin, OUTPUT);` викликає `pinMode()` функцію, передаючи їй два параметри : `ledPin` і `OUTPUT`. Ці параметри використовуються `pinMode()` функцією, щоб вирішити, який штифт і режим встановити.

`pinMode()` Функція налаштовує значення як вхід або вихід. Щоб використовувати його, ви передаєте йому номер налаштованого порту та константу `INPUT` або `OUTPUT`. Коли конфігурується як вхід, порт може виявляти стан датчика, як кнопка; це обговорюється в наступному підручнику . Як вихід, він може керувати виконавчим механізмом, як світлодіод.

`digitalWrite()` функції виводить значення на піну встановить `ledPin`(контакт 13) на ВИСОКИЙ, або 5 вольт. Написання НИЗЬКОГО штифта підключає його до землі, або 0 вольт.

`delay()` - викликає Arduino чекати задану кількість мілісекунд , перш ніж перейти до наступного рядка. У секунду 1000 мілісекунд, тому рядок: створює затримку в одну секунду.

`setup ()` і `loop ()` - є дві спеціальні функції, які є частиною кожного ескізу Arduino:

- `setup()` - викликається один раз, коли ескіз починається. Це гарне місце для виконання завдань налаштування, таких як встановлення режимів штифтів або ініціалізація бібліотек.

- `loop()` - функція викликається знову і знову і основа більшості ескізів. Вам потрібно включити обидві функції у свій ескіз, навіть якщо вони вам ні для чого не потрібні.

### 2.3. Структура програми в Arduino IDE

Структура програми в Arduino IDE може змінюватися програмістом в залежності від завдань і конкретних рішень, але основний її незмінний мінімальний скелет як на цьому прикладі обов'язково зберігається з двох функцій `setup` і `loop`. Без них програма не компілюється.

```
void setup() {  
  }  
void loop() {  
  }
```

Функція - це фрагмент програмного коду, який має унікальну назву і обмежений фігурними дужками, до якого можна звернутися з іншого місця програми.

`Setup` означає налаштування. Пряме призначення цієї функції полягає в тому, щоб програмісти в ній задавали початкові параметри функціоналу контролера і програмних об'єктів, що використовуються в інших частинах програми.

Слово `loop` означає повторюється цикл - зациклення виконання набору виразів, які перебувають в цій функції. Її призначення в тому, щоб зберігати вираження основного циклу програми контролера.

Після старту контролер приймається обробляти вираження, що знаходяться між фігурними дужками секції `setup`.

Після закінчення виконання всіх завдань першої функції, контролер береться за нескінченне послідовне виконання програмного коду секції `loop`, який за задумом повинен виконувати всю корисну роботу, закладену нами, в динамічному режимі.



Тепер розглянемо простий приклад програми з подібною структурою, із закладеною можливістю моніторингу її виконання в моніторі порту, а так само по індикації за допомогою світлодіода на платі контролера. На платі Arduino Nano і UNO є світлодіод, підключений до 13-му піну і ним ця програма може блимати.

```
int led_pin = 13;
void setup() {
  Serial.begin(9600);
  pinMode(led_pin, OUTPUT);
  Serial.println("setup");
}
void loop() {
  digitalWrite(led_pin, HIGH);
  delay(1000);
  digitalWrite(led_pin, LOW);
  delay(1000);
  Serial.println("loop");
}
```

Тут до розглянутої нами структури додався рядок коду, який знаходиться попереду і поза фігурних дужок будь-якої функції. У цьому місці прийнято розміщувати рядки підключення бібліотек, оголошення глобальних змінних, констант і примірників класів.

Отже це були уявлення про найпростіші складові проекту програми-скетчів для контролерів Arduino. Охарактеризували що вміст функції setup обробляється контролером один раз після запуску контролера, а вміст функції loop обробляється циклічно без зупинки. Так само, що скетч не компілюватиметься при відсутності цих функцій.

## 2.4. Висновки

У даному розділі розглянуті програмні засоби для створення системи ультразвукового далекоміру на основі мікроконтролерних систем. Також було розглянуто алгоритм роботи та структури програм, скетчів, які зашиваються до мікроконтролера.

## РОЗДІЛ 3. РЕАЛІЗАЦІЯ УЛЬТРАЗВУКОВОГО ДАЛЕКОМІРУ НА МІКРОКОНТРОЛЛЕРІ

### 3.1. Блок-схема алгоритму роботи ультразвукового далекоміру.

Блок-схема - представлення алгоритму розв'язування або аналізу задачі за допомогою геометричних елементів (блоків), які позначають операції, потік, дані тощо. Блок-схема для нашого макету буде мати наступний вигляд:

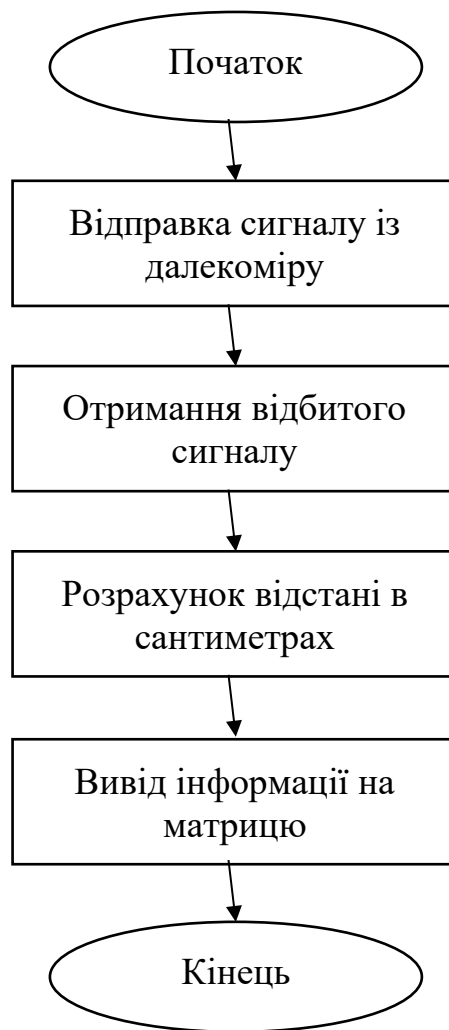


Рис. 3.21 Блок-схема роботи програми для ультразвукового далекоміру.

### 3.2. Проектування функціональної, принципової та структурної схем ультразвукового далекоміру на мікроконтролері.

Функціональна схема містить інформацію про способи реалізації пристроєм заданих функцій[11]. За такою схемою можна визначити, як здійснюються перетворення і які для цього необхідні функціональні елементи. Функціональна схема— схема, що роз'яснює певні процеси, що відбуваються у певних функціональних ланцюгах виробу (устаткування) чи у виробі (устаткуванні) в цілому. На основі функціональної та структурної схем розробляється принципова схема. Такими схемами користуються для вивчення принципу роботи виробів(устаткуванні), а також при їх налагодженні контролі чи устаткуванні[12].

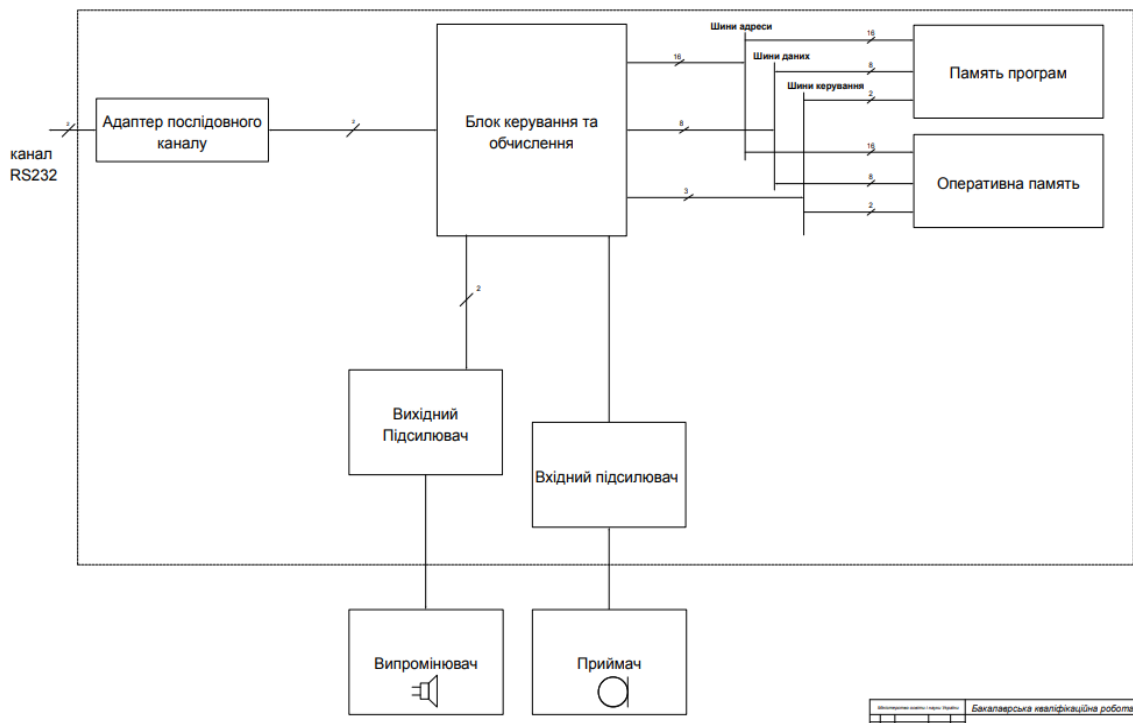


Рис. 3.22 Функціональна схема ультразвукового далекоміру на мікроконтролері..

Електрична принципова схема (ГОСТ 2.702-75) є видом електричної схеми виробу, що дає найповніше уявлення про склад і принцип його роботи[13]. При виконанні схем цифрової обчислювальної техніки керуються ГОСТ 2.708-81.

Цей вид кресленика не враховує габаритних розмірів і реального розташування деталей об'єкта. За рівнем абстракції принципові електричні схеми займають середню позицію між функціональними і монтажними схемами. На принциповій електричній схемі (ГОСТ 2.709-89) зображуються всі складові частини виробу і зв'язки між ними, та елементи, якими закінчуються вхідні та вихідні ланки електричних кіл(роз'єми, затискачі тощо)[13].

Принципові електричні схеми призначені для повного відображення взаємозв'язків пристроїв з урахуванням принципів їх дії і послідовності роботи. На принципових електричних схемах електричні елементи зображують за допомогою умовних позначень, а також вказують лінії зв'язків між ними, блоками та модулями. На схемі, також, розміщується наступна інформація: умовне зображення принципу роботи функціональних вузлів, пояснювальні написи, частини окремих елементів, діаграми переключення контактів, а також перелік використовуваних в даній схемі пристроїв[14].

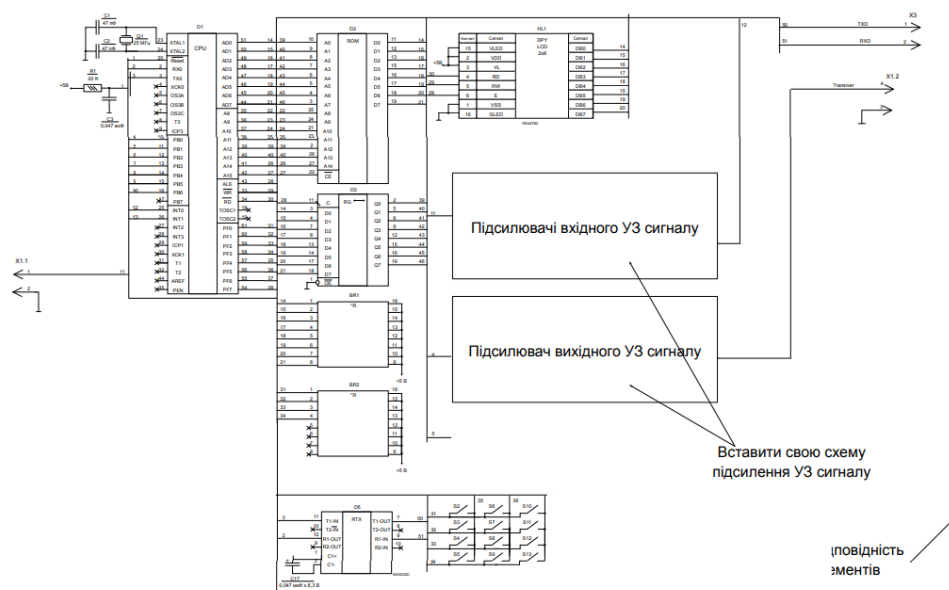


Рис. 3.23 Електрична схема ультразвукового далекоміру на мікроконтролері.

Структурна схема — схема, яка визначає основні функціональні частини виробу, їх взаємозв'язки та призначення. Під функціональною частиною розуміють складову частину схеми: елемент, пристрій, функціональну групу, функціональну ланку[15].

Структурна схема призначена для відображення загальної структури пристрою, тобто його основних блоків, вузлів, частин та головних зв'язків між ними. Із структурної схеми повинно бути зрозуміло, навіщо потрібний даний пристрій і як він працює в основних режимах роботи, як взаємодіють його частини. Позначення елементів структурної схеми можуть обиратись довільно, хоча загальноприйнятих правил виконання схем слід дотримуватись[16].

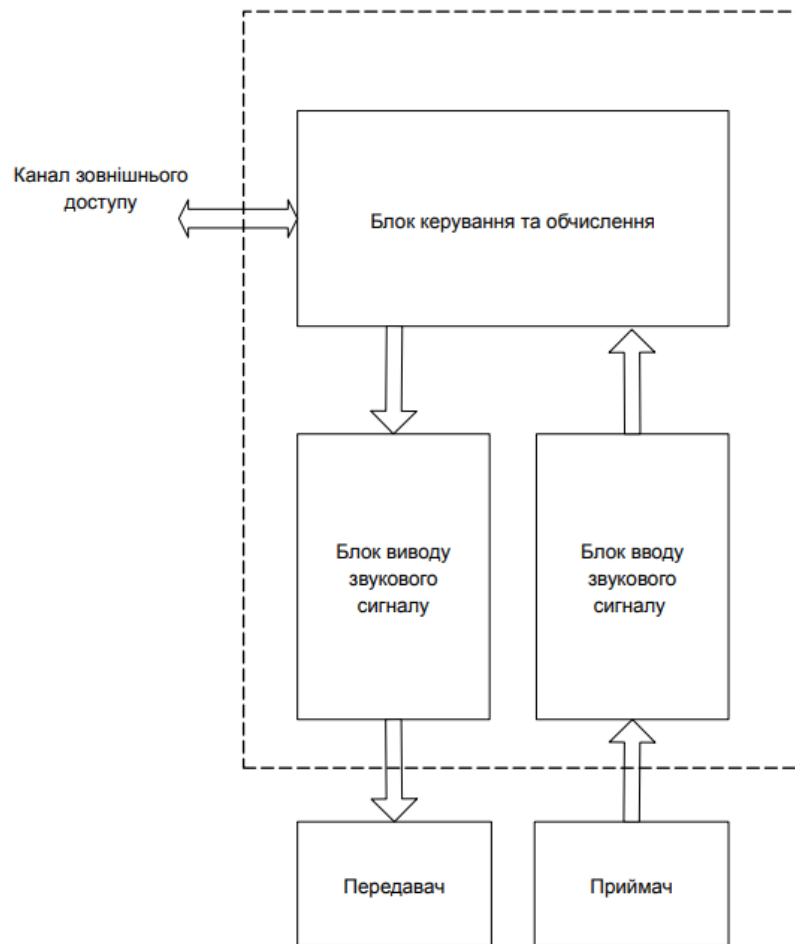


Рис. 3.24 Структурна схема ультразвукового далекоміру на мікроконтролері.

### 3.3. Написання коду скетчу для ультразвукового далекоміра.

Для початку надамо імена пінам для зручності використання, для ультразвукового далекоміру використаємо піни 1 та 2 та надамо їм наступні імена, також додамо змінну в якій будемо зберігати замірену відстань:

```
#define Trig 1
#define Echo2
unsigned int distance =0;
```

Встановимо в функції void setup() пін Trig як вихід, а Echo як вхід:

```
void setup()
{
  pinMode(Trig, OUTPUT);
  pinMode(EchoR, INPUT);
}
```

До функції void loop() додамо код для вимірювання відстані:

```
void loop()
{
  digitalWrite(Trig, HIGH); // Включаємо
  delayMicroseconds(10); // Затримка рівна 10 мікросекундам
  digitalWrite(Trig, LOW); // Включаємо
  impulseTime=pulseIn(Echo, HIGH); // Міряємо довжину імпульсу
  distance =impulseTime/58; // Переводимо в сантиметри
}
```

### **3.4. Написання коду скетчу для чотирьох секційного семисегментного індикатора.**

Для початку потрібно задати піни для кожного розряду та сегменту

```
int anodPins[] = {A1, A2, A3, A4}; // Піни розряду
int segmentsPins[] = {5, 6, 7, 8, 9, 10, 11, 12}; //Піни для сегмента (7 + 1(точка))
```

За допомогою циклу щоб не повторятись встановимо піни на вивід:

```
void setup() {
  for (int i = 0; i < 4; i++) pinMode(anodPins[i], OUTPUT);
  for (int i = 0; i < 8; i++) pinMode(segmentsPins[i], OUTPUT);
}
```

Для цифри описуються кожному сегменті, тому для зручності запишемо усі цифри в масив:

```
int seg[12][8] = {
  {1, 1, 1, 1, 1, 1, 0, 0}, //Цифра 0
```

```

    {0, 1, 1, 0, 0, 0, 0, 0}, //Цифра 1
};

void loop() {
    int digits = 4; // В нас 4 розряди

    // Формуємо дані для відображення:
    int dig = -1;
    for (int i = 0; i < output.length(); i++) {
        String _char = output.substring(i, i + 1);

        if (_char != ".") dig += 1;

        int actualdigit = 11; //За замовчуванням пустий символ
        if ((_char == "-")) {
            actualdigit = 10;
        }
        else if (_char == " " || _char == ".") {
        }
        else {
            actualdigit = _char.toInt();
        }

        if (_char == ".") {
            arr[dig][7] = 1; // Якщо потрібно ставимо крапку
        }
        else {
            for (int n = 0; n <= 7; n++) {
                arr[dig][n] = seg[actualdigit][n];
            }
        }
    }
}

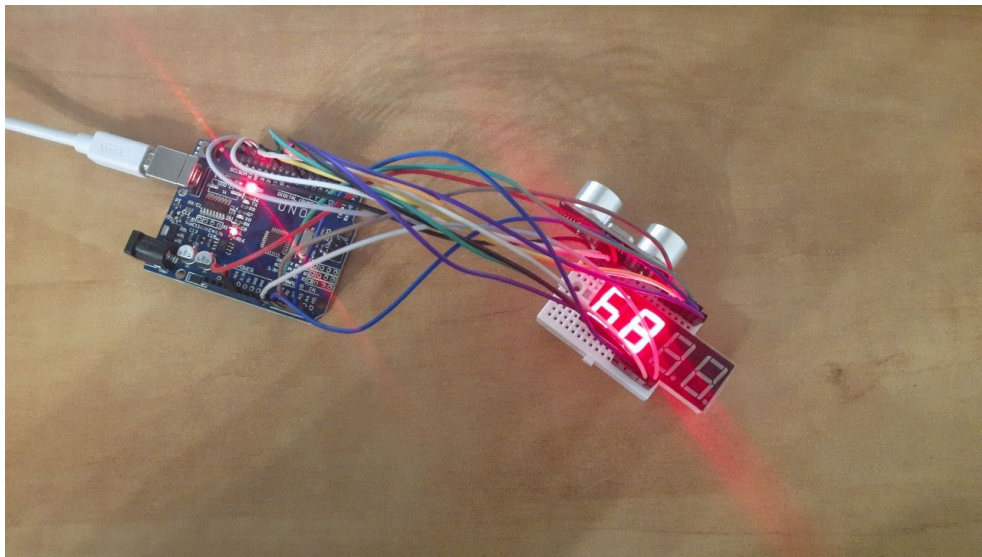
for (int i = 0; i < 4; i++) { // Кожен розряд по черзі
    for (int k = 0; k < 8; k++) { // Кожен сегмент по черзі
        digitalWrite(segmentsPins[k], ((dig[i][k] == 1) ? LOW : HIGH));
    }
    digitalWrite(anodPins[i], HIGH);
    delay(1);
    digitalWrite(anodPins[i], LOW);
}
}

```

### 3.5 Тестування та результати роботи макету

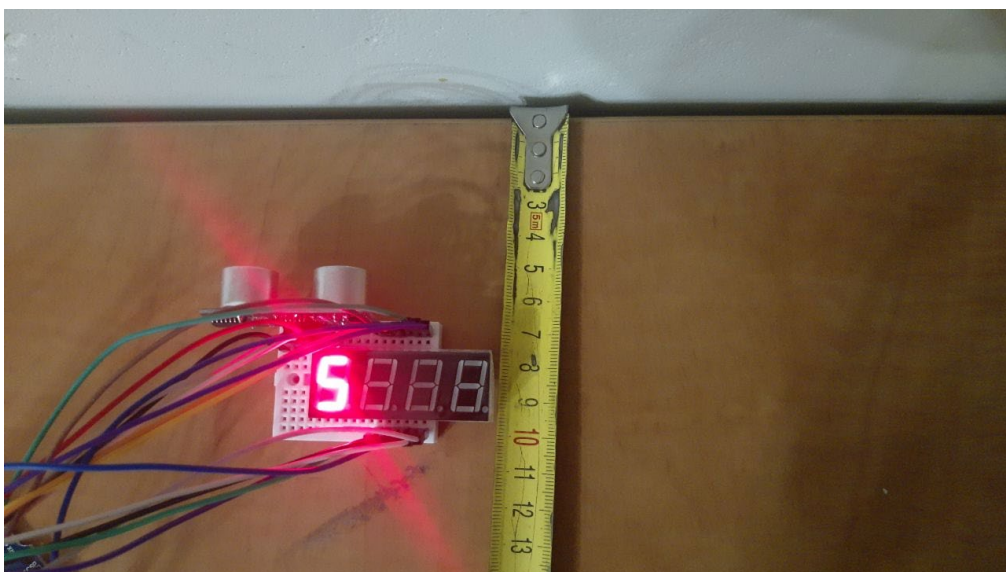
В ході даного курсового проекту був розроблений і зібраний на макетній платі ультразвукової дальномір, заснований на ультразвуковому датчику HC-SR04 і Arduino UNO.

Показання датчика відображаються на чотирьохрозрядному семисегментний індикаторі на чіпі TM1637



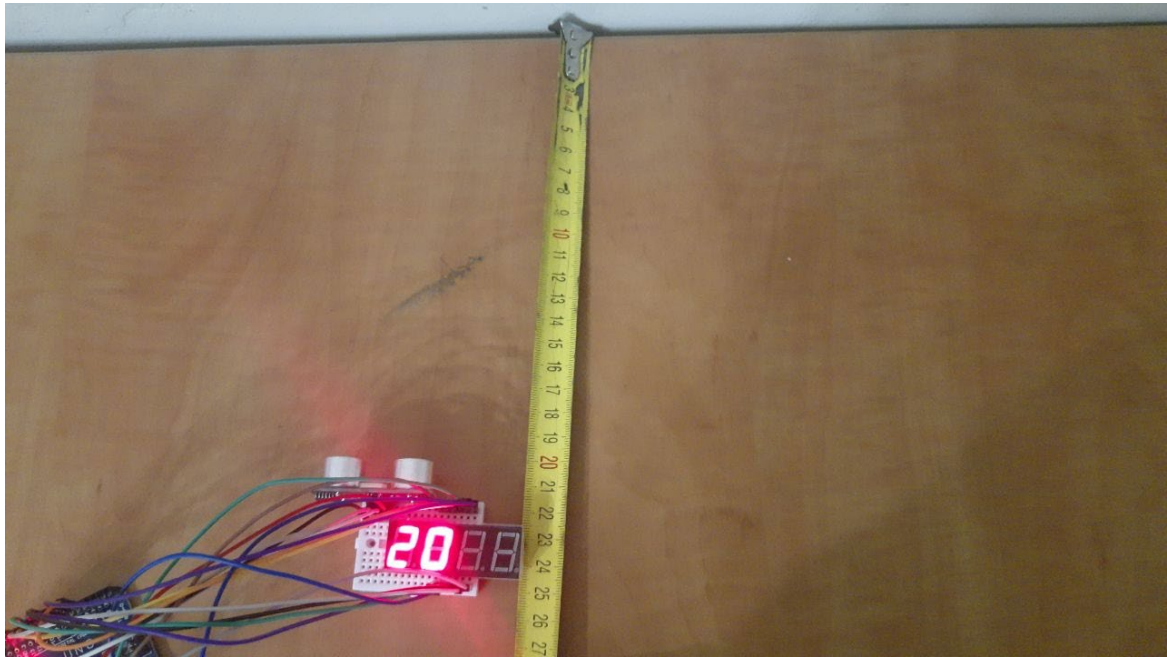
*Рис. 3.25 Макет ультразвукового дальноміру.*

Для тестування макету перед випромінювачем було встановлено предмет до якого потрібно виміряти відстань.



*Рис. 3.26 Результат роботи макету ультразвукового дальноміру на відстані 5 сантиметрів.*





*Рис. 3.27 Результат роботи макету ультразвукового далекоміру на відстані 20 сантиметрів.*

Також враховано можливість видання помилки так як діапазон роботи самого датчика HC-SR04 обмежеі в діапазоні від 3 сантиметрів до 450, при виході за межі трапляється помилка в результаті чого буде показано наступне:



*Рис. 3.28 Помилка в діазоні вимірювання.*

### **3.6. Висновки**

В розділі представляються результати реалізації алгоритму розв'язання поставленої задачі на базі вибраних апаратних, апаратно-програмних засобів, програмного середовища. В результаті чого написано скетч який буде вимірювати відстань та показувати її.

## 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

### 4.1 Опис ідеї проекту

Проаналізовано та подано в вигляді таблиць:

- зміст ідеї (що пропонується);
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач товару (за кожним напрямком застосування);
- чим відрізняється від існуючих аналогів та заміників.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди користувача
Побудова ультра-звукового вимірювача відстані на базі мікроконтроллера	1. Використання вимірювання відстані між об'єктами.	Зручність на відстані декількох метрів
	2. Використання для вимірювання відстані до об'єкту.	Перевага що не потрібно йти до об'єкту вимірювання.
	3. Безконтактне вимірювання.	Нетривіально контактувати із об'єктом вимірювання.
	4. Вимірювання в русі.	Можливість вимірювати відстань до рухомих об'єктів.

Аналіз потенційних техніко-економічних переваг порівняно з пропозиціями конкурентів передбачає:

- визначення переліку техніко-економічних властивостей та характеристик;
- визначення попереднього кола конкурентів або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проведення збору інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів конкурентів;
- проведення порівняльного аналізу показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б)

аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні)  
(табл. 4.2).

Таблиця 4.2 – Визначення характеристик ідеї проекту

№ п / п	Техніко-економічні характеристики ідеї	Потенційні товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Запропонований метод	Загальновживаний метод			
1	Можливість використання різних варіантів зміру корпусу	Дає змогу	Не дає змогу	Проектування різних корпусів має вміщати в себе плату контролера	Можливість встановлення в іншій корпусі більші за розмірами	Рішення є більш дешевим відносно інших
2	Ремонтпридатність шляхом заміни елементів які вийшли із ладу	Дає змогу	Дає змогу	При виході із ладу якогось із елементів потрібна повна його заміна	Майже будьхто зможе самостійно замінити елементи які вийшли з ладу	Досить низька вартість елементів

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційної пропозиції, що є підґрунтям для формування його конкуренто спроможності.

#### 4.2 Технологічний аудит ідеї проекту

Було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту. Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (табл.4.3):

- за якою технологією буде виготовлено товар згідно ідеї проекту;
- чи існують такі технології, чи їх потрібно розробити/допрацювати;

– чи доступні такі технології авторам проекту.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Автономність	Акулумяторна батарея	Наявна	Доступна
2	Вимірювання в режимі реального часу	Семисегментний індикатор	Наявна	Доступна
3	Мобільність	Розміри корпусу	Наявна	Доступна
4	Взаємозамінність елементів	Відсутність пайки	Наявна	Доступна
5	Фіксація останніх замірів	Карта пам'яті	необхідно розробити	Доступна

### 4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів. Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 4.4).

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	1
2.	Динаміка ринку (якісна оцінка)	Зростає
3.	Наявність обмежень для входу (вказати характер обмежень)	Немає
4.	Специфічні вимоги до стандартизації та сертифікації	ГСТУ,СТТУ,ТУУ
5.	Середня норма рентабельності в галузі або по ринку,%	93%

Середня норма рентабельності в галузі (або по ринку) порівнюється із

банківським відсотком на вкладення.

Після визначення потенційних груп клієнтів проведений аналіз ринкового середовища: складені таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. №№ 4.6...4.7). Фактори в таблиці подані в порядку зменшення значущості.

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№ п / п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Зручність та можливість збільшення продуктивності	Люди що працюють на будівництвах, встановленні габаритного обладнання	Залежно від цільової групи експлуатаційні параметри товару не змінюються	<ul style="list-style-type: none"> <li>– надійність</li> <li>– мобільність</li> <li>– доступність</li> <li>– простота</li> <li>– зручність</li> <li>– швидкість</li> </ul>

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Незацікавленість клієнтів	Внаслідок невдалого маркетингу клієнт може не зацікавитись послугами	Зниження цін
2.	Потреба в ресурсах	Для створення більш продуктивного приладу необхідно розробити технічне обладнання та підготувати висококваліфіковані кадри	Укладання договорів з комерційними структурами для фінансування та надання можливостей для розробки, побудови і тестування приладу

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Конкуренція	Спонукає розробляти та виробляти нові продукти, знижувати витрати на їх виробництво	Ускладнення структури та функціональної наповненості пропозиції
2.	Попит	Існування стійкого попиту означає, що більшість клієнтів зацікавлені у введенні більш продуктивного приладу в експлуатацію	Рекламна діяльність, просування товару в Інтернеті

Надалі проведений аналіз пропозиції: визначені загальні риси конкуренції на ринку (табл. 4.8).

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
1. Вказати тип конкуренції – олігополія	На ринку присутня невелика кількість підприємств, що займаються побудовою такого типу систем	Підвищувати якість послуг за рахунок використання прогресивних технологій
2. Локальний	Відсутність єдиного національного постачальника послуг	Окремий підхід до кожної локальної ділянки
3. Внутрішньо-галузєва	Економічна боротьба між різними компаніями, які діють в одній галузі економіки, надають послуги і реалізують однакові товари, що задовольняють одну й ту саму потребу, але мають відмінності у виробничих затратах, якості, ціні тощо	Слідкувати за розробками конкурентів
4. Товарно-видова	Потужність наявних технічних засобів для створення та удосконаленого окремих елементів системи	За необхідності, використання обладнання схожого типу
5. Цінова	Можливість заощадити за допомогою більш дешевих елементів	Гнучка політика цін
6. За інтенсивністю – марочна	Кожна діагностика має бути стандартизованою	Реклама товару, проведення тендерів

Після аналізу конкуренції проведено більш детальний аналіз умов конкуренції в галузі (за моделлю 5 сил М. Портера) (табл. 4.9).

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Технологічні постачальники	Необхідність пошуку постачальників	Залучення малопопулярних	Незалежність у прийнятті	Надання переваги більш авторитетним

	ки	ників	постачальни-ків	клієнтських рішень)	технологічним рішенням
Висновки:	Інтенсивність незначна	Можливість виходу на ринку є	Постачальники диктують цінову політику на обладнання	Клієнти диктують вимоги до якості	Обмеження існують лише у разі відмови від діагностики

Для того щоб бути конкурентоспроможним на ринку для розробки товару, потрібно залучити висококваліфікованих спеціалістів у галузі проектування мікроконтролерних систем, маркетингу та аналітики.

На основі аналізу конкуренції, наведеного в табл. 4.9, а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (табл. №№ 4.6-4.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за табл. 4.10.

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Раціональніший ціновий показник	Можливість більш раціонально використати ресурсів
2	Доступність	Доступна всіх потребує
3	Спектр застосувань	Використання для ряду потреб користувачів.

За визначеними факторами конкурентоспроможності (табл. 4.10) проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін проекту

№	Фактор конкуренто-спроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні							
			-3	-2	-1	0	1	2	3	
1	Раціональніший ціновий показник	18		+						
2	Доступність	20	+							
3	Спектр застосувань	14	+							

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.11).

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад, зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. табл. 4.9, аналіз потенційних конкурентів).

#### 4.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 4.14)

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Будівельні та ремонтні компанії	Готові	Високий	Середня	Середня



	та їх працівники				
2.	Усі хто самостійно виконує заміри	Готові	Високий	Середня	Середня
Які цільові групи обрано цільову групу №1 та №2.					

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (табл. 4.15).

Таблиця 4.15 – Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Використання альтернативних технологій та обладнання	Встановлення нового стандарту якості	Зацікавлення та залучення гігантів у галузі будівництва	Стратегія диференціації
2	Дешевизна проекту	Раціональніші витрати на обладнання, та послуги	Максимальне використання наявного обладнання	Стратегія лідерства по витратах

Обрано стратегію диференціації. Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.16).

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект "першопрхідцем" на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1.	Ні	Забирати існуючих та шукати нових	Так, основний функціонал	Наслідування лідера

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (табл. 4.5), а також в залежності від обраної базової стратегії розвитку (табл. 4.15) та стратегії конкурентної поведінки (табл. 4.16) розробляється стратегія позиціонування (табл. 4.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.17 – Визначення стратегії позиціонування

№	Вимоги до	Базова	Ключові	Вибір асоціацій, які мають
---	-----------	--------	---------	----------------------------

п/п	товару цільової аудиторії	стратегія розвитку	конкурентоспроможні позиції власного стартап- проекту	сформувати комплексну позицію власного проекту (три ключових)
1.	Висока якість послуг	Диференціації	Універсальність запропонованого рішення, гарант якості та продуктивності	Якість, продуктивність, надійне з'єднання

#### 4.5 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл. 4.18 наведені результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Якість	Висока якість, сервісність	Якість
2	Дешевизна	Раціональне використання коштів	Дешевизна

Надалі розробляється трьохрівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.19).

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Якісний товар, стандартизована якість обладнання

Після формування маркетингової моделі товару слід особливо відмітити чим саме проект буде захищений від копіювання. Захист може бути організовано за рахунок захисту ідеї товару (захист інтелектуальної власності), або ноу-хау, чи комплексне поєднання властивостей і характеристик, закладене на другому та третьому рівнях товару.

Наступним кроком є визначення цінових меж, якими необхідно керуватись

при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (табл. 4.20). Аналіз проводиться експертним методом.

Таблиця 4.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	1000 у.о.	500 у.о.	Високий	Н. 50 у.о. – В. 200 у.о. (Товар)

Останнім кроком буде визначення оптимальної системи збуту, в межах якого приймається рішення. Так як налаштування обладнання у кінцевого користувача не потребує певних професійних навиків та персоналізму і обладнання не потребує у налаштуванні, то збут доцільно проводити власними силами без застосування посередників.

#### 4.6 Висновки

У процесі роботи було здійснено опис ідеї проекту, що полягає в побудові ультразвукового далекоміру для зручного вимірювання відстані. Застосування мікроконтролерних технологій для побудови ультразвукового далекоміру дозволяє надати не лише великим компаніям але і звичайним користувачам без досвіду використання сам товар, тобто прилад надасть можливість використовувати його усім бажаючим.

## ВИСНОВОК

Таким чином, в ході розробки системи вимірювання відстані переваги і недоліки ультразвукових датчиків. Розроблено систему визначення відстані до об'єкта та відображення її на екрані пристрою.

Програмна частина була реалізована в коді для прошивки мікроконтролера (Arduino скетч). Дана програма повинна визначати відстань до об'єкту за допомогою випромінювання ультразвукових хвиль та відбиття їх від поверхні.

Для візуального відображення вимірюваної відстані було використано багаторозрядний семисегментний індикатор, семисегментний індикатор на базі мікросхеми MAX

Основним матеріалом для дослідження було взято Arduino UNO на основі якого і було побудовано макет приладу для вимірювання відстані.

На основі даної моделі можуть бути спроектовані на розроблені десятки пристроїв для полегшення роботи інженерам, лікарям, водіям, будівельникам, інвалідам. Також ця модель може бути використана при створенні «розумного» будинку.

Для програмування мікроконтролера використовується мова C. Середовище розробки транслює програму на мову Assembler - зрозумілий для мікроконтролера.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Медведев В. С., Лесков А. Г., Ющенко А. С. Системы управления манипуляционных роботов. — М.: Наука, 1978. — 416 с.
2. Фу К., Гонсалес Р., Ли К. Робототехника / Пер. с англ. — М.: Мир, 1989. — 624 с.
3. Автентифікація - <https://uk.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F>
4. Микушин А. Занимательно о микроконтроллерах. — М.: БХВПетербург, 2006
5. Далекомір [Електронний ресурс] // Режим доступу:  
<https://uk.wikipedia.org/wiki/%D0%94%D0%B0%D0%BB%D0%B5%D0%BA%D0%BE%D0%BC%D1%96%D1%80>
6. Ультразвуковий датчик відстані Arduino HC SR04 [Електронний ресурс] // Режим доступу:  
[https://wiki.tntu.edu.ua/%D0%A3%D0%BB%D1%8C%D1%82%D1%80%D0%B0%D0%B7%D0%B2%D1%83%D0%BA%D0%BE%D0%B2%D0%B8%D0%B9\\_%D0%B4%D0%B0%D1%82%D1%87%D0%B8%D0%BA\\_%D0%B2%D1%96%D0%B4%D1%81%D1%82%D0%B0%D0%BD%D1%96\\_Arduino\\_HC\\_SR047](https://wiki.tntu.edu.ua/%D0%A3%D0%BB%D1%8C%D1%82%D1%80%D0%B0%D0%B7%D0%B2%D1%83%D0%BA%D0%BE%D0%B2%D0%B8%D0%B9_%D0%B4%D0%B0%D1%82%D1%87%D0%B8%D0%BA_%D0%B2%D1%96%D0%B4%D1%81%D1%82%D0%B0%D0%BD%D1%96_Arduino_HC_SR047).
7. Сайт о программировании [Електронний ресурс] // Режим доступу:  
metanit.com
8. <https://en.wikipedia.org/wiki/ATmega328>: [Електронний ресурс] // Режим доступу: [https://uk.m.wikipedia.org/wiki/Arduino\\_Uno](https://uk.m.wikipedia.org/wiki/Arduino_Uno)
9. ATmega328: [Електронний ресурс] // Режим доступу:  
<https://en.wikipedia.org/wiki/ATmega328>
10. Пропорційно-інтегрально-диференціальний закон регулювання: [Електронний ресурс] // Режим доступу:  
<https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D0%BE%D1>

[%80%D1%86%D1%96%D0%B9%D0%BD%D0%BE-%D1%96%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE-%D0%B4%D0%B8%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D1%96%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9\\_%D0%B7%D0%B0%D0%BA%D0%BE%D0%BD\\_%D1%80%D0%B5%D0%B3%D1%83%D0%B%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8](#)

11. Функціональна схема: [Електронний ресурс] // Режим доступу: [https://uk.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D1%96%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0\\_%D1%81%D1%85%D0%B5%D0%BC%D0%B0](https://uk.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D1%96%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D1%81%D1%85%D0%B5%D0%BC%D0%B0)

12. Функціональна схема: [Електронний ресурс] // Режим доступу: [https://stud.com.ua/84222/tehnika/elektrichni\\_shemi\\_pravila\\_vikonannya](https://stud.com.ua/84222/tehnika/elektrichni_shemi_pravila_vikonannya)

13. Принципова електрична схема: [Електронний ресурс] // Режим доступу: [https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%86%D0%B8%D0%BF%D0%BE%D0%B2%D0%B0\\_%D0%B5%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%B0\\_%D1%81%D1%85%D0%B5%D0%BC%D0%B0](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%86%D0%B8%D0%BF%D0%BE%D0%B2%D0%B0_%D0%B5%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%B0_%D1%81%D1%85%D0%B5%D0%BC%D0%B0)

14. ГОСТ 2.702-75 Единая система конструкторской документации (ЕСКД). Правила выполнения электрических схем (с Изменениями N 1, 2, 3): [Електронний ресурс] // Режим доступу: <http://docs.cntd.ru/document/1200001981>

15. Структурна схема: [Електронний ресурс] // Режим доступу: [https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BD%D0%B0\\_%D1%81%D1%85%D0%B5%D0%BC%D0%B0](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BD%D0%B0_%D1%81%D1%85%D0%B5%D0%BC%D0%B0)

16. Структурна схема: [Електронний ресурс] // Режим доступу: [https://wiki.tntu.edu.ua/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BD%D0%B0\\_%D1%81%D1%85%D0%B5%D0%BC%D0%B0](https://wiki.tntu.edu.ua/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%BD%D0%B0_%D1%81%D1%85%D0%B5%D0%BC%D0%B0)

## ДОДАТКИ

### Додаток А. Лістинг скетчу далекоміру.

```
#define a 5
#define b 6
#define c 7
#define d 8
#define e 9
#define f 10
#define g 11

#define d1 A4
#define d2 A3
#define d3 A2
#define d4 A1
#define d5 A0

int trigPin = 12;
int echoPin = 13;
long duration, cm, del = 5;

void setup() {
}

void loop() {
  dist();
  clearLEDs();

  if (cm <= 9){
    pickDigit(1);
    pickNumber(cm);
    delay(del);
  }

  if (cm > 9 && cm < 100){

    pickDigit(1);
    pickNumber((cm-(cm%10))/10);
    delay(del);

    pickDigit(2);
    pickNumber(cm%10);
```

```

    delay(del);
}

if (cm > 99 && cm < 1000){

    pickDigit(1);
    pickNumber((cm-(cm%10))/100);
    delay(del);

    pickDigit(2);
    pickNumber((cm%100-cm%10)/10);
    delay(del);

    pickDigit(3);
    pickNumber(cm%10);
    delay(del);
}

if (cm > 999){

    for (int i = 0; i < 5;i++){
        pickDigit(i);
        pickNumber(10);
        delay(del);

    }
}

void pickDigit(int x) {
    digitalWrite(d1, LOW);
    digitalWrite(d2, LOW);
    digitalWrite(d3, LOW);
    digitalWrite(d4, LOW);
    switch (x) {
        case 1: digitalWrite(d1, HIGH); break;
        case 2: digitalWrite(d2, HIGH); break;
        case 3: digitalWrite(d3, HIGH); break;
        case 4: digitalWrite(d4, HIGH); break;
    }
}

void dist()
{

```



```
digitalWrite(trigPin, HIGH);  
delayMicroseconds(5);  
digitalWrite(trigPin, LOW);  
  
pinMode(echoPin, INPUT);  
  
duration = pulseIn(echoPin, HIGH);  
cm = duration/57;  
}
```