

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ УПРАВЛІННЯ, ПСИХОЛОГІЇ
ТА БЕЗПЕКИ**

Кафедра інформаційних технологій

**РОЗРОБЛЕННЯ ЦИФРОВОГО ПРИСТРОЮ ДЛЯ УНИКНЕННЯ
ДРОНОМ ПЕРЕШКОД В ЗАМКНУТИХ ТА ЗАДИМЛЕНИХ
ПРИМІЩЕННЯХ**

Кваліфікаційна робота
здобувача вищої освіти
4 курсу денної форми навчання
Максим МАРХАЙЧУК

Науковий керівник:
доцент, кандидат технічних наук_
Андрій ГОЛОВАТИЙ

Рецензент:
доцент, кандидат технічних наук_

Кваліфікаційна робота допущена до захисту
« ____ » _____ 2026 р., протокол № ____

Завідувач кафедри інформаційних технологій
_____ **Олег ЗАЧЕК**
(підпис)

Львів
2026

СПИСОК УМОВНИХ СКОРОЧЕНЬ

АЗ – апаратне забезпечення (технічна складова системи).

ВС – вбудована система (спеціалізована система керування).

ЕК – електрична схема.

МК – мікроконтролер.

ОЗП (RAM) – оперативний запам'ятовуючий пристрій (енергозалежна пам'ять для тимчасового зберігання даних).

ПЗ – програмне забезпечення.

ПЗП (ROM) – постійний запам'ятовуючий пристрій (тип енергонезалежної пам'яті для зберігання сталого масиву даних).

РКД (LCD) – рідкокристалічний дисплей (пристрій візуалізації інформації).

САПР – система автоматизованого проєктування.

ЦП – цифровий пристрій.

AVR – архітектура 8-бітних мікроконтролерів, розроблена компанією Atmel.

EEPROM – тип енергонезалежної пам'яті, що допускає багаторазове перезаписування та стирання даних за допомогою електричного сигналу.

I2C – послідовний шинний інтерфейс, що забезпечує зв'язок між інтегральними модулями за допомогою двох ліній (SDA та SCL).

SPI – високошвидкісний синхронний послідовний протокол передачі даних у повнодуплексному режимі для взаємодії МК із периферійними компонентами.

АНОТАЦІЯ

Мархайчук М., Головатий А. (керівник). Розроблення цифрового пристрою для уникнення дроном перешкод в замкнутих та задимлених приміщеннях. Бакалаврська кваліфікаційна робота. – Львівський державний університет внутрішніх справ, Львів, 2026.

У роботі реалізовано цифровий пристрій на базі МК родини AVR (платформа Arduino) та ультразвукового датчика HC-SR04 для забезпечення автономного маневрування дрона в умовах обмеженого простору та сильного задимлення.

Розроблений пристрій забезпечує швидкий збір даних щодо наявності об'єктів у зоні сканування та точне визначення дистанції до них. Наявність адаптивного меню конфігурації забезпечує можливість індивідуального налаштування режимів візуальної індикації та звукового сповіщення у разі фіксації об'єктів у зоні сканування.

Функціонал детектора забезпечує візуалізацію поточного стану системи та результатів вимірювань на символьному LCD-індикаторі. Користувачеві доступні дані про режим функціонування, виявлені об'єкти та точну дистанцію до них. Окрім індикації вимірів, дисплей відображає параметри конфігурації (порогові відстані, режими звукової сигналізації). Для зовнішнього моніторингу передбачено трансляцію даних через UART.

Створено структурну та електричну принципову схеми детектора перешкод. Розроблено керуючу програму на мові C та алгоритм взаємодії компонентів пристрою в середовищі Arduino IDE. Працездатність цифрового детектора перешкод підтверджено шляхом моделювання в емуляторі Proteus ISIS.

Ключові слова: ультразвуковий далекомір HC-SR04, пристрій виявлення перешкод, мікроконтролерна платформа Arduino, архітектура AVR, імітаційне моделювання Proteus VSM, інтегроване середовище Arduino IDE.

ABSTRACT

Markhaychuk M., Holovatyy A. (supervisor). Development of a digital device for drone obstacle avoidance in enclosed and smoky spaces. Bachelor's thesis. – Lviv State University of Internal Affairs, Lviv, 2026.

The work implements a digital device based on the AVR family microcontroller (Arduino platform) and the HC-SR04 ultrasonic sensor to ensure autonomous drone maneuvering in confined spaces and heavy smoke.

The developed device provides rapid data collection regarding the presence of objects in the scanning zone and accurate determination of the distance to them. The presence of an adaptive configuration menu provides the ability to individually configure visual indication modes and sound notification in case of fixation of objects in the scanning zone.

The detector functionality provides visualization of the current system status and measurement results on a symbolic LCD indicator. The user has access to data on the operating mode, detected objects and the exact distance to them. In addition to the measurement indication, the display shows configuration parameters (threshold distances, sound alarm modes). For external monitoring, data transmission via UART is provided.

The structural and electrical schematic diagrams of the interference detector have been created. A control program in C language and an algorithm for the interaction of the device components in the Arduino IDE environment have been developed. The operability of the digital interference detector has been confirmed by simulation in the Proteus ISIS emulator.

Keywords: ultrasonic rangefinder HC-SR04, obstacle detection device, Arduino microcontroller platform, AVR architecture, Proteus VSM simulation, Arduino IDE integrated environment.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ.....	9
1.1. Методи вимірювання віддалі та види далекомірів.....	9
1.2. Огляд існуючих підходів для побудови систем вимірювання відстані.....	15
РОЗДІЛ 2. ЗАСОБИ РОЗРОБКИ АПАРАТНОГО І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО ПРИСТРОЮ ДЛЯ УНИКНЕННЯ ПЕРЕШКОД ДРОНОМ В ЗАМКНУТИХ ТА ЗАДИМЛЕНИХ ПРИМІЩЕННЯХ.....	22
2.1. Інструментарій проектування та симуляції програмованих пристроїв Proteus VSM....	22
2.2. Інструментарій Arduino IDE для розробки програмного забезпечення МК-систем на платформі Arduino.....	26
РОЗДІЛ 3. АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО ПРИСТРОЮ ДЛЯ УНИКНЕННЯ ДРОНОМ ПЕРЕШКОД В ЗАМКНУТИХ ТА ЗАДИМЛЕНИХ ПРИМІЩЕННЯХ.....	36
3.1. Визначення апаратних засобів для проектування цифрового пристрою для уникнення перешкод.....	36
3.2. Проектування цифрового пристрою для уникнення дроном перешкод в замкнутих та задимлених приміщеннях в САПР Proteus VSM.....	50
РОЗДІЛ 4. ПРОГРАМНО-АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО ПРИСТРОЮ ДЛЯ УНИКНЕННЯ ДРОНОМ ПЕРЕШКОД В ЗАМКНУТИХ ТА ЗАДИМЛЕНИХ ПРИМІЩЕННЯХ.....	57
4.1. Алгоритм роботи цифрового пристрою для уникнення перешкод.....	57
4.2. Створення програмного забезпечення для інтерфейсу з ультразвуковим далекоміром HC-SR04.....	66
4.3. Створення програмних компонентів для керування режимами роботи пристрою для уникнення перешкод.....	66
4.4. Розроблення основних програмних функцій цифрового пристрою для уникнення перешкод.....	70
4.5. Програмна симуляція та оцінка результатів функціонування цифрового пристрою для уникнення перешкод в середовищі Proteus ISIS.....	70
ВИСНОВКИ.....	87
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	89
ДОДАТКИ.....	91

ВСТУП

Сучасний розвиток сенсорних технологій та мікроконтролерних систем відкриває широкі можливості для проектування бюджетних інтелектуальних детекторів. Прикладом такої розробки є цифровий пристрій на базі Arduino, призначений для навігації дронів у складних умовах, зокрема в задимлених або замкнених просторах. Пристрій забезпечує автоматичне виявлення об'єктів, прецизійне дистанціювання та оповіщення оператора через звукову індикацію. Крім того, пристрій оснащено символьним РК-дисплеєм із розгалуженим меню для адаптивного керування режимами роботи.

Сфера експлуатації даного детектора є досить широкою. Зокрема, він може слугувати допоміжним засобом для соціальної адаптації осіб з інвалідністю по зору, а також інструментом для фахівців, що працюють в умовах обмеженої видимості (рятувальники, спелеологи). Окрім цивільного сектору, пристрій знаходить застосування в оборонних технологіях, наприклад, у системах автономного пілотування БПЛА. Функціональні можливості апарату дозволяють використовувати його для забезпечення техніки безпеки на промислових підприємствах, у медичній реабілітації, а також для проведення замірів у місцях, де прямий доступ до об'єктів неможливий через природні або технічні бар'єри.

Роль детектора перешкод у системі виконує цифровий ультразвуковий датчик HC-SR04. Цей сучасний сенсор дозволяє проектувати ефективні далекоміри малого радіуса дії. Серед ключових переваг модуля варто виділити його економічну доступність, компактні габарити та здатність забезпечувати достатню точність при дистанційному зондуванні об'єктів.

Проектування апаратної частини детектора перешкод базується на використанні платформи Arduino, серцем якої є мікроконтролер архітектури AVR від компанії Atmel. Такий вибір зумовлений гнучкістю апаратних рішень та широкими можливостями інтеграції периферійних пристроїв для обробки сигналів у реальному часі.

Екосистема Arduino стала основою для проектування великої кількості сучасних інтелектуальних систем та контрольно-вимірювальних приладів. Використання даного інструментарію суттєво прискорює етап програмної реалізації керуючих алгоритмів. Платформа базується на мікроконтролерах архітектури AVR, які оснащені розвиненою периферією: універсальними портами вводу-виводу та стандартними комунікаційними інтерфейсами.

Аналіз останніх досліджень і публікацій. Дослідженню можливостей використання МК для створення цифрових пристроїв присвячено різноманітні публікації і проекти, що дають основу для подальшої розробки цифрового пристрою для уникнення дронам перешкод в замкнутих та задимлених приміщеннях.

Актуальність теми зумовлена обмеженістю сучасних систем навігації БПЛА, що базуються на оптичних сенсорах та GPS. У замкнутих просторах супутниковий сигнал блокується, а в умовах задимлення камери та лідари стають неефективними через розсіювання світла. Розроблення цифрового пристрою на основі альтернативних методів зондування (наприклад, ультразвукових) дозволяє забезпечити безаварійне маневрування дрона в умовах нульової видимості, що є критично важливим для автономних інспекційних систем.

У зв'язку зі сказаним можемо сформулювати мету та завдання кваліфікаційної роботи.

Метою роботи є розробка архітектури, електричної схеми та програмного коду цифрового детектора перешкод з використанням екосистеми Arduino.

Для досягнення мети необхідно вирішити наступні **завдання**:

- спроектувати архітектуру цифрового пристрою для уникнення дронам перешкод в замкнутих та задимлених приміщеннях в середовищі Proteus VSM на платформі Arduino Uno з використанням ультразвукового датчика HC-SR04 та LCD-індикатора WH1602B;

- написати програмний модуль для точного вимірювання дистанції шляхом обробки сигналів датчика HC-SR04;
- розробити логіку прийняття рішень для маневрування дрона при виявленні об'єктів у складних умовах навколишнього середовища;
- написання основного коду в Arduino IDE, що об'єднує збір даних, їх обробку та виведення поточної інформації на дисплей;
- проєктувати працездатності пристрою в симуляторі Proteus ISIS, провести аналіз точності спрацювання алгоритмів у віртуальному середовищі.

Об'єкт дослідження – фізичні принципи та технічні характеристики ультразвукових засобів дистанційного зондування та вимірювання відстані.

Предмет дослідження – методи проєктування та алгоритми функціонування детекторів перешкод із функцією обчислення дистанції до об'єктів.

Методи досліджень. Для досягнення поставленої мети та розв'язання визначених завдань було використано метод системного проєктування для розробки структурної схеми пристрою, що об'єднує Arduino, ультразвуковий датчик та систему індикації/керування, метод комп'ютерного моделювання у Proteus VSM для створення інтерактивної моделі пристрою для перевірки правильності логіки роботи МК в режимі реального часу, метод віртуальної наладки для аналізу часових діаграм сигналів, що критично важливо для точності HC-SR04, метод ітераційного прототипування для поступового вдосконалення програмного коду на основі результатів симуляції перед перенесенням на реальну плату.

Структура роботи. Кваліфікаційна робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел, додатків. Обсяг основного тексту роботи складає 87 сторінок, 76 рисунків, 5 таблиць, 1 додаток і 20 бібліографічних джерела. Загальний обсяг роботи – 94 сторінки.

РОЗДІЛ 1

СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1. Методи вимірювання віддалі та види далекомірів

Дистанція між двома точками у просторі – це кількісний показник їхнього взаємного розташування. У фізичних розрахунках та повсякденному житті під цим терміном зазвичай розуміють протяжність прямолінійного відрізка, що пролягає між цими об'єктами.

Для позначення дистанції у розрахунках найчастіше використовують символ d . Відповідно до стандартів системи SI та загальноприйнятих метричних норм, її виражають у таких одиницях поздовжньої міри, як метри (m), кілометри (km) або їхні похідні (сантиметри, міліметри тощо).

Обчислення дистанції є критично важливим етапом при відтворенні різноманітних фізичних процесів. Традиційно для цього застосовують стандартні фізичні еталони (зокрема метр), проте сучасні технологічні рішення дозволяють використовувати методи дистанційного зондування. До таких безконтактних засобів належать дальноміри, які забезпечують високу точність вимірів без прямого дотику до об'єкта.

Далекомір (міжнародні назви: англ. *Rangefinder*, нім. *Entfernungsmesser*) являє собою спеціалізований пристрій, призначений для оперативного визначення просторової дистанції між точкою спостереження та ціллю або безпосередньо між двома об'єктами. Завдяки високій точності та швидкодії, ці прилади стали незамінними інструментами у картографії, мореплавстві, дослідженні космічних тіл, а також у складі сучасних тактичних систем та озброєння.

Найбільшого розповсюдження в інженерній практиці набули оптичні системи та далекоміри з активним принципом зондування. Перші базуються на використанні візуальних ефектів та лінз, тоді як другі самостійно випромінюють сигнал (звук, світло або радіохвилю) для обчислення дистанції до об'єкта.

Оптичні далекоміри. В основі функціонування оптичних далекомірних систем лежить метод реєстрації оптичного паралакса. Процес обчислення дистанції в таких пристроях базується на тригонометричних розрахунках параметрів рівнобедреного трикутника. Для знаходження шуканої величини необхідно знати довжину фіксованої хорди (бази) та величину відповідного паралактичного кута, що лежить навпроти неї.

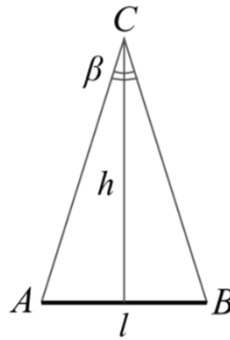


Рис. 1.1. Схематичне представлення оптичного способу визначення дистанції

Шукану величину h (відстань до цілі C) можна обчислити, спираючись на геометричну залежність. Для цього необхідно мати вихідні дані про протяжність бази l , а також значення прилеглих до неї кутів A та B .

Залежно від способу формування паралактичного кута, оптичні далекомірні системи поділяють на візуальні, світлопроекційні та комбіновані (проекційно-візуальні). Відмінність між ними полягає у фізичній природі осей, що утворюють кут. У візуальних приладах він обмежений візирними лініями спостереження. У світлопроекційних — центральними осями випромінюваних світлових потоків. У змішаних — взаємним розташуванням візирної осі та осі світлового пучка.

Розрізняють два основні типи розташування бази відносно приладу: зовнішнє: база знаходиться на протилежному кінці дистанції, що вимірюється (характерно для далекомірів із рейками); внутрішнє: база інтегрована в саму конструкцію приладу (прикладом є безрейкові та внутрішньобазові пристрої).

Серед оптичних візуальних далекомірів найбільшого поширення набули ниткові моделі, а також монокулярні прилади подвійного зображення з

використанням рейок. Окрему групу становлять внутрішньобазові пристрої, де за формування паралактичного кута відповідає оптичний компенсатор. Також широко застосовуються стереоскопічні далекоміри, робота яких ґрунтується на природній здатності людини сприймати об'ємне зображення обома очима (стереоефект).

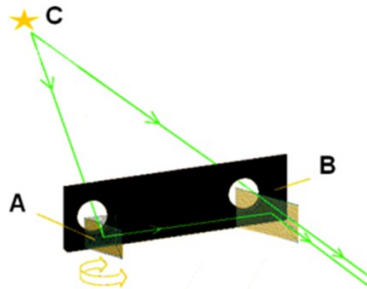


Рис. 1.2. Конструктивні особливості та схема функціонування

Далекоміри активного типу. Функціонування активних далекомірних систем базується на реєстрації часового інтервалу, необхідного для проходження зондувального сигналу від випромінювача до цілі та в зворотному напрямку. В основі розрахунків лежить порівняльний аналіз параметрів випроміненого та відбитого (ехо-сигналу) імпульсів за умови, що швидкість поширення хвилі (світлової або звукової) є константою. Ця методологія є базовою для роботи лазерних, електрооптичних, радіотехнічних, а також акустичних та GPS-далекомірів.

Дистанційне визначення координат без безпосереднього контакту з об'єктами значно розширює технічні можливості у багатьох галузях. Серед найбільш доступних та економічно ефективних засобів, створених інженерами, виділяється ультразвуковий далекомір. Такі пристрої, що часто класифікуються як ехолоти, отримали значне поширення завдяки своїй універсальності та простоті інтеграції у різні системи моніторингу та навігації.

Архітектура ультразвукового далекоміра вирізняється своєю лаконічністю. Конструктивно пристрій об'єднує в собі модулі випромінювання та прийому сигналу, а також мікропроцесорний блок. Останній відповідає за логічні операції: накопичення інформації, її математичну обробку та подальше виведення результатів вимірювання.

Функціонування даного детектора базується на ехо-методі: прилад генерує акустичний імпульс у діапазоні, невидимому для людського вуха. Цей сигнал, досягаючи перешкоди, відбивається і повертається до сенсора. Оскільки швидкість звукових хвиль у повітряному середовищі при незмінній густині є стабільною величиною, це дає змогу точно обчислити дистанцію до об'єкта.

Еволюція наукових знань та вдосконалення промислових потужностей сприяли поступовій модернізації конструкції вимірювальних приладів. Зокрема, для забезпечення прецизійного спрямування акустичного потоку на цільовий об'єкт було створено ультразвукові далекоміри, інтегровані з лазерним цілевказівником. Таке технічне рішення суттєво спростило експлуатацію пристрою та мінімізувало похибки при визначенні дистанції.

Завдяки інтеграції сучасного програмного забезпечення та високопродуктивних мікроконтролерів (МК), функціонал новітніх далекомірів вийшов далеко за межі звичайної фіксації дистанції. Сучасні пристрої здатні виконувати комплексні обчислювальні операції, зокрема визначати площу об'єктів та розраховувати кутові координати цілей у реальному часі. Розширення аналітичних можливостей у поєднанні з низькою вартістю, порівняно з лазерними системами, зумовило стрімке зростання попиту на ультразвукові вимірювальні рішення.

Поряд із перевагами, необхідно виділити суттєві обмеження, характерні для засобів ультразвукової локації. Ключовим чинником, що впливає на метрологічні характеристики, є залежність точності від параметрів середовища поширення хвиль. Оскільки швидкість звуку корелює з густиною середовища, будь-які коливання останньої безпосередньо призводять до виникнення похибок. Крім того, фізичні властивості акустичних коливань накладають обмеження на максимальний радіус дії таких пристроїв.

Гранична дальність функціонування таких пристроїв зазвичай не перевищує 20 метрів. Проте найбільш критичне обмеження ультразвукових вимірювачів обумовлене фундаментальними властивостями самого

акустичного сигналу. Ключовим аспектом, що визначає поведінку системи, є хвильова природа звуку, яка безпосередньо впливає на особливості його поширення та взаємодії з перешкодами.

Для розуміння впливу хвильової природи звуку на метрологічні характеристики приладу можна скористатися гідродинамічною аналогією. Подібно до того, як хвилі на поверхні води поступово розширюються, віддаляючись від місця падіння об'єкта, акустичний фронт також збільшує свою площу в міру віддалення від випромінювача. Це явище, відоме як розходження (дивергенція) звукового пучка, призводить до того, що на значних дистанціях енергія сигналу розподіляється на більшу площу, що ускладнює точне фокусування на цілі.

Наявність лазерного цілевказівника в сучасних ультразвукових далекомірах часто створює хибне враження, що вимірювання відбувається строго в точці лазерної плями. Проте, через конусоподібну форму поширення звукового фронту, сигнал може відбитися від будь-якого об'єкта (наприклад, меблів), що опинився в межах його діаграми спрямованості. Якщо така перешкода розташована ближче за цільову стіну, приймач зафіксує передчасне відлуння, що призведе до хибного результату. Саме широка апертура променя ускладнює коректне вимірювання відстаней у кутах приміщень або визначення довжини діагоналей.

Крім фізичних параметрів середовища, на метрологічну надійність ультразвукового далекоміра суттєво впливають акустичні характеристики поверхні об'єкта. Матеріали з високим коефіцієнтом звукопоглинання (наприклад, пористі або м'які структури) здатні частково гасити енергію зондувального імпульсу. Це призводить до ослаблення амплітуди відбитого ехо-сигналу, що негативно позначається на здатності приладу коректно фіксувати часовий інтервал та знижує загальну точність вимірювання.

Експлуатація приладів на основі ультразвукової локації є доцільною лише у випадках, коли до результатів не висуваються високі вимоги щодо точності, а цільові об'єкти розташовані в межах їхнього обмеженого радіуса

дії. Для проведення прецизійних вимірювань або роботи на значних дистанціях стандартом є використання лазерних далекомірних систем, які позбавлені багатьох недоліків акустичного методу.

Лазерні вимірювальні прилади характеризуються значним робочим діапазоном: мінімальна дистанція вимірювання стартує від 20–30 мм, тоді як максимальна може сягати 100–200 метрів. Показники похибки при цьому залишаються мінімальними: для пристроїв базового рівня вони не перевищують ± 3 мм, а в професійному сегменті забезпечується прецизійна точність до $\pm 1,5$ мм.

Попри високу ефективність, лазерні рулетки мають певні експлуатаційні обмеження. Зокрема, інтенсивне сонячне випромінювання суттєво знижує їхню робочу дистанцію. У сонячну погоду реальна дальність вимірювання може скоротитися зі заявлених 100–200 м до 30–50 м. Це зумовлено високим рівнем фонових завад у сонячному спектрі, які накладаються на лазерний сигнал, що призводить до значного зниження співвідношення сигнал/шум та ускладнює детектування відбитого променя.

Ефективний радіус дії лазерного далекоміра також корелює з коефіцієнтом відбиття поверхні цільового об'єкта. Оскільки різні матеріали мають різну поглинальну здатність, для досягнення максимальних дистанцій, зазначених у технічному паспорті, необхідно застосовувати спеціалізовані світловідбивальні пластини. Як правило, такі аксесуари не постачаються в базовій комплектації пристрою та потребують окремого придбання.

Окрему групу чинників, що дестабілізують роботу лазерних далекомірів, становлять специфічні оптичні властивості об'єктів та стан навколишнього середовища. Взаємодія з прозорими або дзеркальними поверхнями часто призводить до отримання недостовірних результатів, оскільки скло здатне пропускати лазерне випромінювання крізь себе. Крім того, експлуатація приладів у середовищах із високою концентрацією пилу або диму є малоефективною, оскільки завислі частки викликають інтенсивне розсіювання променя, що унеможливорює коректну фіксацію відбитого сигналу.

Практичний досвід підтверджує, що для визначення метричних характеристик під час будівництва лазерний далекомір є незамінним інструментом, якому наразі немає рівних за ефективністю.

Сучасне покоління лазерних далекомірів вирізняється розширеним функціоналом. Зокрема, пристрої оснащені місткою внутрішньою пам'яттю, інтерфейсами для синхронізації з ПК та інструментами для складних обчислень (визначення об'єму, кутів нахилу або розрахунку гіпотенузи). Слід зауважити, що наявність таких аналітичних можливостей значно підвищує вартість обладнання.

1.2. Огляд існуючих підходів для побудови систем вимірювання відстані

Аналіз наукових джерел та сучасних інтернет-ресурсів дозволив зібрати достатній обсяг даних, необхідних для проєктування та технічної реалізації пристрою виявлення перешкод.

Сучасний ринок пропонує уніфіковані підходи до побудови систем дистанційного вимірювання. Найбільшого поширення набули пристрої, що базуються на лазерних та ультразвукових технологіях.

Модель NoHawk A2-1000 Camo належить до класу прецизійних лазерних далекомірів, що забезпечують визначення відстані в діапазоні від 3,5 до 1500 метрів. Похибка вимірювань для цього пристрою не перевищує $\pm 0,5$ м (Рис. 1.3).



Рис. 1.3. Зовнішній вигляд прецизійного лазерного далекоміра NoHawk A2-1000 Camo

Прилад Mächtz MDM-70 P (Рис. 1.4) призначений для визначення відстаней у межах від 0,05 до 70 метрів. Його відмінною рисою є висока точність, де похибка становить $\pm 0,02$ мм на кожен метр дистанції. Додатково пристрій оснащений вбудованим цифровим датчиком для вимірювання кутів у діапазоні 0-90°.



Рис. 1.4. Лазерний вимірювальний прилад Mächtz MDM-70 P, оснащений електронним інклінометром

Пристрій Mächtz MDM-70 P оснащений інтегрованим модулем пам'яті, що дозволяє зберігати до 99 останніх результатів вимірювань. Крім того, функціонал приладу передбачає можливість перемикання між різними системами одиниць, зокрема доступні метри, фути та дюйми.

З ультразвукових далекомірів на ринку є:

Ультразвуковий прилад марки Stanley (Рис. 1.5) призначений для визначення дистанцій у межах до 12 метрів, при цьому відносна похибка вимірювань не перевищує $\pm 0,5\%$. Апаратне забезпечення пристрою дозволяє автоматично розраховувати площу та об'єм об'єктів, а на дисплеї одночасно відображаються дані трьох попередніх замірів.



Рис. 1.5. Зовнішній вигляд ультразвукового далекоміра Stanley для побутового та професійного використання

Акустичний далекомір Technoline WZ1100 (рис. 1.6) забезпечує визначення відстані в діапазоні до 16 метрів з граничною похибкою 1%. Окрім стандартного функціонала (визначення довжини, площі та об'єму), пристрій підтримує арифметичне додавання результатів та візуалізацію проміжних обчислень. Додатковими особливостями моделі є вбудований термометр для контролю температури середовища та індикатор стану елементів живлення.



Рис. 1.6. Зовнішній вигляд ультразвукового далекоміра Technoline моделі WZ1100 Silver/Green

Ультразвуковий сенсор моделі HC-SR04 (Рис.1.7) забезпечує детектування об'єктів на відстані до 4 метрів із мінімальною похибкою, що не перевищує 3 мм. Завдяки своїй низькій вартості та простоті інтеграції, даний модуль є оптимальним рішенням для реалізації систем виявлення перешкод у бюджетних проєктах.



Рис. 1.7. Ультразвуковий сенсорний модуль HC-SR04 для безконтактного вимірювання відстані

Принцип роботи ультразвукового модуля базується на використанні акустичних хвиль для виявлення об'єктів та визначення дистанції до них (Рис. 1.8). Пристрій функціонує у високочастотному діапазоні, що знаходиться поза межами сприйняття людського слуху.

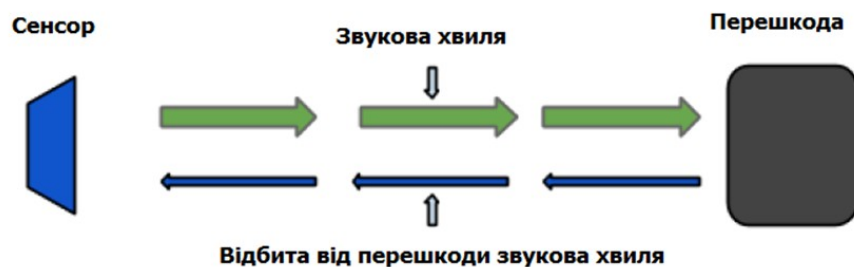


Рис. 1.8. Схематичне представлення ехолокаційного принципу визначення відстані

Випромінювач датчика генерує акустичний сигнал на фіксованій частоті 40 кГц, після чого система переходить у режим очікування ехо-сигналу, відбитого від об'єкта. Внутрішня логіка модуля фіксує часовий інтервал між моментом випромінювання та моментом реєстрації відбитої хвилі.

Маючи дані про швидкість поширення акустичних коливань у конкретному середовищі, можна розрахувати дистанцію до об'єкта. Оскільки звукова хвиля долає відстань двічі (до перешкоди та у зворотному напрямку), математична залежність набуває вигляду:

$$d = v \cdot \frac{t}{2} \quad (1.1)$$

де d – шукана відстань до об'єкта; v – швидкість звуку в робочому середовищі (для повітря при 20°C становить приблизно 343 м/с); t – повний час проходження сигналу (від випромінювання до реєстрації ехо-імпульсу).

Фазова швидкість поширення акустичних хвиль у повітряному середовищі не є константою, оскільки вона детермінована станом атмосфери. Ключовими чинниками, що впливають на цей параметр, є термодинамічна температура, відносна вологість та атмосферний тиск.

Найбільш суттєвий вплив на швидкість поширення акустичних хвиль має температурний фактор: спостерігається пряма кореляція між ростом температури середовища та швидкістю звуку. Так, при 0°C цей параметр становить 331 м/с , тоді як при $+20^{\circ}\text{C}$ він зростає до 343 м/с . При подальшому нагріванні до $+60^{\circ}\text{C}$ та $+100^{\circ}\text{C}$ швидкість досягає 366 м/с та 387 м/с відповідно. Слід також зазначити, що у рідких середовищах швидкість звуку зазвичай значно перевищує показники, характерні для атмосферного повітря.

У Табл.1.1 наведено вплив температурних показників середовища на швидкість поширення акустичних коливань у повітрі.

Таблиця 1.1. Вплив температурних показників середовища на швидкість поширення акустичних коливань у повітрі

$\theta, ^{\circ}\text{C}$	$s, \text{м/с}$	$\rho, \text{кг/м}^3$	$Z, \text{Н}\cdot\text{с/м}^{-3}$
-10	325,2	1,342	436,1
-5	328,3	1,317	432,0
0	331,3	1,292	428,4
+5	334,3	1,269	424,3
+10	337,3	1,247	420,6
+15	340,3	1,225	416,8
+20	343,2	1,204	413,2
+25	346,1	1,184	409,8
+30	349,0	1,165	406,3

Результати експериментальних досліджень підтверджують, що в широкому спектрі значень тиск навколишнього середовища практично не впливає на швидкість поширення звукових хвиль. Помітне зростання швидкості спостерігається лише в умовах екстремально високого тиску

(близько 100 атм.). Водночас температурний фактор відіграє ключову роль: при нагріванні газового середовища швидкість звуку зростає. Зокрема, у повітрі за умов, близьких до кімнатних ($+20^{\circ}\text{C}$), кожне підвищення температури на один градус спричиняє збільшення швидкості приблизно на 0,6 м/с.

Акустичні коливання пружного середовища, частота яких перевищує поріг чутності людського вуха (20 кГц), класифікуються як ультразвук. Характерною особливістю ультразвукових сенсорів є наявність діаграми спрямованості (кута детектування), геометрія якої змінюється відповідно до дистанції до об'єкта (Рис. 1.9).

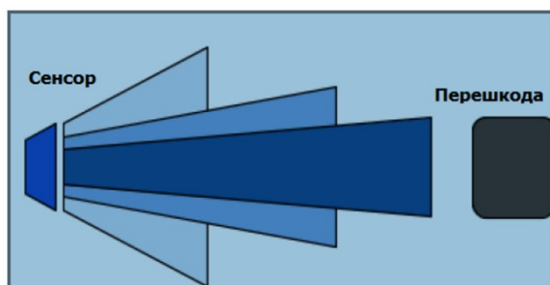


Рис. 1.9. Діаграма спрямованості ультразвукового випромінювача: кореляція між кутом виявлення та дистанцією до об'єкта

Ефективність функціонування ультразвукового модуля визначається не лише дистанцією, а й просторовою орієнтацією об'єкта відносно осі випромінювання. Існує ймовірність явища дифракції, коли акустична хвиля огинає перешкоду, або дзеркального відбиття під кутом, через що відлуння не повертається до приймача сенсора.

Використання ультразвукових систем є доцільним у випадках, коли пріоритетом не є висока точність метрологічних показників або детектування об'єктів у безпосередній близькості до датчика.

У кваліфікаційній роботі реалізовано програмно-апаратний комплекс на базі контролера Arduino та ультразвукового сенсора HC-SR04. Розроблений пристрій забезпечує автоматичне виявлення перешкод, прецизійне вимірювання дистанції до них та генерацію попереджувальних сигналів у разі критичного зближення. Передбачено механізм моніторингу даних через

послідовний порт і можливість оперативного корегування робочих параметрів детектора.

Спроектований цифровий детектор оснащений інтерфейсом користувача у вигляді меню для оперативного керування робочими режимами. На символьний LCD-дисплей виводяться поточні налаштування та результати вимірювання дистанції до об'єктів. Функціонал пристрою передбачає систему звукової та візуальної індикації тривожних станів, а також блок контролю енергоспоживання, що сповіщає про критичне зниження напруги живлення.

Програмно-апаратну реалізацію алгоритмів роботи детектора доцільно здійснити на базі платформи Arduino. Вибір цього рішення обумовлений наявністю розлогого інструментарію та численних відкритих бібліотек драйверів, сумісних із середовищем розробки Arduino IDE, що значно спрощує процес інтеграції компонентів.

РОЗДІЛ 2

ЗАСОБИ РОЗРОБКИ АПАРАТНОГО І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО ПРИСТРОЮ ДЛЯ УНИКНЕННЯ ПЕРЕШКОД ДРОНОМ В ЗАМКНУТИХ ТА ЗАДИМЛЕНИХ ПРИМІЩЕННЯХ

2.1. Інструментарій проектування та симуляції програмованих пристроїв Proteus VSM

Середовище Proteus VSM – це ефективний інструментарій для розробки та тестування електронних схем. Програма дозволяє здійснювати програмну емуляцію складних аналого-цифрових пристроїв, включаючи мікропроцесори та контролери різних архітектур, що робить її незамінною для верифікації проектів на базі платформ типу Arduino без використання фізичного обладнання. Даний програмний комплекс класу САПР (Electronic Design Automation) є продуктом британської компанії Labcenter Electronics. В основу системи покладено принципи схемотехнічного аналізу, що використовують адаптовані математичні моделі компонентів стандарту PSpice. Ключовою перевагою Proteus Design є підтримка динамічної симуляції обчислювальних засобів: від базових мікроконтролерів до складних мікропроцесорних структур та цифрових сигнальних процесорів (DSP).

Інструментарій Proteus VSM надає розробнику доступ до 6000+ віртуальних компонентів із відповідними специфікаціями та готових проектів для тестування. Окрім засобів симуляції, пакет містить спеціалізований модуль для професійного проектування друкованих плат, що забезпечує комплексну розробку пристрою. Архітектура Proteus розділена на два функціональні блоки: ISIS для моделювання схемотехніки та ARES для трасування плат. Комплект поставки містить готові проекти для тестування можливостей системи. Завдяки наявності спеціалізованих драйверів COMPM та USBCONN, розробник може синхронізувати роботу віртуальної моделі з реальними портами введення-виведення комп'ютера.

Інтеграція зовнішніх пристроїв через ці інтерфейси дозволяє віртуальній схемі функціонувати в режимі реального часу, імітуючи роботу справжнього заліза. Середовище Proteus VSM забезпечує підтримку популярних інструментаріїв розробки: від WinAVR та CodeVisionAVR для мікроконтролерів AVR до професійних засобів Keil та ICC для ARM та Motorola. Важливою особливістю є можливість використання сторонніх моделей електронних вузлів, експортованих із форматів PSpice.

Функціонал Proteus VSM дозволяє симулювати роботу електронних вузлів і проводити діагностику помилок ще до створення фізичного прототипу. Використовуючи віртуальний інструментарій, розробник може не лише аналізувати фізичні параметри схеми, а й здійснювати повний цикл дебагінгу (відлагодження) прошивок для різних архітектур мікропроцесорів.

Попри те, що віртуальна симуляція не завжди на 100% ідентична реальним процесам, вона є ефективним інструментом для тестування мікроконтролерного ПЗ. Пакет Proteus пропонує функцію 3D-візуалізації, яка забезпечує наочне відображення спроектованих друкованих плат у просторі. Дана САПР демонструє стабільну роботу під управлінням ОС Windows (включаючи як застарілі, так і сучасні версії).

Середовище Proteus володіє вбудованою базою електронних модулів, кожен з яких супроводжується необхідною документацією для симуляції. Наявність детальних специфікацій для мікроконтролерів та напівпровідникових приладів дозволяє розробнику точно відтворювати логіку функціонування апаратної частини проєкту.

Інструментарій Proteus надає доступ до величезної бази даних, що охоплює тисячі найменувань дискретних приладів та інтегральних схем різного типу. У разі відсутності специфічного компонента, програма пропонує засоби для створення власних моделей. Важливою перевагою є підтримка галузевого стандарту SPICE, що дає змогу інтегрувати в проєкт актуальні моделі від провідних світових виробників напівпровідників.

До складу системи Proteus входять два функціональні блоки: ISIS (інструментарій для моделювання та схмотехнічного синтезу) і ARES (засіб розробки друкованих плат). Останній підтримує інтелектуальні алгоритми розстановки елементів та автоматичне розведення провідників за допомогою трасувальника ELECTRA. Для швидкого старту та ознайомлення з можливостями ПЗ користувачеві доступний набір готових прикладних рішень.

На Рис. 2.1 наведено загальний вигляд робочого простору системи Proteus ISIS. Результат тривимірного моделювання спроектованого вузла засобами візуалізації пакета Proteus ARES відображено на Рис. 2.2.

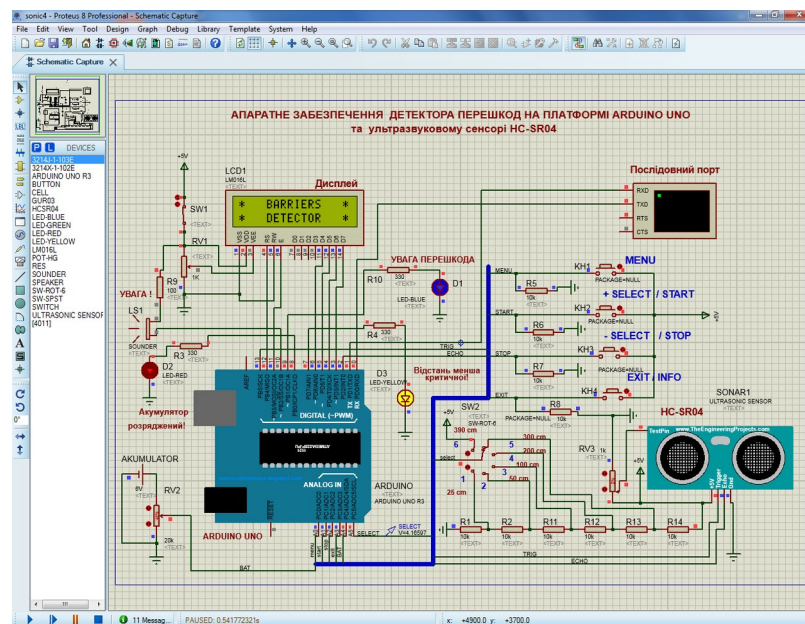


Рис. 2.1. Середовище схмотехнічного моделювання Proteus ISIS

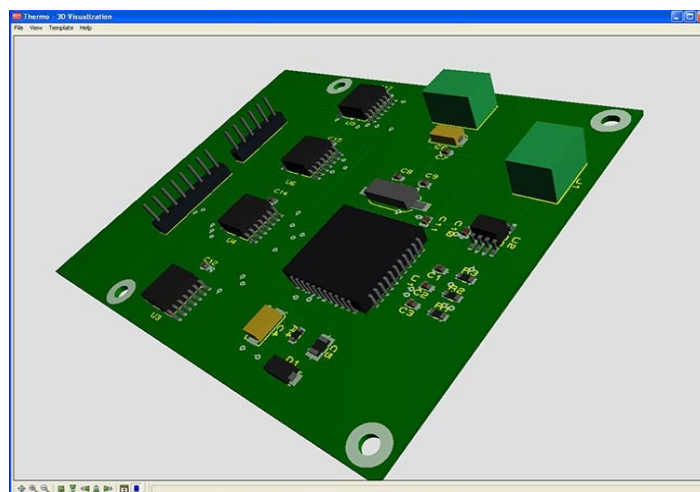


Рис. 2.2. Результат тривимірного моделювання топології друкованої плати в модулі Proteus ARES

Стислий опис функціональних можливостей панелі інструментів (Module Bar) у середовищі Proteus ISIS наведено на Рис. 2.3-2.4, Табл.2.1.

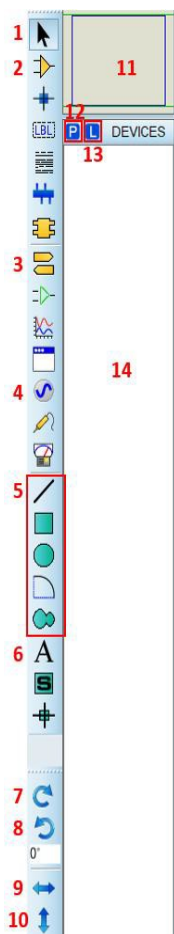


Рис. 2.3. Склад і структура головної панелі інструментів середовища Proteus

Таблиця 2.1. Функція та призначення інструментів головної панелі

№	Функція	Технічне призначення
1–4	Робота з проектами	Створення, відкриття, запис даних та закриття робочого файлу.
5–8	Оптимізація перегляду	Центрування, регулювання масштабу та повне відображення схеми.
9–10	Скасування/Повтор	Керування історією змін для виправлення помилок проєктування.
11–13	Буфер обміну	Стандартні операції для дублювання або переміщення вузлів схеми.
14	Селектор	Активація режиму вибору та налаштування параметрів елементів.

Бокова вертикальна панель інструментів містить (Рис.2.4):



1. Select Mode – режим виборки.
2. Component Mode – режим електронних компонентів.
3. Terminals Mode – термінальний режим.
4. Generator Mode – генераторний режим
5. 2D Shapes – 2D шаблони.
6. Text Mode – текстовий режим.
7. Rotate (90°) clockwise – повернути за годинниковою стрілкою.
8. Rotate (90°) counterclockwise – повернути проти годинниковою стрілкою.
9. X-mirror – відобразити дзеркально відносно осі X.
10. Y-mirror – відобразити дзеркально відносно осі Y.
11. Component/View finder – пошук- вибір компонентів.
12. Device Picker – вибір інструменту.
13. Part Libraries – вибір бібліотеки.
14. List of Items – перелік компонент.

Рис. 2.4. Склад і структура бічної інструментальної панелі середовища ISIS

2.2. Інструментарій Arduino IDE для розробки програмного забезпечення МК-систем на платформі Arduino

Arduino IDE – це універсальний програмний інструментарій, сумісний з операційними системами Windows, Linux та MacOS. Дане середовище, розроблене на базі мов C/C++, забезпечує повний цикл роботи з мікроконтролерами: від написання програмного коду до його інсталяції на плати сімейства Arduino або аналогічні контролери від сторонніх розробників.

Завдяки широкому розповсюдженню екосистеми Arduino, сторонні розробники почали активно інтегрувати власні компілятори та відкриті інструментарії (так звані “ядра”) у цю програмну платформу. Це розширило функціонал середовища, дозволивши розробляти та інсталювати програмний код (скетчі) на мікроконтролери, які першочергово не входили до офіційного переліку підтримуваних пристроїв Arduino.

Архітектура Arduino IDE побудована за модульним принципом, що дозволяє розширювати стандартний функціонал середовища шляхом інтеграції додаткових бібліотек на базі C++. Це забезпечує гнучкість платформи та спрощує взаємодію з різноманітними апаратними компонентами.

Інтегроване середовище розробки Arduino IDE є ефективним інструментарієм, призначеним для оперативного проектування та налагодження програмного забезпечення мікроконтролерних систем на базі архітектури Arduino. Дане ПЗ оптимізує процес написання коду та його подальшої інтеграції в апаратну частину пристрою.

Інтерфейс Arduino IDE об'єднує в собі кілька функціональних зон: текстовий редактор для написання вихідного коду, область сповіщень про стан системи, вікно консольного виводу, а також панель інструментів зі швидким доступом до основних операцій і головне меню. Однією з ключових переваг середовища є спрощений процес прошивки мікроконтролерів: програмний код передається безпосередньо через USB-інтерфейс, що усуває потребу у використанні додаткових апаратних програматорів.

На Рис. 2.10 представлено інтерфейс інтегрованого середовища, призначеного для проектування програмного забезпечення платформ Arduino.

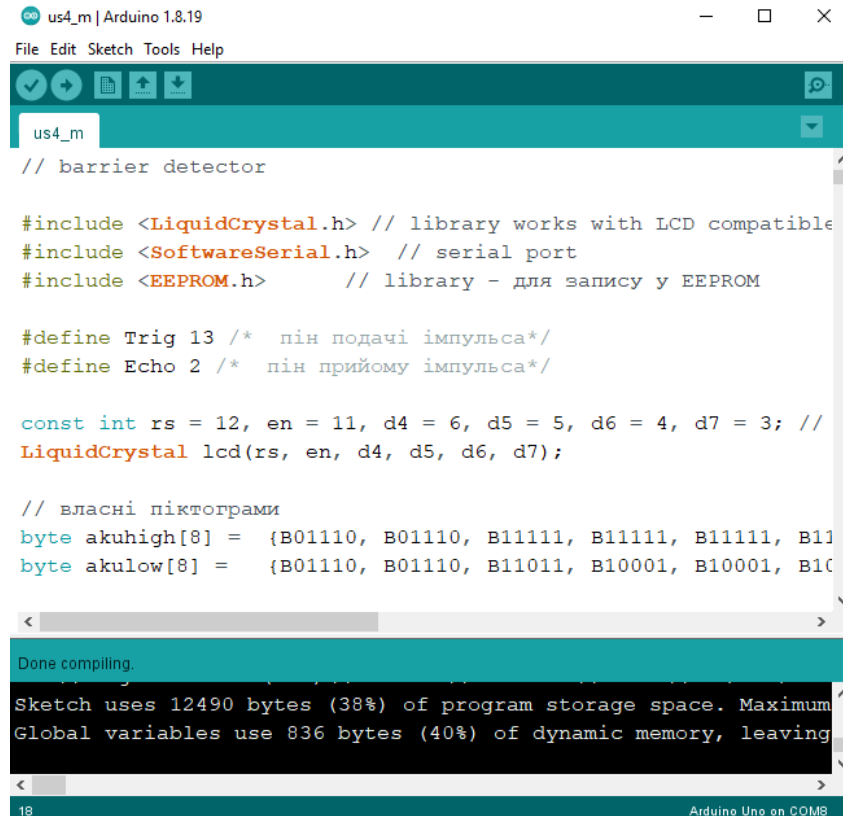


Рис. 2.10. Загальний вигляд інтерфейсу середовища розробки Arduino IDE

Програмні проекти в екосистемі Arduino називаються скетчами. Для їхнього створення передбачено інтегрований редактор, який надає розробнику стандартний набір інструментів для редагування: від операцій з буфером обміну (вирізання/вставка) до функцій навігації та швидкої корекції коду через інструменти пошуку й заміни.

Процес розробки скетчу супроводжується динамічним зворотним зв'язком у спеціалізованих зонах інтерфейсу. Область повідомлень слугує для виведення коротких статусів та ідентифікації помилок у синтаксисі. Більш детальна інформація, що включає розширені звіти про критичні помилки та результати компіляції, відображається в системній консолі.

Панель інструментів містить елементи керування, необхідні для верифікації коду, збірки проекту та менеджменту файлів, зокрема створення, відкриття та збереження скетчів.

Функціональні можливості кнопок панелі інструментів

- Verify/Compile (Перевірка/Компіляція): використовується для аналізу програмного коду на наявність синтаксичних помилок та подальшої його збірки.
- Stop (Зупинка): припиняє активний сеанс обміну даними у вікні послідовного терміналу.
- New (Новий): генерує чистий шаблон для написання нового проєкту (скетчу).
- Open (Відкрити): надає доступ до переліку наявних файлів у вашій робочій папці (блокноті) для їх редагування.
- Save (Зберегти): фіксує поточні зміни в коді, записуючи їх у пам'ять комп'ютера.
- Upload to I/O Board (Завантаження): трансформує вихідний код у машинний формат і передає його безпосередньо в мікроконтролер плати Arduino.
- Serial Monitor (Монітор порту): активує інтерфейс для моніторингу даних, що надходять через віртуальний послідовний порт.

Розширений набір інструментів і налаштувань розподілений за тематичними вкладками: File (Файл), Edit (Правка), Sketch (Скетч), Tools (Інструменти) та Help (Довідка). Важливо враховувати, що інтерфейс є адаптивним: доступність певних опцій у цих меню безпосередньо залежить від поточного стану проєкту та дій, які виконує користувач у середовищі розробки.

Функціонал вкладки Edit включає додаткові опції копіювання:

- Copy for Forum – перенесення коду до буфера обміну у вигляді, придатному для форумних повідомлень.
- Copy as HTML – експорт синтаксису скетчу в HTML-код для подальшої
- Функціональні можливості вкладки Sketch включають:
- Include Library: автоматичне додавання посилання на необхідну бібліотеку через інструкцію `#include`.

- Show Sketch Folder: виклик вікна провідника, що вказує на місце розташування файлів скетчу на диску.
- Add File...: включення додаткових програмних компонентів до проєкту, що призводить до появи нових робочих вкладок у редакторі.

Вкладка Tools призначена для сервісних операцій та конфігурації обладнання:

- Auto Format – інструмент для автоматичної оптимізації та впорядкування вихідного коду.
- Board – вибір цільової апаратної платформи Arduino.
- Serial Port – перелік актуальних портів для передачі даних, який актуалізується щоразу при зверненні до меню.
- Burn Bootloader – запис завантажувача в мікроконтролер. Процедура потребує попередньої перевірки типу плати та налаштування порту в разі роботи через інтерфейс AVR ISP.

У вкладці File особливе місце займає розділ Sketchbook (Блокнот), який функціонує як основне сховище для розроблених програм (скетчів). Доступ до збережених проєктів можна отримати безпосередньо через ієрархію меню File>Sketchbook або скориставшись стандартною кнопкою Open на панелі інструментів.

Середовище Arduino IDE самостійно створює кореневу папку для скетчів при першому відкритті, при цьому зміна локалізації цієї папки доступна через меню конфігурації Preferences. Архітектура програми передбачає багатозадачність інтерфейсу (використання окремих вкладок для різних скетчів). Робочий простір підтримує наступні формати файлів: стандартні файли Arduino; код на мові C (.c); код на мові C++ (.cpp); файли заголовків (.h).

Прошивка МК Arduino. Для успішного завантаження коду в мікроконтролер необхідно попередньо визначити відповідну модель плати та активний послідовний порт у меню інструментів (Tools). Класифікація та опис доступних платформ Arduino наведені нижче.

Для встановлення зв'язку з пристроєм у середовищі Windows використовуються позначення типу COMx. Порти з низькими номерами (1 або 2) зазвичай зарезервовані для стандартного RS-232, тоді як USB-з'єднання отримують вищі порядкові номери. Алгоритм завантаження передбачає вибір активного порту, вказання платформи Arduino та натискання клавіші завантаження (аналог пункту Upload to I/O Board у розділі Sketch).

Архітектура платформ Arduino зазвичай передбачає автоматичне перезавантаження при ініціалізації з'єднання, хоча старі ревізії плат потребують механічного натискання клавіші Reset. Під час завантаження програмного коду в мікроконтролер Arduino Uno можна спостерігати за роботою індикаторів передачі (TX) та прийому (RX) даних — їхнє швидке блимання вказує на коректний перебіг запису.

Використання Bootloader (системного завантажувача) спрощує прошивку мікроконтролера, оскільки ця програма вже міститься в його пам'яті. Завдяки завантажувачу запис нових алгоритмів у МК реалізується без залучення додаткового обладнання, що значно полегшує процес розробки на базі платформ Arduino.

Функціонування Bootloader активується під час перезапуску платформи Arduino або в момент ініціалізації процесу прошивки мікроконтролера. Візуальним індикатором роботи завантажувача є характерне мерехтіння вбудованого світлодіода (зазвичай підключеного до 13-го цифрового виводу).

Бібліотеки. Використання бібліотек дозволяє значно розширити можливості скетчів, особливо в контексті взаємодії з периферійними пристроями та датчиками. Для інтеграції зовнішнього модуля в проєкт слід скористатися командою меню Sketch->Include Library. Після цього у верхній частині програмного коду автоматично з'явиться препроцесорна директива `#include`, що підключає необхідні ресурси.

Оскільки підключення зовнішніх бібліотек призводить до додаткових витрат пам'яті мікроконтролера, рекомендується вилучати з коду всі зайві модулі. Оптимізація скетчу шляхом видалення невикористовуваних ресурсів

дозволяє раціонально використовувати обмежені апаратні можливості платформи Arduino.

Частина стандартних бібліотек інтегрована безпосередньо в дистрибутив Arduino IDE, тоді як специфічні модулі доступні для завантаження з відкритих інтернет-джерел. Процедура ручного встановлення передбачає створення каталогу “libraries” у кореневій директорії блокнота, куди згодом розпаковується архів із файлами відповідної бібліотеки.

Моніторинг послідовної шини забезпечує двосторонній зв'язок із мікроконтролером, дозволяючи відстежувати вхідні дані в режимі реального часу. Відправка команд або даних у порт реалізується шляхом введення символів у рядок запиту з подальшим підтвердженням через Enter. Коригування системних налаштувань редактора, таких як шляхи до файлів чи параметри відображення, виконується у вкладці Preferences.

Програмування МК Arduino. Для зв'язку із зовнішніми компонентами на платах Arduino передбачено набір спеціальних контактів (пінів). Дані порти є реконфігурованими: залежно від завдань проєкту вони можуть працювати в режимі зчитування сигналу (вхід) або його генерації (вихід). Окремо виділена група аналогових входів також має здатність до повної адаптації під цифрові операції вводу/виводу за аналогією до основних портів.

Для конфігурації режимів роботи цифрових виводів використовується стандартна функція `pinMode()`. При роботі з портами вводу критично важливо забезпечити визначений логічний стан, щоб уникнути “плаваючих” значень. Це досягається шляхом інтеграції підтягуючих (pull-up) або стягуючих (pull-down) резисторів, які з'єднують вхід із шиною живлення (+5V) або загальним проводом (GND). Мікроконтролери сімейства ATmega оснащені внутрішніми підтягуючими резисторами з номінальним опором 20 кОм, які можна активувати програмно.

Для програмної конфігурації виводів мікроконтролера застосовуються базові функції середовища розробки. Зокрема, команда `pinMode(pin, INPUT);` ініціалізує обраний пін як вхід для зчитування сигналів. Якщо

необхідно забезпечити стабільний логічний рівень на вході, використовують функцію `digitalWrite(pin, HIGH);`, яка програмно активує внутрішній підтягуючий резистор до шини живлення.

Порти вводу/виводу платформ Arduino розраховані на роботу з низькострумowymi сигналами (до 40 мА). Спроба прямого підключення індуктивних навантажень (двигунів, реле) призведе до перевантаження вихідних каскадів мікроконтролера. Для запобігання виходу ATmega з ладу через надлишковий струм необхідно використовувати проміжні драйвери або транзисторні ключі.

Виходи ШІМ (PWM). Функціонування виходів з підтримкою ШІМ (широтно-імпульсної модуляції) ґрунтується на генерації послідовності імпульсів різної тривалості. Це дозволяє програмно регулювати середнє значення вихідної напруги в діапазоні від 0 В до 5 В. Використання функції `analogWrite()` передбачає 8-бітне кодування (0–255 дискретних значень), де параметр 255 відповідає повному заповненню періоду (100%, постійні 5 В), а значення 127 формує сигнал із коефіцієнтом заповнення 50%, що еквівалентно середній напрузі 2,5В.

Аналогові входи. Мікроконтролери сімейства ATmega, що інтегровані в платформи Arduino, оснащені шестиканальним аналого-цифровим перетворювачем (АЦП). Роздільна здатність даного модуля становить 10 біт, що забезпечує отримання цифрового значення в діапазоні від 0 до 1023. Характерною особливістю аналогових входів є їхня здатність до переконфігурації в режим стандартних цифрових портів із нумерацією від 14 до 19. Наприклад, ініціалізація виводу як цифрового виходу та подача на нього логічної одиниці реалізується командами `pinMode(14, OUTPUT);` та `digitalWrite(14, HIGH)`.

Структура програми на Arduino. Розробка програмного забезпечення в середовищі Arduino IDE здійснюється на високорівневій мові Wiring. Проте варто зауважити, що окремого компілятора для неї не існує; натомість код проходить процес препроцесорної обробки, під час якого він трансформується

у повноцінну програму на мові C/C++. Подальша збірка виконується за допомогою компілятора AVR-GCC. Сама оболонка Arduino IDE є спеціалізованим інструментарієм, оптимізованим виключно для взаємодії з апаратними рішеннями екосистеми Arduino.

Апаратні платформи Arduino. Вибір конкретної апаратної платформи безпосередньо визначає ключові параметри системи: продуктивність виконання програмних алгоритмів, інтенсивність обміну даними, а також доступний обсяг пам'яті для розміщення скетчу. Крім того, тип обраної плати диктує специфіку конфігурації системного завантажувача (Bootloader) мікроконтролера. Нижче наведено характеристики основних апаратних рішень екосистеми Arduino.

- Апаратна архітектура Arduino Nano та Duemilanove базується на мікроконтролері ATmega328, що працює на тактовій частоті 16 МГц. Аналогічна конфігурація обчислювального ядра застосовується також у моделях Arduino Pro та Pro Mini. Варто відзначити еволюцію апаратної бази: рання платформа Diecimila, а також перші ревізії Duemilanove будувалися на основі чипа Atmel ATmega168, тоді як у найбільш ранніх версіях систем використовувався мікроконтролер ATmega8.
- Arduino Mega (ATmega1280): високопродуктивна платформа з тактовою частотою 16 МГц. Відрізняється від стандартних моделей використанням мікроконтролера з розширеною периферією та обсягом пам'яті 128 КБ.
- LilyPad Arduino (ATmega328): низьковольтна версія платформи (3,3 В) із тактовою частотою 8 МГц. Оснащена схемою автоматичного перезавантаження. Завдяки специфічній формі та малій вазі орієнтована на створення “розумного” одягу та гнучких електронних модулів.

На Рис. 2.11 представлена архітектура апаратної платформи Arduino Uno, серцем якої є мікроконтролер ATmega328. Даний обчислювальний модуль функціонує на тактовій частоті 16 МГц при номінальній напрузі

живлення 5 В. Ця конфігурація є базовим стандартом для розробки прототипів вбудованих систем керування.

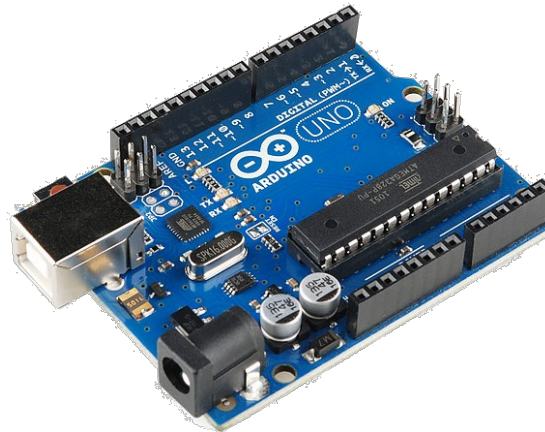


Рис. 2.11. Конструктивне виконання апаратної платформи Arduino Uno

Перед початком роботи з Arduino Uno через USB-інтерфейс необхідно підтвердити правильність налаштувань у середовищі розробки. Для цього в меню Tools (Інструменти) у розділі Board (Плата) слід обрати відповідний тип контролера – Arduino/Genuino Uno (Рис.2.12).

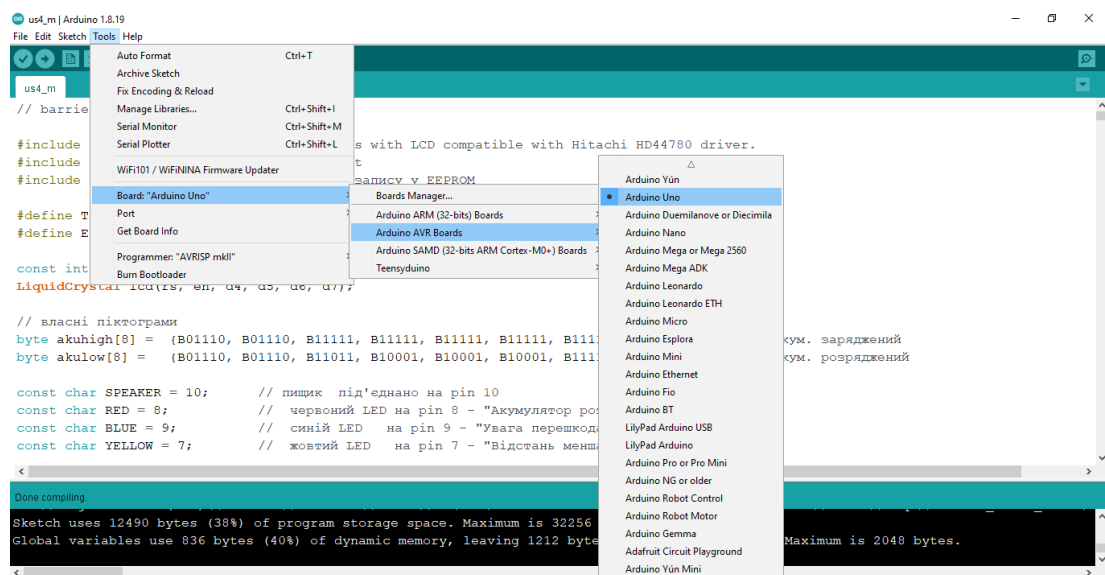


Рис. 2.12. Процедура ідентифікації цільової апаратної платформи в середовищі розробки

Для встановлення зв'язку з пристроєм необхідно в меню інструментів (Tools) перейти до розділу портів (Port) і визначити відповідний активний інтерфейс підключення.

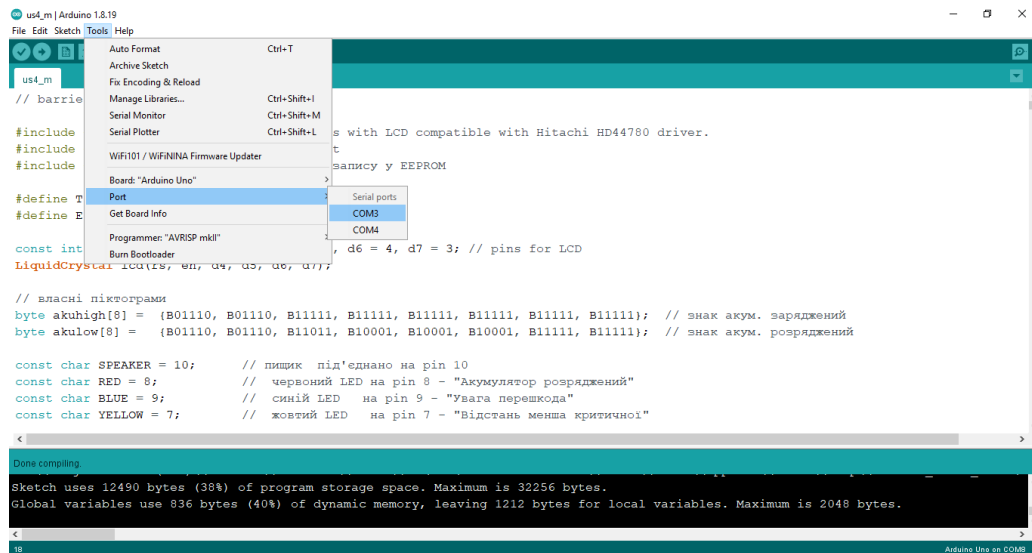


Рис. 2.13. Процедура ідентифікації та вибору послідовного порту для комунікації з Arduino Uno

Процес завантаження скомпільованого коду в пам'ять мікроконтролера ATmega328P здійснюється через меню Sketch->Upload. Під час передачі даних на платі Arduino Uno спостерігається характерна світлова індикація (миготіння світлодіодів TX/RX), що підтверджує активний обмін інформацією.

РОЗДІЛ 3

АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО ПРИСТРОЮ ДЛЯ УНИКНЕННЯ ДРОНОМ ПЕРЕШКОД В ЗАМКНУТИХ ТА ЗАДИМЛЕНИХ ПРИМІЩЕННЯХ

3.1. Визначення апаратних засобів для проєктування цифрового пристрою для уникнення перешкод

Для розробки цифрового пристрою для виявлення перешкод дроном в умовах обмеженого простору та задимленості визначено оптимальний склад апаратних засобів. До обраної елементної бази увійшли:

- Контролер Arduino Uno на базі восьмибітного МК ATmega328P;
- Акустичний далекомір (ультразвуковий датчик) моделі HC-SR04;
- Символьний РК-індикатор WH1602B (конфігурація 16x2);
- Допоміжні компоненти: кнопки, перемикачі, пасивні елементи, а також модулі світлової та звукової сигналізації (світлодіоди та зумер).

Опис Arduino Uno з МК ATmega328P. Апаратна реалізація цифрового пристрою для маневрування дрона в задимлених локаціях ґрунтується на платформі Arduino Uno. Використання МК ATmega328P забезпечує необхідну продуктивність системи за умови низької собівартості компонентів, що робить проєктований пристрій економічно вигідним та технічно надійним. Зовнішній вигляд Arduino Uno разом із специфікацією її інтерфейсних роз'ємів представлено на Рис. 3.1.

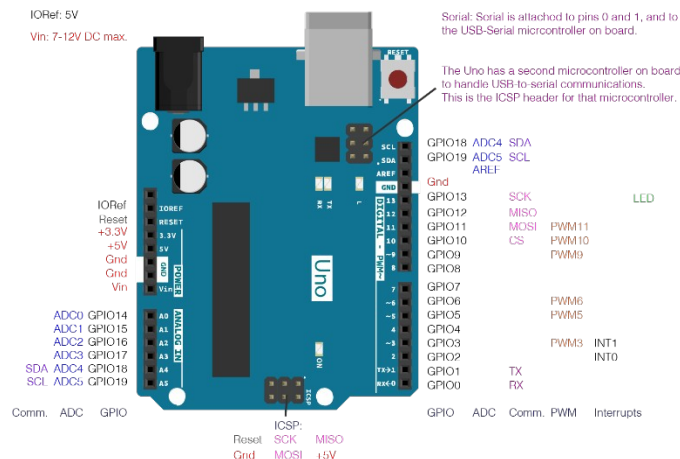


Рис. 3.1. Плата Arduino Uno

Аналіз схеми роз'ємів показує, що інтерфейс плати забезпечує доступ як до ліній живлення, так і до функціональних виводів мікроконтролера ATmega328P.

Апаратні можливості обраної платформи включають 14 цифрових каналів вводу-виводу, шість із яких підтримують режим широтно-імпульсної модуляції (ШИМ), та 6 аналогових входних портів. Тактування системи забезпечується кварцовим резонатором із частотою 16 МГц. Конструкція пристрою передбачає наявність інтерфейсу USB, силового роз'єму, порту ICSP та кнопки апаратного скидання (Reset). Функціонування модуля можливе як за умови USB-з'єднання з ПК, так і при підключенні автономного джерела живлення (акумулятора або адаптера) з напругою в діапазоні 7–12 В.

Таблиця 3.1. Основні параметри контролера Arduino Uno

Параметр	Значення та опис
Обчислювальне ядро	Мікроконтролер ATmega328P (архітектура AVR)
Робоча напруга	5 В (логічні рівні)
Зовнішнє живлення	Рекомендований діапазон 7–12 В
Навантажувальна здатність	Максимальний струм на лініях вводу-виводу – 40 мА
Енергонезалежна пам'ять	32 КБ Flash (із них 0,5 КБ зарезервовано під завантажувач)
Оперативна пам'ять (SRAM)	2 КБ для зберігання тимчасових даних
Пам'ять EEPROM	1 КБ для конфігураційних параметрів
Продуктивність	Тактова частота зовнішнього кварцу – 16 МГц

Енергозабезпечення Arduino Uno може здійснюватися через інтерфейс USB або від автономного зовнішнього джерела, причому вибір активного каналу живлення відбувається в автоматичному режимі. Попри те, що допустимий діапазон входної напруги становить від 6 В до 20 В, експлуатація пристрою при напрузі < 7 В не рекомендується через можливу нестабільність роботи контролера. Водночас перевищення порогу у 12 В може призвести до критичного перегріву вбудованого стабілізатора та виходу плати з ладу. Таким чином, найбільш сприятливим для стабільної роботи є діапазон від 7...12 В.

Таблиця 3.2. Характеристика ліній живлення Arduino Uno

Вивід	Функціональне призначення	Джерело формування
VIN	Вхід для стороннього живлення	Зовнішній адаптер або АКБ
5V	Основна лінія живлення логіки	Регулятор напруги (від VIN) або USB
3V3	Допоміжна низьковольтна лінія	Вбудований LDO-стабілізатор (max 50 мА)
GND	Загальний провід ("земля")	Спільний мінус схеми

Конфігурація цифрових портів Arduino Uno. Платформа оснащена 14 цифровими контактами, кожен з яких є універсальним і може функціонувати як у режимі введення, так і виведення сигналу. Програмне керування станом цих пінів здійснюється за допомогою стандартних функцій середовища Arduino IDE: `pinMode()`, `digitalWrite()` та `digitalRead()`.

Функціональні особливості виводів Arduino Uno. Окрім загального призначення, певні піни платформи мають специфічні апаратні функції для реалізації протоколів зв'язку та обробки сигналів:

- Послідовний інтерфейс (UART): Піни 0 (RX) та 1 (TX) відповідають за приймання та передачу даних за стандартом TTL. Вони апаратно з'єднані з мікросхемою ATmega8U2 (або 16U2), що забезпечує міст USB-to-Serial для зв'язку з комп'ютером.
- Зовнішні переривання: Контакти 2 та 3 підтримують функцію `attachInterrupt()`. Це дозволяє мікроконтролеру миттєво реагувати на зміну стану сигналу (фронт, спад або зміна рівня), що критично для роботи з енкодерами або датчиками швидкості.
- Широтно-імпульсна модуляція (ШИМ/PWM): Виводи 3, 5, 6, 9, 10 та 11 підтримують 8-бітну ШІМ через функцію `analogWrite()`. Це дозволяє керувати яскравістю світлодіодів або швидкістю обертання двигунів.
- Інтерфейс SPI: Для швидкісного обміну даними з периферією використовуються піни 10 (SS), 11 (MOSI), 12 (MISO) та 13 (SCK) за підтримки стандартної бібліотеки `SPI.h`.
- Інтерфейс I2C (TWI): Реалізується через контакти A4 (SDA) та A5 (SCL). Для роботи з цим протоколом у середовищі розробки передбачена бібліотека `Wire.h`.
- Аналого-цифрове перетворення (ADC): Платформа має 6 каналів АЦП (A0–A5) з роздільною здатністю 10 біт (1024 дискретних значення). За замовчуванням діапазон вимірювання становить 0–5 В, проте верхню

межу можна коригувати через пін AREF та функцію `analogReference()`.

- Індикація (LED): До виводу 13 підключено інтегрований світлодіод. Він активується при подачі на порт високого лог. рівня (HIGH).
- Службові виводи: AREF: опорна напруга для налаштування точності аналогових вимірювань. Reset: подача низького рівня (LOW) на цей вхід ініціює апаратне перезавантаження мікроконтролера.

На Рис. 3.2 наведено схемотехнічне рішення плати Arduino Uno із зазначенням зв'язків керуючого контролера ATmega328P.

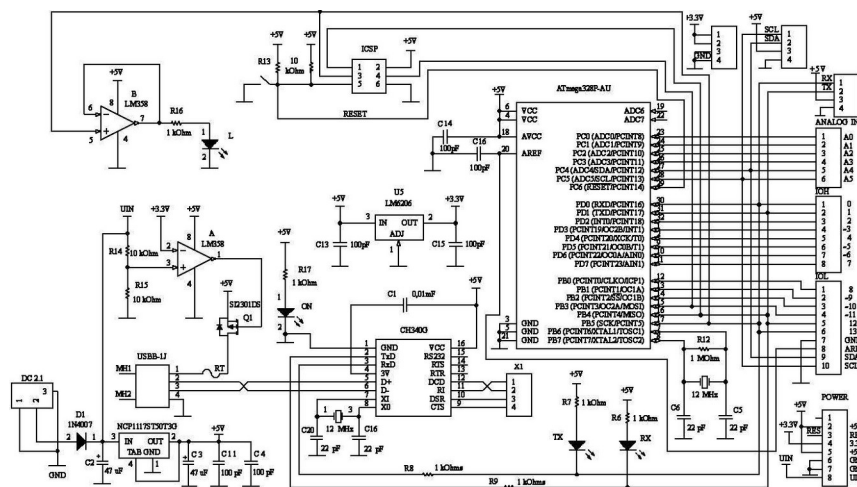


Рис. 3.2. Схемотехнічна реалізація платформи Arduino Uno на базі мікроконтролера ATmega328P

Обмін інформацією. Платформа Arduino Uno оснащена набором комунікаційних засобів для взаємодії з ПК, іншими модулями Arduino чи сторонніми контролерами. Основний мікроконтролер ATmega328P реалізує послідовну передачу даних за стандартом UART TTL (рівень напруги 5 В), використовуючи для цього виводи 0 (RX) та 1 (TX). Апаратну трансляцію цього інтерфейсу в USB-протокол забезпечує допоміжний чип ATmega8U2. Завдяки цьому програмне забезпечення на стороні комп'ютера розпізнає плату як віртуальний послідовний порт (COM-порт).

Програмне забезпечення мікросхеми ATmega8U2 базується на стандартних драйверах класу USB CDC, що позбавляє потреби у використанні стороннього ПЗ. Для належної ідентифікації пристрою в середовищі Windows

необхідно лише наявність інформаційного файлу `ArduinoUNO.inf`. Для візуалізації та аналізу обміну даними у середовищі розробки передбачено інструмент “Монітор порту” (Serial Monitor), який забезпечує двосторонню передачу текстової інформації між ПК та платформою.

Візуальний контроль обміну даними здійснюється за допомогою світлодіодних індикаторів RX та TX, які активуються під час трансляції сигналів через USB-з’єднання. Крім апаратного інтерфейсу, платформа дозволяє організувати передачу даних через довільні цифрові порти. Це реалізується на програмному рівні за допомогою спеціалізованої бібліотеки `SoftwareSerial`, яка емулює роботу послідовного порту на вибраних виводах.

Розробка та завантаження програмного коду здійснюється через спеціалізоване середовище Arduino IDE. У налаштуваннях (меню Tools->Board) необхідно вказати відповідну модель – “Arduino Uno”. Окрім стандартного методу, передбачена можливість прямого програмування мікроконтролера ATmega328P через інтерфейс ICSP, що дозволяє оминати використання вбудованого завантажувача (bootloader) та вивільнити додатковий об’єм пам’яті.

Опис МК AVR ATmega328P. Основою платформи є 8-бітний мікроконтролер ATmega328P, який поєднує низьке енергоспоживання з ефективною RISC-архітектурою. Основні технічні характеристики МК ATmega328 наведено у Табл.3.3.

Таблиця 3.3. Технічні характеристики мікроконтролера ATmega328P

Параметр	Значення/Опис
Архітектура	8-бітна RISC (131 інструкція)
Продуктивність	До 16 MIPS (при 16 МГц)
Пам’ять (Flash/SRAM/EEPROM)	32 КБ/2 КБ/1 КБ
Таймери	2×8-біт, 1×16-біт, Watchdog
Аналогові модулі	8-канальний 10-біт АЦП, компаратор
Інтерфейси	UART, SPI, I2C (TWI)
Напруга живлення	2,7 В ... 5,5 В
Типи корпусів	PDIP-28, TQFP-32

Конфігурація та призначення виводів мікроконтролера ATmega328P для корпусів типів PDIP та TQFP продемонстровані на Рис. 3.3.

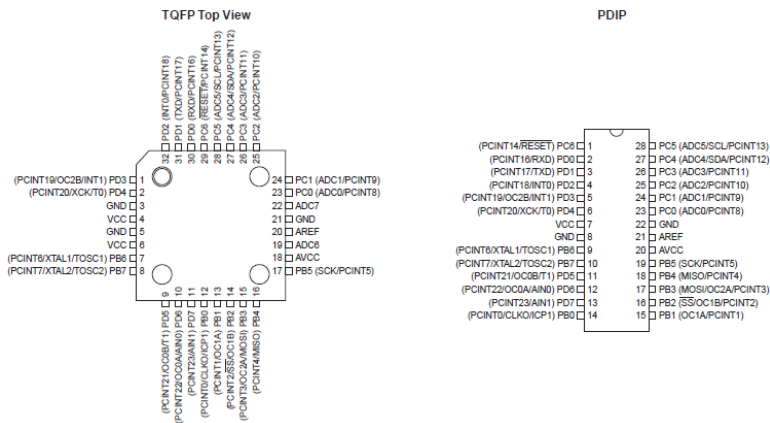


Рис. 3.3. Схематичне представлення виводів ATmega328P для різних типів конструктивного виконання (TQFP та PDIP)

Внутрішня архітектура та взаємозв'язки функціональних блоків мікроконтролера ATmega328P представлені у вигляді блок-схеми на Рис. 3.4.

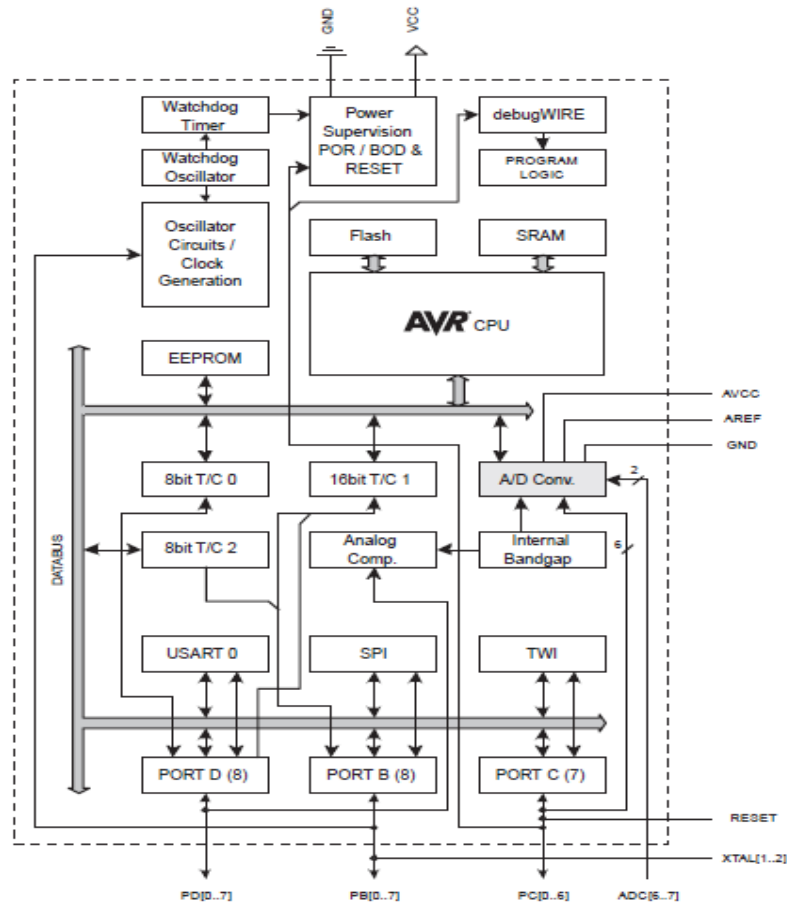


Рис. 3.4. Структурна схема внутрішньої архітектури МК ATmega328P

Завдяки використанню RISC-архітектури, обчислювальна потужність ATmega328P на порядок перевищує показники швидкодії аналогічних мікроконтролерів, побудованих на базі CISC-архітектури.

Для оновлення програмного коду у Flash-пам'яті передбачено два основні методи: використання послідовного периферійного інтерфейсу (SPI) або запис через попередньо встановлений завантажувач (bootloader).

Висока продуктивність мікроконтролера ATmega328P досягається завдяки синергії вдосконаленої 8-бітної RISC-архітектури процесора та інтегрованої Flash-пам'яті, що дозволяє оптимізувати виконання команд.

Опис ультразвукового модуля HC-SR04. Функціонування ультразвукового датчика HC-SR04 базується на методі ехолокації. Пристрій генерує акустичні імпульси та реєструє відлуння, що повертається після зіткнення з об'єктом. Дистанція до перешкоди визначається шляхом вимірювання часового інтервалу, необхідного для проходження хвилі в обох напрямках. Конструктивно модуль складається з електронного блока керування та двох п'єзоелементів, один з яких виконує роль передавача, а інший – детектора сигналу.

Зовнішній вигляд ультразвукового модуля HC-SR04 разом із вбудованими елементами керування представлено на Рис. 3.13.

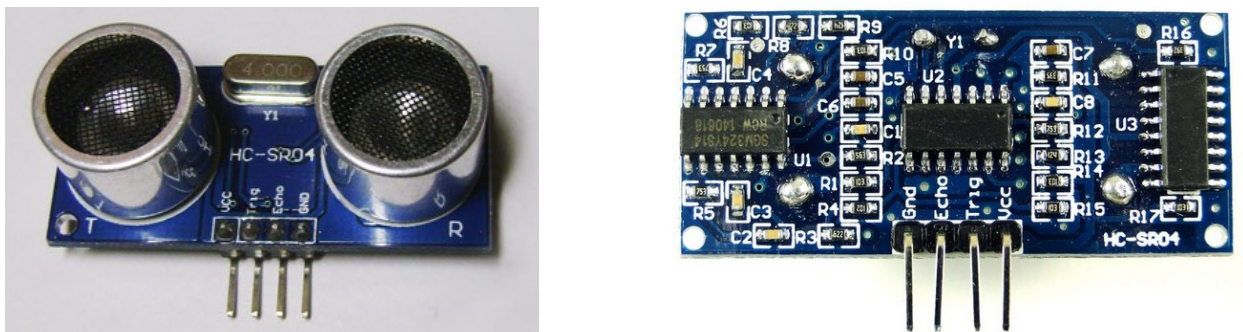


Рис. 3.13. Зовнішній вигляд та конструкція модуля HC-SR04

У Табл.3.4 наведено основні технічні параметри модуля HC-SR04.

Таблица 3.4. Технічні параметри модуля HC-SR04

Параметр	Значення
Робоча напруга (DC)	5 В (діапазон 4,8...5,0 В)
Струм під час роботи	15 мА

Вимірювальна дистанція	20–4000 мм
Роздільна здатність (точність)	± 3 мм
Температурний режим	0...60 °C (похибка $\pm 10\%$)
Габарити модуля	42x20x15 мм

Взаємодія модуля з мікроконтролером реалізується через чотириконтактний інтерфейс. Група живлення включає виводи VCC та GND (номінальна напруга 5 В), тоді як передача даних та керування здійснюються за допомогою сигнальних ліній TRIG і ECHO (Рис. 3.14).



Рис. 3.14. Конфігурація та функціональне призначення виводів ультразвукового модуля HC-SR04

Вхід TRIG слугує для ініціації роботи сенсора. Процес вимірювання активується шляхом подачі на цей пін короткого імпульсу високого рівня тривалістю 10–15 мкс. Після цього модуль генерує пачку з восьми ультразвукових імпульсів на частоті 40 кГц. Одночасно з початком випромінювання на виході ЕЧНО встановлюється логічна одиниця. Стан цього виходу змінюється на логічний нуль лише тоді, коли приймач зафіксує відбиту від об'єкта хвилю. Саме часовий інтервал, протягом якого на ЕЧНО утримувався високий рівень, використовується як базовий параметр для розрахунку дистанції.

Часовий інтервал сигналу на виході ЕЧНО прямо пропорційний дистанції до об'єкта, від якого відбилася акустична хвиля. Мікроконтролер фіксує тривалість цього імпульсу для подальшого розрахунку відстані. Зазвичай цей показник не перевищує 25 мс, проте у випадку відсутності перешкоди в зоні досяжності сенсора, тривалість сигналу становить близько 36 мс. Математична модель для визначення дистанції наведена у формулі (3.1):

$$S = \frac{t \cdot v}{2} \quad (3.1)$$

де: S — шукана відстань; t — виміряна тривалість імпульсу; v — швидкість поширення звуку в повітрі (приймається за 340 м/с).

Функціонування модуля базується на методі ехолокації та виконується за наступною схемою:

1. Запуск вимірювання: Мікроконтролер генерує на вході TRIG (Trigger) ініціювальний імпульс високого рівня тривалістю щонайменше 10 мкс.
2. Акустичне випромінювання: Отримавши стартовий сигнал, датчик автоматично транслює у простір пачку з восьми ультразвукових коливань на частоті 40 кГц.
3. Поширення та відбиття: Ультразвукові хвилі досягають об'єкта, відбиваються від його поверхні та повертаються до приймального елемента сенсора.
4. Реєстрація ехо-сигналу: На виході ECHO формується логічний рівень, тривалість якого еквівалентна часу, за який звук подолав шлях до перешкоди та назад.
5. Обчислення результату: МК фіксує час затримки сигналу на виході ECHO та, враховуючи швидкість звуку, визначає точну відстань до цілі (Рис. 3.15).

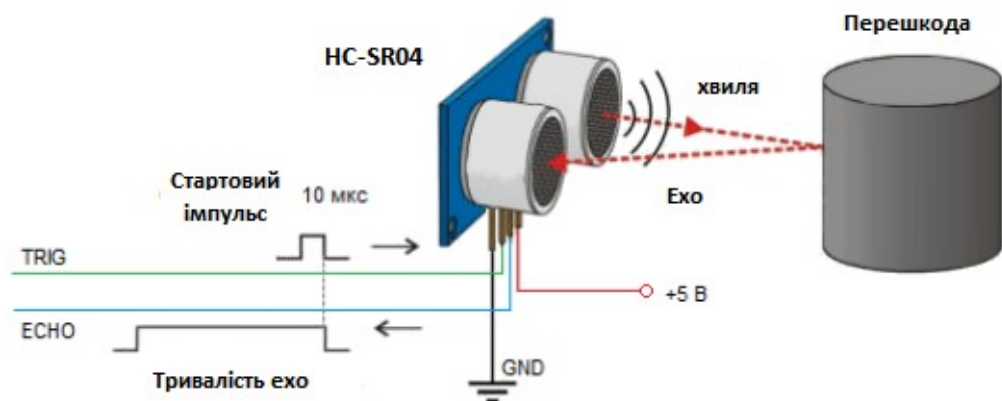


Рис.3.15. Схема функціонування ультразвукового модуля HC-SR04

Алгоритм функціонування модуля HC-SR04 наочно представлений на часовій діаграмі (Рис. 3.16).

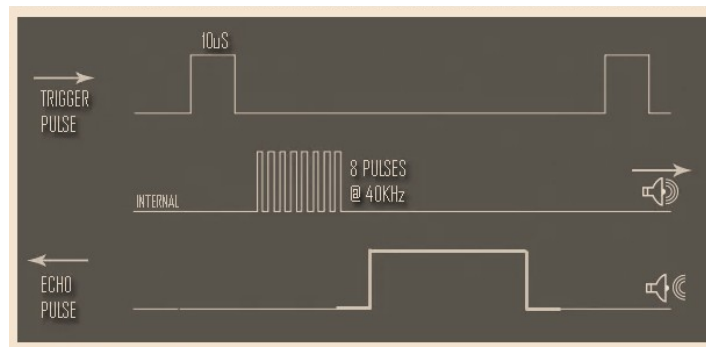


Рис. 3.16. Часові інтервали та послідовність сигналів модуля HC-SR04

Ефективне відбиття сигналу від об'єктів забезпечується випромінюванням ультразвукових хвиль на частоті 40 кГц. Дану функцію виконує спеціалізований п'єзокерамічний елемент (Рис. 3.17), конструкція якого оптимізована для генерації коливань високої частоти.

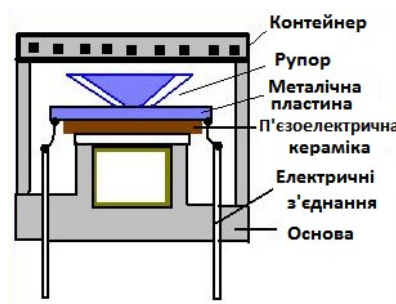


Рис. 3.17. Внутрішня будова та основні компоненти ультразвукових п'єзoeлектричних перетворювачів

Конструктивне виконання випромінювача забезпечує формування вузьконаправленого акустичного потоку замість ізотропного поширення хвиль. Специфіку просторового розподілу сигналу для стандартного ультразвукового сенсора ілюструє діаграма спрямованості (Рис. 3.18).

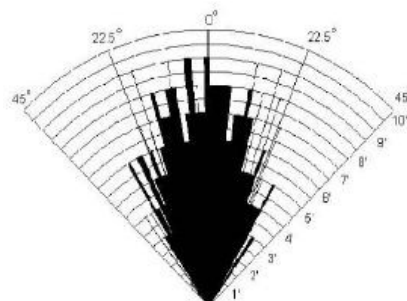


Рис. 3.18. Характеристика просторової спрямованості випромінювання ультразвукового сенсора

Згідно з наведеною характеристикою, сектор огляду стандартних ультразвукових сенсорів зазвичай коливається в межах 50–60°. Водночас для

модуля HC-SR04 цей показник є значно меншим і становить лише 15° , що свідчить про його високу спрямованість. Завдяки широкій зоні розсіювання хвилі, ультразвуковий датчик здатний розпізнавати малогабаритні об'єкти (наприклад, ніжку крісла), які лазерний далекомір може пропустити через надмірну вузькість променя. Поряд із цією перевагою, варто виділити два ключові недоліки, притаманні ультразвуковим системам вимірювання.

По-перше, матеріали з високим коефіцієнтом пористості інтенсивно поглинають акустичну енергію. Це перешкоджає формуванню зворотного ехо-сигналу, через що сенсор стає нездатним коректно визначити дистанцію до такої поверхні. Другим суттєвим обмеженням виступає відносно низька швидкість поширення акустичних хвиль у повітряному середовищі. Це створює часову затримку між випромінюванням та прийомом сигналу, що лімітує максимально можливу частоту оновлення даних про відстань.

Розглянемо приклад: при виявленні об'єкта на відстані 4 метри сумарний час проходження сигналу в обох напрямках становить близько 24 мс. Цей часовий лаг є критичним параметром при інтеграції далекоміра в системи керування безпілотними літальними апаратами, оскільки він обмежує швидкість реакції робота. Електрична структура, що забезпечує роботу модуля HC-SR04, представлена на принциповій схемі (Рис. 3.19).

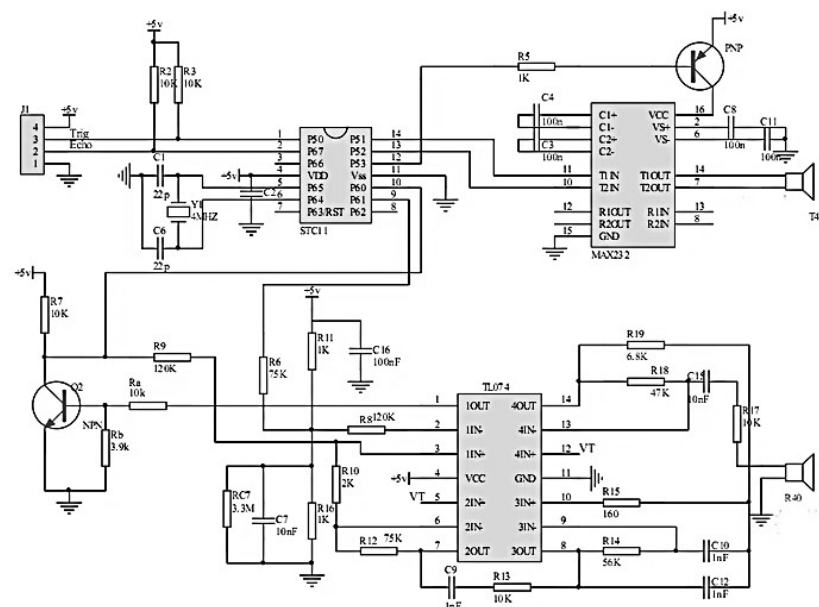


Рис. 3.19. Схема електрична принципова ультразвукового модуля HC-SR04

Огляд рідкокристалічного дисплея з контролером HD44780.

Рідкокристалічні індикатори (РКІ) є поширеним засобом візуалізації символічних даних. Завдяки низькому рівню енергоспоживання та наявності вбудованого LED-підсвічування, дані пристрої забезпечують чітке зчитування інформації в умовах недостатньої видимості або повної темряви. Експлуатаційні характеристики модулів дозволяють стабільно працювати в широкому температурному діапазоні від -20°C до $+70^{\circ}\text{C}$.

Структура модуля базується на керуючому чипі HD44780 та рідкокристалічній матриці. Зовнішня конфігурація дисплея представлена на Рис. 3.20. Пристрій забезпечує виведення інформації у два рядки, кожний з яких містить до 16 знакомісць. Формування символів здійснюється за допомогою матриці розміром 5×8 пікселів, причому для чіткості візуалізації між сусідніми знаками передбачено технологічний проміжок завширшки в одну точку.

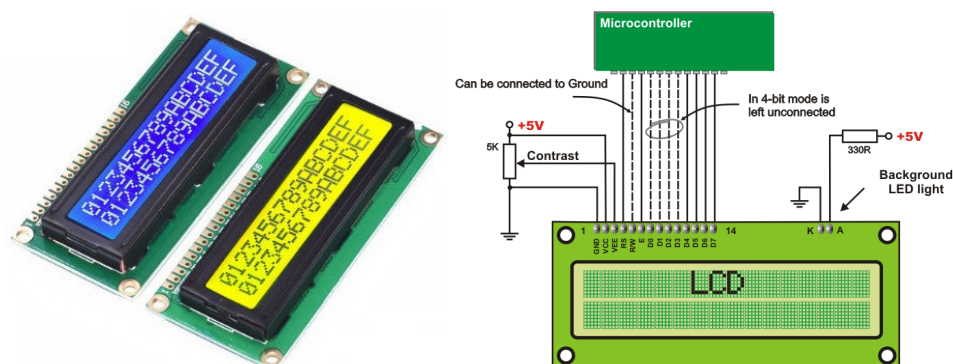


Рис. 3.20. Зовнішній вигляд символічного РК-дисплея 1602 та схема його електричного підключення

Візуалізація кожного знака на рідкокристалічній матриці базується на відповідності його коду конкретній адресі у внутрішній пам'яті контролера. Архітектура модуля передбачає розділення пам'яті на два типи: область збереження стандартних символів (знакогенератор) та сегмент для програмування користувацьких знаків, що координується вбудованою логікою керування.

Вибраний дисплей характеризується розширеним набором функцій, серед яких варто виділити підтримку кириличного алфавіту завдяки наявності

двох сторінок знакогенератора. Гнучкість підключення забезпечується вибором розрядності шини даних (4 або 8 біт). Архітектура контролера дозволяє не лише записувати інформацію в ОЗП, а й зчитувати її назад на шину. Важливою перевагою є можливість створення 8 власних графічних об'єктів. Візуальний комфорт досягається за допомогою вибору типу курсора та регулювання параметрів яскравості й контрасту екрана.

Структура та логіка контролера HD44780. Управління дисплейним модулем реалізовано через систему команд, що надходять до вбудованого контролера. Основними вузлами обробки є регістр даних (DR) та інструкцій (IR). Для розмежування потоків інформації використовується лінія вибору регістра RS. При подачі низького рівня сигналу на цей вхід система адресує регістр інструкцій, а при високому рівні — перемикається на роботу з оперативною пам'яттю через регістр даних.

Регістр даних (DR) забезпечує запис інформації у відеопам'ять дисплея (DDRAM) або в область пам'яті знакогенератора (CGRAM). Конкретна комірка пам'яті для запису визначається поточним значенням лічильника адреси (AC). Паралельно з цим, вміст регістра інструкцій (IR) надходить до дешифратора, де відбувається інтерпретація та виконання відповідних мікрокоманд керування модулем.

Внутрішня відеопам'ять дисплея має місткість 80 байт і призначена для утримання кодів символів. Логічно DDRAM розділена на дві незалежні стрічки по 40 байт. Така уніфікована модель адресації дозволяє контролеру працювати з даними як з двома рядками по 40 знаків, що забезпечує сумісність з РК-модулями різної геометрії та роздільної здатності.

Архітектурно чип HD44780 обмежений 40 виходами сегментного керування. Це накладає ліміт на фізичну довжину дисплея, дозволяючи драйверу самостійно обслуговувати лише 8-символьні РК-індикатори без використання додаткових розширювачів.

Конструкція дисплеїв форматом до 8x2 знакомісць базується на одному контролері HD44780. Однак для панелей із більшою інформаційною місткістю

архітектура доповнюється каскадом драйверів розширення. Зокрема, мікросхема HD44100 забезпечує керування додатковими 40 сегментами матриці. При цьому стан пристрою та поточні режими його функціонування визначаються системою внутрішніх прапорців контролера HD44780.

Підключення РК-дисплея. Програмно-апаратна реалізація інтерфейсу РК-дисплея. Для взаємодії з модулями на базі контролера Hitachi HD44780 середовище розробки CodeVisionAVR надає набір стандартних високорівневих бібліотек. Це дозволяє абстрагуватися від низькорівневих протоколів та спрощує процес виведення даних. Апаратна конфігурація з'єднань між мікроконтролером (МК) та РК-панеллю детально представлена на Рис. 3.21.

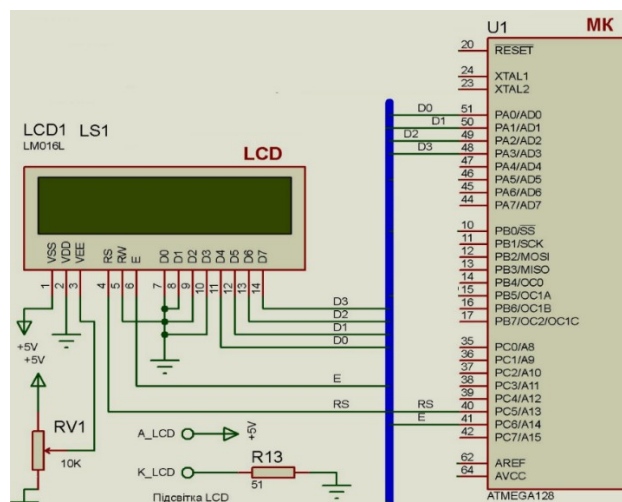


Рис. 3.21. Схема з'єднань дисплейного модуля з МК: конфігурація шини даних та ліній керування

Регулювання чіткості відображення символів на рідкокристалічній матриці здійснюється за допомогою змінного резистора RV1, підключеного до виводу керування контрастністю. Живлення внутрішньої системи підсвічування реалізовано через виводи A_LCD (анод) та K_LCD (катод). Для обміну інформацією між мікроконтролером та дисплеєм використано 4-бітний інтерфейс, задіявши лінії даних D4...D7.

3.2. Проєктування цифрового пристрою для уникнення дроном перешкод в замкнутих та задимлених приміщеннях в САПР Proteus VSM

Цифровий пристрій для уникнення дроном перешкод в умовах обмеженого простору та низької видимості (задимлення) реалізує комплексний алгоритм моніторингу оточення. Пристрій забезпечує ідентифікацію об'єктів та обчислення дистанції до них з подальшою обробкою та трансляцією даних через послідовний інтерфейс. У разі скорочення відстані до критичної межі активується багаторівнева система оповіщення (звукова та світлова). Функціонал приладу дозволяє користувачеві гнучко корегувати робочі параметри й одиниці вимірювання через інтерактивне меню, причому всі зміни фіксуються в енергонезалежній пам'яті (EEPROM). Додатково реалізовано підсистему діагностики стану джерела живлення.

Конструктивні особливості цифрового детектора, побудованого на базі мікроконтролерної платформи Arduino та ультразвукового сенсора HC-SR04, представлені на Рис. 3.22. Дана система є комплексною розробкою, що поєднує в собі апаратну частину та відповідне програмне забезпечення.

Технічна реалізація детектора перешкод включає обчислювальне ядро на базі Arduino Uno, до якого під'єднано периферійні пристрої: ультразвуковий далекомір HC-SR04, модуль візуальної індикації (РКД) та систему світло-звукового попередження. Користувацький інтерфейс реалізовано за допомогою набору тактових кнопок (КН1–КН4), які забезпечують взаємодію з меню налаштувань.

Обчислювальним ядром системи виступає платформа Arduino Uno на базі мікроконтролера ATmega328 (архітектура AVR). Вона здійснює поточну обробку сигналів від ультразвукового сенсора HC-SR04, забезпечує візуалізацію результатів на рідкокристалічному індикаторі та трансляцію даних через послідовний інтерфейс (UART). Програмний алгоритм контролера відповідає за активацію тривожних сигналів, збереження користувацьких конфігурацій у внутрішній енергонезалежній пам'яті (EEPROM) та безперервний моніторинг стану акумуляторної батареї.

На Рис. 3.22 представлено архітектуру побудови цифрового пристрою для детектування перешкод. В основу структурної схеми покладено мікроконтролерну платформу Arduino Uno, що взаємодіє з ультразвуковим сенсором моделі HC-SR04.

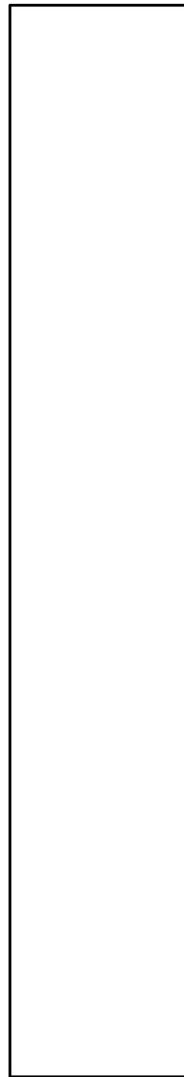


Рис. 3.22. Функціональна організація цифрового ультразвукового детектора перешкод на базі МК ATmega328

На Рис. 3.23 відображено склад апаратної частини детектора перешкод, де ключовими елементами виступають обчислювальна платформа Arduino та вимірювальний модуль HC-SR04.

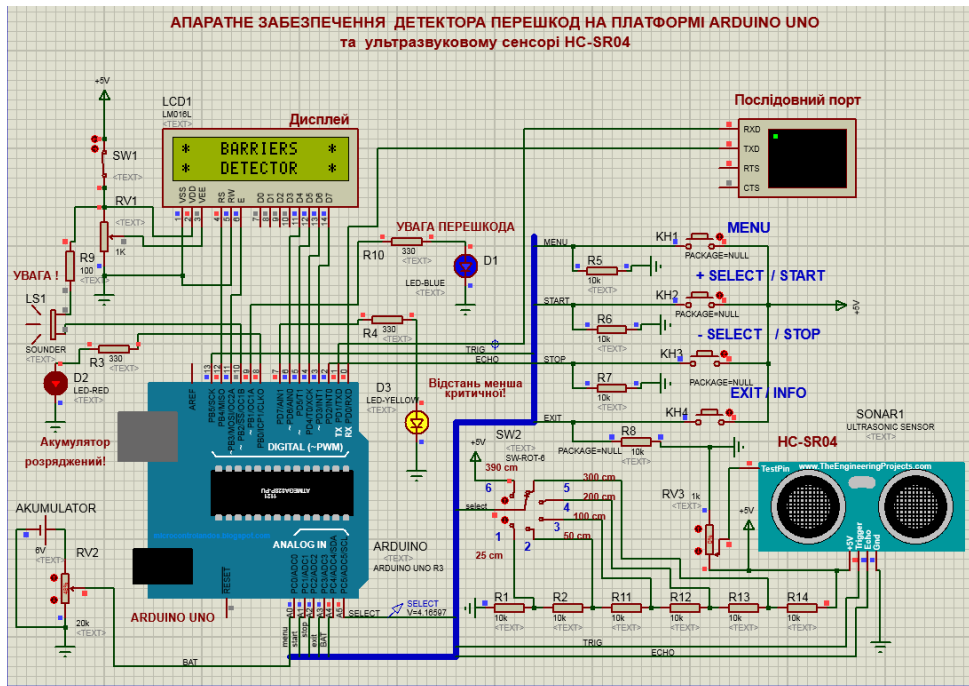


Рис. 3.23. Апаратна реалізація цифрового детектора перешкод на мікроконтролерній платформі Arduino та модулі HC-SR04

Алгоритм функціонування детектора передбачає постійний моніторинг дистанції до перешкод та їх порівняння з заданою межею спрацювання. Процес супроводжується динамічним оновленням даних на РКД, а також паралельною передачею інформаційних пакетів через COM-порт.

Програмна логіка детектора забезпечує безперервну оцінку відстані. При порушенні встановленого безпечного діапазону ініціюється попереджувальний сигнал: світлодіод D3 переходить у режим блимання, а п'єзовипромінювач LS1 генерує звукове сповіщення. Стан детектування об'єкта візуалізується за допомогою синього світлодіода D1, при цьому інформація про тривожну подію автоматично надсилається у послідовний порт для реєстрації.

Параметри критичного наближення до перешкод піддаються коригуванню за допомогою інтерактивного меню керування системою.

Інформація про поточний стан системи відображається на РК-дисплеї у вигляді відповідних індикаторів режиму:

- KN: Активація контролю небезпечної відстані через селектор SW2, де індекс N (від 1 до 6) вказує на вибране положення перемикача.
- MP (Memory Programming): Робота за попередньо встановленим параметром ALARM DISTANCE. Для переходу в цей режим у меню MODE KN/MP слід вибрати пункт MP та задати необхідний поріг спрацювання в розділі ALARM DISTANCE.
- XX: Режим моніторингу критичної дистанції деактивовано.

Візуалізація стану ультразвукового датчика HC-SR04 реалізована через вивід відповідних індикаторів на екран. Передбачено два основні алгоритми функціонування: Індикація “W”: Активна фаза, що супроводжується генерацією та прийомом ехо-сигналів від об’єктів. Індикація “S”: Стан енергозбереження або паузи, за якого випромінювання хвиль припиняється, а процес детектування деактивується.

Взаємодія з детектором та зміна параметрів його функціонування здійснюється за допомогою чотирьох багатоцільових кнопок: КН1 (“MENU”) – вхід до головного меню; КН2 (“+SELECT/START”) – вибір параметра або запуск процесу; КН3 (“-SELECT/STOP”) – зменшення значення або зупинка роботи; КН4 (“EXIT/INFO”) – вихід із налаштувань або перегляд довідкової інформації. Кожна клавіша має розширений функціонал залежно від поточного стану системи.

Клавіша КН1 (“MENU”) призначена для активації інтерфейсу конфігурування системи. Після її натискання користувач отримує доступ до ієрархічного меню, що включає такі параметри:

- DISTANCE FORMAT – параметр, що дозволяє обрати одиниці вимірювання дистанції до об’єкта. Користувач може налаштувати відображення даних на рідкокристалічному дисплеї (РКД) у метрах, сантиметрах або дюймах.

- **ALARM DISTANCE** – параметр, що визначає порогове значення дистанції до об'єкта. При досягненні або порушенні цієї межі система ініціює сигнал тривоги (акустичний або візуальний), сповіщаючи про небезпечне наближення.
- **MODE KN/MP** – параметр, що визначає алгоритм моніторингу граничної дистанції до об'єктів. Налаштування передбачає три стани: використання поточного положення селектора, зчитування заздалегідь збереженого в пам'яті значення або повну деактивацію функції контролю (OFF).

Навігація між розділами меню реалізована за допомогою кнопок КН2 та КН3, що відповідають за переміщення вгору та вниз відповідно. Для підтвердження вибору конкретного пункту використовується клавіша КН1. Безпосереднє коригування значень параметрів у межах обраного розділу здійснюється за допомогою повного набору функціональних клавіш (КН1–КН4).

Усі встановлені параметри записуються в енергонезалежну пам'ять мікроконтролера, що гарантує їх збереження навіть після вимкнення живлення. Кнопка КН2, позначена як “+SELECT/START”, виконує подвійну функцію. Вона використовується для активації режиму виявлення перешкод датчиком HC-SR04, а також слугує інструментом для вибору розділів у меню пристрою. Кнопка КН3 (“-SELECT/STOP”) виконує подвійну роль: вона зупиняє роботу датчика HC-SR04, переводячи його в пасивний режим без детектування об'єктів, а також слугує для навігації та вибору пунктів у системному меню.

Кнопка КН4 (“EXIT/INFO”) забезпечує доступ до поточних параметрів конфігурації детектора перешкод та одночасно слугує для повернення з підменю до попереднього рівня інтерфейсу.

Система забезпечує безперервне відображення даних на РК-моніторі: статус виявлення перешкод, дистанцію до об'єктів (з можливістю вибору одиниць вимірювання — см, м або дюйми), активний режим та рівень енергії

акумулятора. Паралельно ідентичний масив інформації транслюється через послідовний порт для подальшого програмного моніторингу.

Під час активного функціонування модуля детектування здійснюється циклічне сканування стану клавіш КН1–КН4 та зчитування даних із ультразвукового сенсора. Отримана інформація підлягає аналізу для виявлення об'єктів і розрахунку дистанції до них. З урахуванням заданих конфігурацій система формує та передає відповідні звіти на рідкокристалічний індикатор і послідовний інтерфейс.

У стані очікування функція сканування простору вимикається, проте пристрій продовжує трансляцію службових даних на РК-дисплей та послідовний інтерфейс. Для оптимізації енергоспоживання передбачено апаратне вимкнення індикатора за допомогою перемикача SW1, що жодним чином не перешкоджає стабільній роботі самого детектора.

Під час запуску детектора здійснюється первинна конфігурація та ініціалізація всіх периферійних модулів. Процес охоплює налаштування РК-дисплея, запуск послідовного інтерфейсу обміну даними, підготовку портів для опитування клавіш, а також активацію ШІМ-генератора для керування звуковим випромінювачем. Додатково виконується процедура самодіагностики світлодіодних індикаторів.

Програмне забезпечення зчитує з енергонезалежної пам'яті (EEPROM) мікроконтролера ключові конфігураційні параметри: поточний робочий режим, порогове значення критичної відстані та обрані одиниці вимірювання дистанції. Про успішний запуск системи свідчить виведення вітального повідомлення “BARRIERS DETECTOR” на рідкокристалічний дисплей, а також дублювання аналогічного інформаційного рядка у термінал послідовного порту.

На наступному етапі система переходить у стан активного функціонування, що передбачає безперервний цикл детектування об'єктів. Програмне забезпечення виконує розрахунок дистанції до перешкод та здійснює моніторинг стану клавіш керування КН1–КН4 для оперативного

реагування на дії користувача. За допомогою клавіші KN2 (“+SELECT/START”) користувач перемикає пристрій у стан функціонування, при якому запускається процес циклічного детектування перешкод. Візуалізація інформації на РКД розділена на дві зони. Верхня частина дисплея відображає наявність перешкод та критичні попередження. Нижня частина містить дані про відстань і поточний режим функціонування, де латинська літера “W” позначає активну детекцію, а “S” – стан спокою пристрою.

Додатково передбачено відображення статусу контролю дистанції: якщо ця функція активна, на екрані з’являється індикатор “KN” або “MP” залежно від обраних налаштувань, у протилежному випадку виводиться маркування “XX”, що свідчить про вимкнений режим моніторингу.

У разі порушення встановлених меж безпечної дистанції система генерує тривожне сповіщення “ALARM!!!”, яке транслюється через послідовний інтерфейс. Одночасно з цим активується акустична сигналізація та вмикається переривиста світлова індикація за допомогою світлодіода, що сигналізує про зменшення відстані до критичної позначки.

Система здійснює безперервний моніторинг стану джерела живлення. У випадку критичного зниження напруги акумулятора активується світлодіодна індикація червоного кольору (“Акумулятор розряджений”), а на графічному дисплеї відображається символ пустої батареї. Паралельно з цим через послідовний інтерфейс передається службове сповіщення “LOW BATTERY CHARGE”, що сигналізує про необхідність підзарядки.

Розробка програмного забезпечення здійснювалася в інтегрованому середовищі Arduino IDE. Для реалізації функціонала використано стандартні та спеціалізовані бібліотеки платформи. Перенесення відкомпільованого машинного коду в мікроконтролер плати Arduino Uno виконувалося за допомогою вбудованих засобів завантаження. Для попереднього тестування, аналізу роботи схеми та верифікації алгоритмів застосовано пакет моделювання Proteus ISIS.

РОЗДІЛ 4
ПРОГРАМНО-АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОГО
ПРИСТРОЮ ДЛЯ УНИКНЕННЯ ДРОНОМ ПЕРЕШКОД В
ЗАМКНУТИХ ТА ЗАДИМЛЕНИХ ПРИМІЩЕННЯХ

4.1. Алгоритм роботи цифрового пристрою для уникнення перешкод



Рис. 4.1. Схема функціонування детектора перешкод (початковий етап)

Рис. 4.2. Схема функціонування детектора перешок (продовження 1)

Рис.4.3. Схема функціонування детектора перешкод (продовження 2)

Рис. 4.4. Схема функціонування детектора перешкод (продовження 3)

Рис. 4.5. Схема функціонування детектора перешкод (продовження 4)

Рис. 4.6. Схема функціонування детектора перешок (продовження 5)

Рис. 4.7. Схема функціонування детектора перешкод (продовження б)

Рис .4.8. Схема функціонування детектора перешкод (продовження 7)

Рис. 4.9. Схема функціонування детектора перешикод (закінчення)

4.2. Створення програмного забезпечення для інтерфейсу з ультразвуковим далекоміром HC-SR04

Програмна взаємодія з цифровим модулем виявлення перешкод реалізована за допомогою спеціалізованого набору підпрограм. Зокрема, до складу розробленої бібліотеки входять інструменти для конфігурування сенсора HC-SR04:

```
// Налаштування пінів ультразвукового модуля
const int PIN_TRIGGER = 13;
const int PIN_ECHO = 2;

void setup_sensor() {
    // Конфігурація режимів роботи портів
    pinMode(PIN_TRIGGER, OUTPUT);
    pinMode(PIN_ECHO, INPUT);

    // Гарантуємо низький рівень на тригері при старті
    digitalWrite(PIN_TRIGGER, LOW);
}

unsigned int get_distance_cm() {
    long duration_us;
    unsigned int result_cm;

    // Формування акустичного запиту
    digitalWrite(PIN_TRIGGER, LOW);
    delayMicroseconds(2);
    digitalWrite(PIN_TRIGGER, HIGH);
    delayMicroseconds(10); // Стандартна тривалість стробуючого імпульсу
    digitalWrite(PIN_TRIGGER, LOW);

    // Фіксація часу відлуння (в мікросекундах)
    duration_us = pulseIn(PIN_ECHO, HIGH);

    // Обчислення дистанції за фізичною константою (швидкість звуку)
    // Формула: (час * 0.034) / 2 або просто розподіл на 58.2
    result_cm = (unsigned int)(duration_us / 58);

    return result_cm;
}
```

4.3. Створення програмних компонентів для керування режимами роботи пристрою для уникнення перешкод

```
// Визначення бітових масок для індикації стану живлення
// Кожен байт відповідає одному рядку пікселів (5x8 точок)
const uint8_t batteryFullIcon[8] = {
    0b01110, // ### (ковпачок)
```

```

0b01110, // ###
0b11111, // #####
0b11111, // #####
0b11111, // #####
0b11111, // #####
0b11111, // #####
0b11111 // #####
};

const uint8_t batteryEmptyIcon[8] = {
    0b01110, // ###
    0b01110, // ###
    0b11011, // ##  ##
    0b10001, // #   #
    0b10001, // #   #
    0b10001, // #   #
    0b11111, // #####
    0b11111 // #####
};

// Реєстрація гліфів у пам'яті контролера дисплея (CGRAM)
void initCustomChars() {
    display.createChar(0, batteryFullIcon); // Присвоєння індексу 0 повному заряду
    display.createChar(1, batteryEmptyIcon); // Присвоєння індексу 1 низькому заряду
}

// Приклад використання в коді:
display.write(byte(0)); // Візуалізація зарядженої батареї
display.write(byte(1)); // Візуалізація розрядженої батареї

```

Функція моніторингу заряду акумуляторної батареї:

```

/* Функція виконує замір напруги на АКБ та виводить стан на екран */
void monitorPowerSupply() {
    // Зчитування сигналу з дільника напруги
    int rawAdcValue = analogRead(BAT_VOLT_PIN);
    // Обчислення напруги (масштабоване значення для порівняння)
    // Використовуємо long для запобігання переповненню при множенні
    long scaledVoltage = (rawAdcValue * 100L) / 1024;
    // Позиція для відображення іконки (крайній правий кут нижнього рядка)
    display.setCursor(15, 1);
    if (scaledVoltage >= 55) { // Напруга в нормі (понад 5.5В)
        display.write(byte(0)); // Відобразити іконку повного заряду
    } else { // Критично низький рівень енергії
        display.write(byte(1)); // Відобразити іконку розрядженої батареї
        // Повідомлення в діагностичний порт
        Serial.println(F("SYSTEM ALERT: Critical voltage drop!"));
        // Світлова індикація тривоги
        triggerLowBatWarning(); } }
// Допоміжна функція для світлового сигналу
void triggerLowBatWarning() {
    digitalWrite(LED_LOW_BAT, HIGH); delay(1000); // Активна фаза попередження
    digitalWrite(LED_LOW_BAT, LOW);
}

```

Функція запису long int-даних в пам'ять EEPROM:

```
/* Збереження багатобайтового числа в енергонезалежну пам'ять
 * @param addr Початкова адреса в EEPROM
 * @param data Значення типу long для запису */
void commitValueToMemory(int addr, long data) {
    // Метод put автоматично визначає розмір типу long (4 байти)
    // та записує його за вказаною адресою
    EEPROM.put(addr, data);
}
```

Функція зчитування long int-даних з пам'яті EEPROM:

```
/* Зчитування 32-бітного числа з енергонезалежної пам'яті
 * @param addr Початкова адреса комірки
 * @return Відновлене значення типу long */
long fetchStoredValue(int addr) {
    long retrievedData = 0;
    // Автоматично зчитує 4 послідовні байти та реконструює число
    EEPROM.get(addr, retrievedData);
    return retrievedData;
}
```

Функція подачі звукового сигналу оповіщення. Тривалість сигналу: 2 секунди:

```
void triggerAudioAlert() {
    // Встановлюємо пін як вихід перед початком генерації
    pinMode(BEEP_PIN, OUTPUT);
    // Коефіцієнт заповнення ШІМ (50%) для отримання чистого звуку
    const byte dutyCycle = 127;
    // Активація генерації сигналу
    analogWrite(BEEP_PIN, dutyCycle);
    // Витримка часу (використовуємо один виклик для кращої читаності)
    delay(2000);
    // Зупинка генерації ШІМ-сигналу
    analogWrite(BEEP_PIN, 0);
    // Переведення піна у стан високого опору для енергозбереження
    pinMode(BEEP_PIN, INPUT);
}
```

Функція визначення активного положення багатопозиційного перемикача. Обробляє аналоговий сигнал з дільника напруги селектора SW2:

```
void updateSelectorState() {
    // Зчитуємо сире значення з піна селектора
    int rawSelectorValue = analogRead(SEL_INPUT_PIN);
    // Масштабування для зручності порівняння (аналог вашої логіки)
    // Використано цілочисельну арифметику для швидкості
    byte normalizedLevel = (byte)((rawSelectorValue * 50L) / 1024);
    // Визначення індексу режиму (tv) на основі порогів напруги
    if (normalizedLevel >= 49) { targetMode = 6; }
    else if (normalizedLevel >= 41) { targetMode = 5; }
    else if (normalizedLevel >= 33) { targetMode = 4; }
    else if (normalizedLevel >= 25) { targetMode = 3; }
    else if (normalizedLevel >= 16) { targetMode = 2; }
```

```

else if (normalizedLevel >= 8) { targetMode = 1; }
else { targetMode = 0; // Стан, коли нічого не вибрано або мінімальне значення
}
// Відображення індексу на LCD (якщо потрібно)
display.setCursor(12, 1); // display.print(targetMode);
}

```

Функція світлової індикації тривоги (червоний світлодіод). Використовується при критичних помилках або низькому заряді акумулятора.

```

void triggerRedAlert() {
    const int pulseDuration = 1000; // Тривалість спалаху в мс
    digitalWrite(PIN_LED_RED, HIGH); delay(pulseDuration);
    digitalWrite(PIN_LED_RED, LOW); }

```

Функція світлової індикації статусу (синій світлодіод). Повідомляє про штатну роботу або успішне зчитування даних:

```

void notifyBlueStatus() {
    // Використовуємо прямий запис для візуальної відмінності від попередньої функції
    digitalWrite(PIN_LED_BLUE, HIGH); delay(1000); digitalWrite(PIN_LED_BLUE, LOW); }

```

Інтерфейс головного меню конфігурації. Керування за допомогою кнопок навігації та підтвердження:

```

void displayConfigurationMenu() {
    // Список доступних опцій налаштування
    const char *configOptions[] = {
        "UNIT FORMAT  ", // Формат одиниць виміру
        "ALARM RANGE  ", // Поріг спрацювання тривоги
        "SYSTEM MODE  " // Вибір режиму роботи (KN/MP)
    };
    byte currentOption = 0; const byte totalOptions = 3;
    // Очищення та підготовка заголовка
    lcd.clear(); lcd.setCursor(0, 0); lcd.print(F("--- SETTINGS ---"));
    while (true) {
        // Відображення активного пункту меню
        lcd.setCursor(0, 1); lcd.print(configOptions[currentOption]);
        // Логіка перемикання "ВПЕРЕД" (KH2)
        bool btnNext = digitalRead(KH2);
        if (btnNext && !old_val2) {
            currentOption = (currentOption + 1) % totalOptions; delay(50); }
        old_val2 = btnNext;
        // Логіка перемикання "НАЗАД" (KH3)
        bool btnPrev = digitalRead(KH3);
        if (btnPrev && !old_val3) {
            currentOption = (currentOption == 0) ? (totalOptions - 1) : (currentOption - 1);
            delay(50); } old_val3 = btnPrev;
        // Кнопка "ВИХІД" (KH4)
        bool btnExit = digitalRead(KH4);
        if (btnExit && !old_val4) { lcd.clear(); old_val4 = true; return; }
    }
}

```

```

old_val4 = btnExit;
// Кнопка "ПІДТВЕРДЖЕННЯ/ОК" (KH1)
bool btnEnter = digitalRead(KH1);
if (btnEnter && !old_val1) {
    switch (currentOption) {
        case 0: configureDistanceFormat(); break;
        case 1: setAlarmThreshold(); break;
        case 2: toggleOperationMode(); break; }
    // Після повернення з підменю відновлюємо заголовок
    lcd.setCursor(0, 0); lcd.print(F("--- SETTINGS ---")); }
old_val1 = btnEnter; delay(20); // Стабілізація циклу
} }

```

Функція налаштування відображення дистанції. Для вибору способу представлення даних про відстань передбачено підменю “DISTANCE FORMAT”. Керування цим параметром здійснюється за допомогою функції: `void lenformat (void);`

Налаштування порогу спрацьовування сигналізації. У підменю “ALARM DISTANCE” користувач має можливість визначити гранично допустиму дистанцію (мінімальну критичну відстань). Програмна реалізація здійснюється функцією: `void alarmdistance (void).`

Налаштування алгоритму моніторингу дистанції. У розділі меню “MODE KN/MP” користувач може обрати конкретний режим відстеження критичної відстані. За логіку перемикання та функціонування цього режиму відповідає функція: `void controldc (void).`

4.4. Розроблення основних програмних функцій цифрового пристрою для уникнення перешкод

Програмне забезпечення цифрового пристрою для виявлення перешкод базується на платформі Arduino та сенсори HC-SR04. Послідовність виконання операцій відображена у графічних матеріалах (Рис. 4.1–4.8), а програмна реалізація представлена у додатку до роботи.

4.5. Програмна симуляція та оцінка результатів функціонування цифрового пристрою для уникнення перешкод в середовищі Proteus ISIS

Результатом компіляції скетчу в середовищі Arduino IDE для контролера Arduino Uno (на базі мікросхеми ATmega328P) є файл формату `.hex`, що містить об'єктний машинний код. Дана прошивка завантажується в пам'ять мікроконтролера для подальшої верифікації функціонування пристрою детекції перешкод.

Для верифікації апаратної конфігурації детектора та аналізу алгоритмів керуючого ПЗ доцільно використовувати середовище моделювання Proteus ISIS. На наведених нижче ілюстраціях продемонстровано результати симуляції цифрового пристрою для виявлення перешкод, побудованої на базі контролера Arduino та ультразвукового датчика HC-SR04.

Процес функціонування детектора перешкод розпочинається з етапу системної ініціалізації після подачі напруги живлення. На старті мікроконтролер AVR активує виконання вбудованого програмного забезпечення, що послідовно здійснює конфігурацію внутрішніх змінних, а згодом – налаштування підключених сенсорів та периферійних модулів.

Програмне забезпечення виконує конфігурацію АЦП, виводу ШІМ для керування звуковим випромінювачем (піщалкою) LS, а також налаштовує роботу рідкокристалічного дисплея за 4-бітною шиною (лінії D0–D3). Одночасно мікроконтролер переходить до моніторингу стану клавіші меню КН1. Для ініціалізації символьного РКД застосовується функція `lcd.begin(16, 2)`. По завершенню підготовчих процедур на екран виводиться привітальне повідомлення "BARRIERS DETECTOR", як показано на Рис. 4.10.

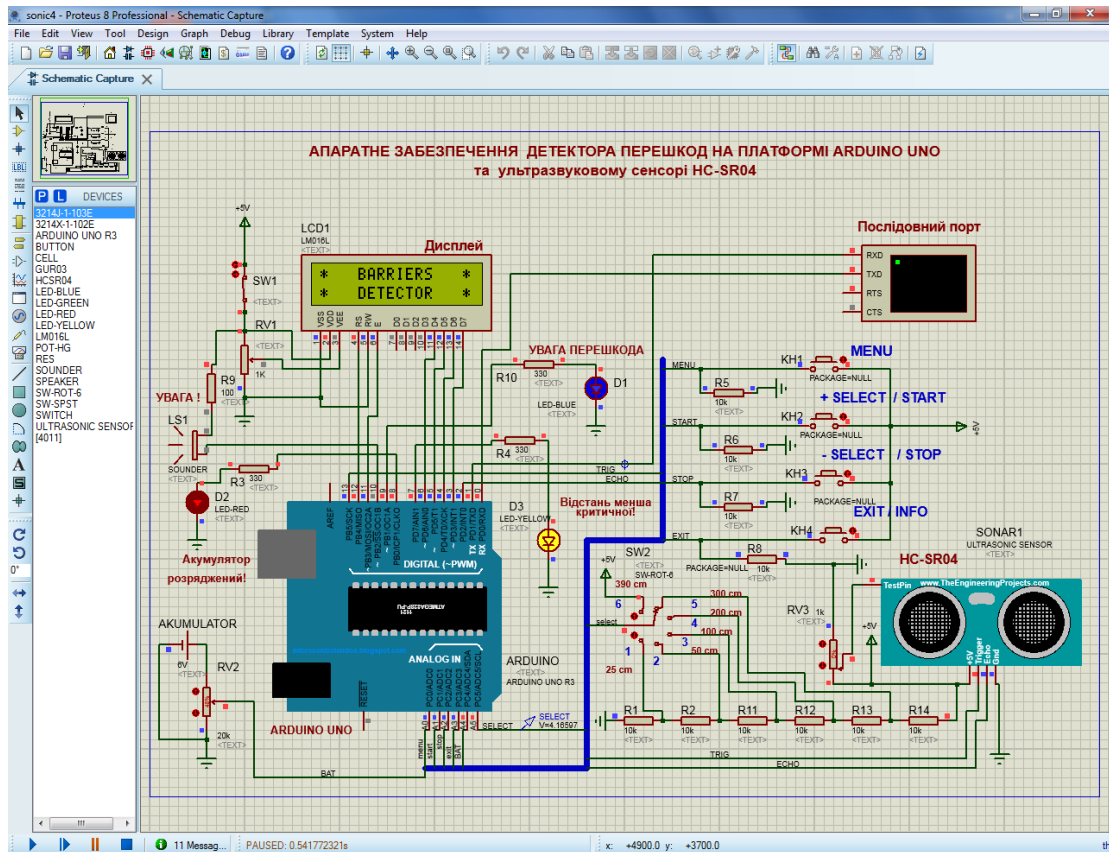


Рис. 4.10. Візуалізація етапу запуску детектора перешкод у середовищі *Proteus ISIS*

Пристрій розпочинає роботу з процедури самодіагностики: активуються світлодіодні індикатори D1–D3 та спрацьовує звуковий сигнал LS1. Після цього мікроконтролер вилучає з енергонезалежної пам'яті (EEPROM) ключові параметри конфігурації, зокрема вибраний режим, порогове значення дистанції та спосіб відображення даних про перешкоди.

Верхній рядок РК-дисплея відведений для відображення даних про виявлені об'єкти. У випадку, коли дистанція до найближчої перешкоди перевищує межу у 4 метри, на екрані з'являється символ “?” (що сигналізує про відсутність перешкод у зоні видимості), як показано на Рис. 4.11.

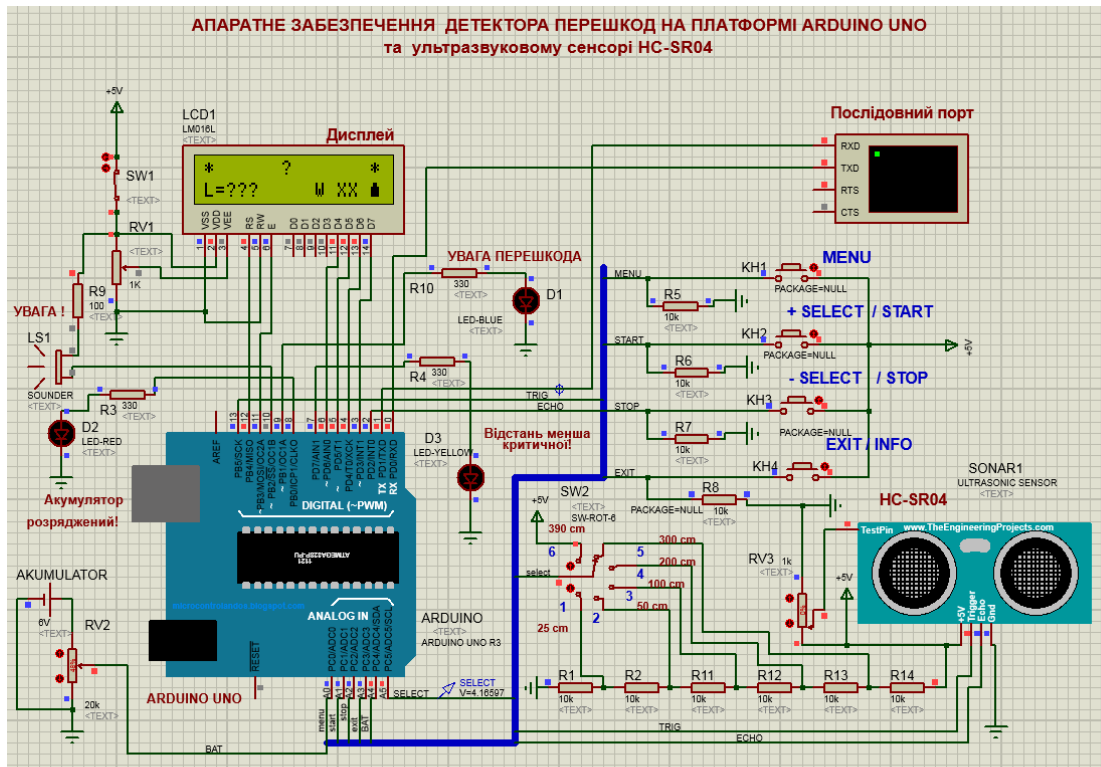


Рис. 4.11. Результати симуляції детектора перешкод у Proteus ISIS

Другий рядок LCD-дисплея призначений для індикації поточної дистанції до перешкоди. Якщо дані про відстань відсутні або знаходяться поза межами вимірювання, з'являється напис "L=???". Поруч відображається статус активності пристрою: "W": активний моніторинг із використанням ультразвукового датчика; "XX": режим, у якому аналіз безпечної відстані призупинено. Справа розміщено графічний символ акумулятора, заповнений колір якого свідчить про належний рівень напруги живлення.

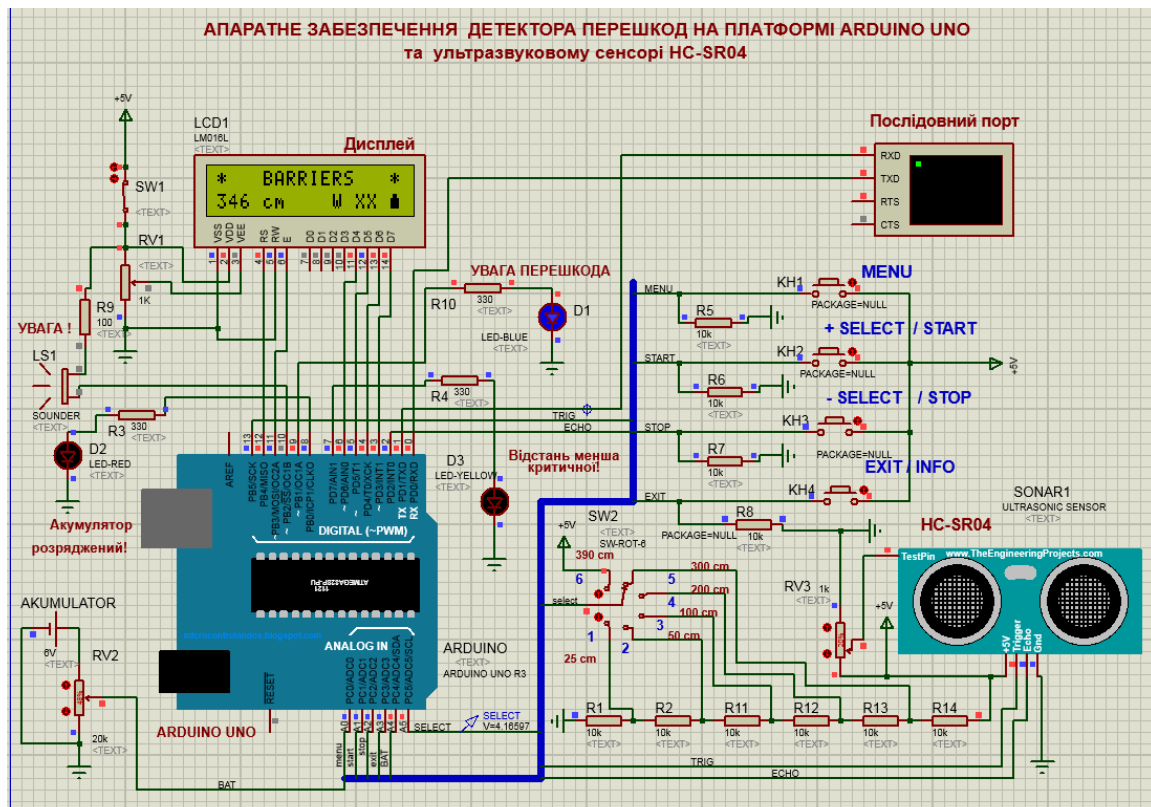


Рис. 4.12. Візуалізація функціонування системи детекції об'єктів у середовищі Proteus ISIS ("W" - активний моніторинг)

У разі виявлення об'єкта ультразвуковим датчиком HC-SR04, на верхній рядок РК-дисплея виводиться попередження "BARRIER". Одночасно з цим у нижньому рядку відображається дистанція до знайденої перешкоди (одиниця виміру – сантиметри). Візуальне сповіщення доповнюється миготінням синього світлодіода D1, що відповідає за сигнал "УВАГА ПЕРЕШКОДА" (Рис. 4.12, 4.13).

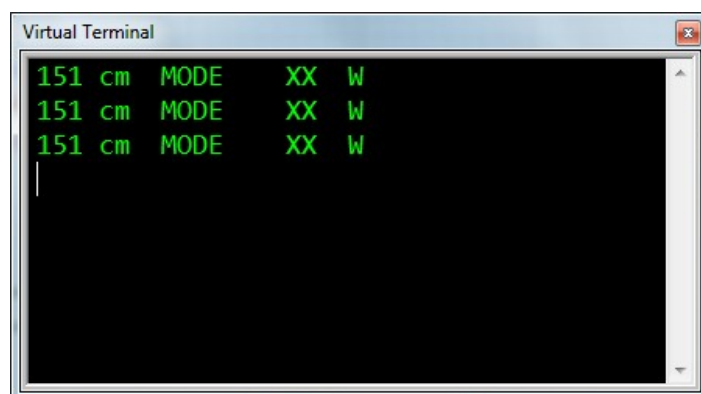


Рис. 4.13. Дублювання даних у послідовний інтерфейс (Serial Port)

Ключовою функцією детектора є можливість генерації попереджувальних сигналів у разі наближення до об'єкта на відстань, що

менша за встановлений поріг. Параметр критичної дистанції визначається за допомогою перемикача SW2 (для швидкого вибору) або через системне меню для більш прецизійного регулювання. У налаштуваннях також обирається метод контролю: спрощений (за положенням селектора) або високоточний (згідно із записаними в меню значеннями). Навігація та керування параметрами інтерфейсу здійснюються за допомогою функціональних клавіш КН1–КН4. Для активації функції моніторингу критичної дистанції необхідно перейти до меню конфігурації. Це здійснюється натисканням функціональної клавіші КН1 (“MENU”), як продемонстровано на Рис. 4.14 та 4.15.

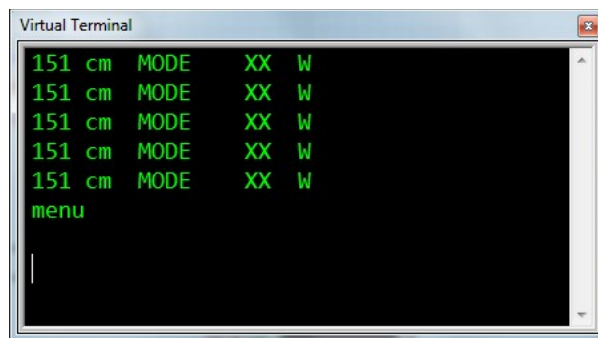


Рис. 4.14. Підтвердження входу в режим налаштувань через термінал послідовного порту

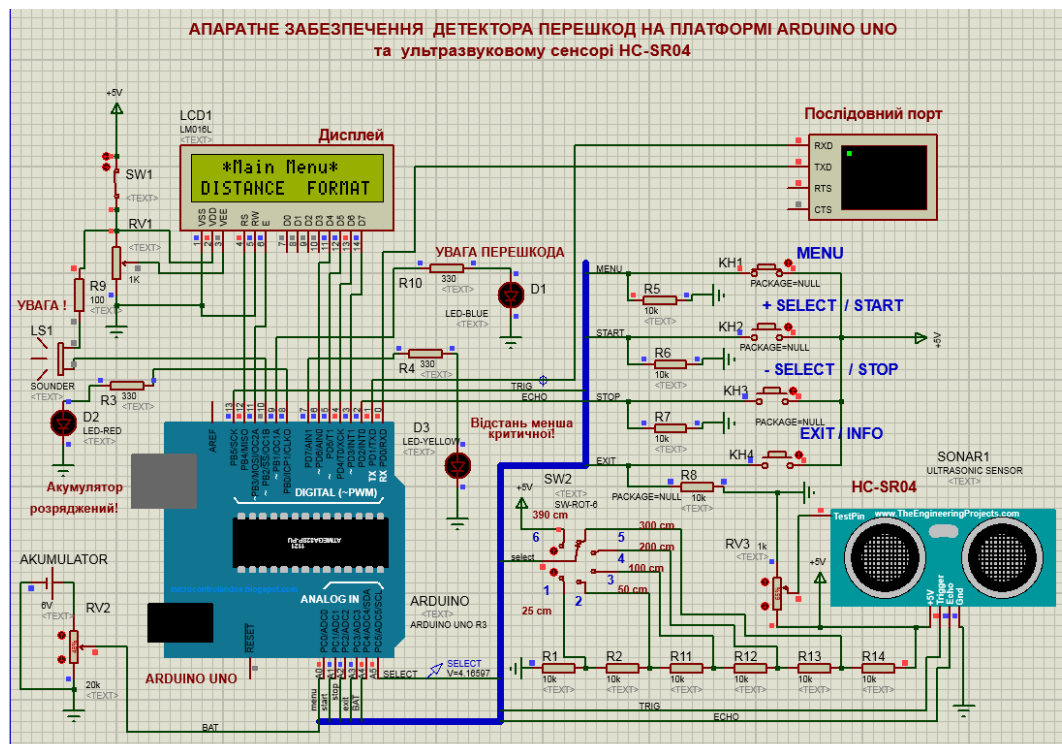
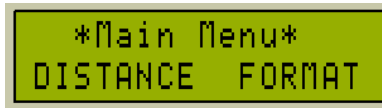


Рис. 4.15. Візуалізація інтерфейсу налаштувань детектора в середовищі Proteus ISIS

Структура інтерфейсу керування складається з таких основних пунктів (підменю):



DISTANCE FORMAT – підменю, що дозволяє визначити зручний формат представлення результатів вимірювання. Система підтримує три варіанти виводу відстані на LCD: см, м або дюйми.



ALARM DISTANCE — підменю регулювання небезпечної дистанції. Тут задається конкретна відстань до перешкоди, яка вважається критичною для безпечного руху або роботи пристрою.



MODE KN/MP – підменю налаштування способу контролю граничної відстані. Дозволяє перемикатися між режимом зчитування з селектора та режимом роботи за значенням, що зафіксоване в пам'яті (ALARM DISTANCE), як показано на Рис. 4.16.

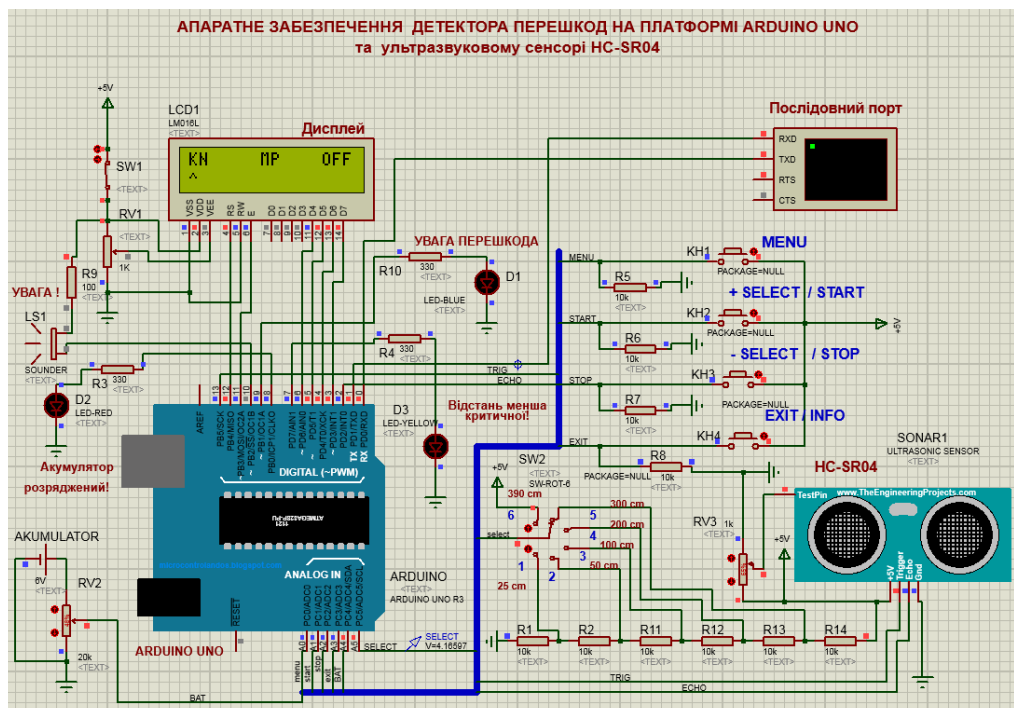


Рис. 4.16. Симуляція інтерфейсу вибору режиму KN/MP в Proteus ISIS

За допомогою функціональних кнопок “+SELECT/START” та “-SELECT/STOP” здійснюється вибір алгоритму моніторингу граничної дистанції. Передбачено три варіанти функціонування:

- KN: контроль базується на положенні апаратного перемикача SW2.
- MP: аналіз виконується згідно зі значенням, зафіксованим у розділі ALARM DISTANCE.
- OFF: режим стеження за критичним наближенням деактивовано.

Для збереження встановлених параметрів у пам’яті EEPROM пристрою необхідно натиснути клавішу “MENU”. Після завершення налаштувань вихід із поточного розділу забезпечується натисканням кнопки “EXIT/INFO”.

Інформація про активний метод контролю відображається безпосередньо на екрані пристрою за допомогою спеціальних індикаторів. Приклади відображення різних режимів наведено на Рис. 4.22, 4.23, 4.24 та 4.25. Як приклад розглянемо активацію режиму KN, при якому поріг спрацьовування сигналізації визначається положенням апаратного селектора SW2 (Рис. 4.17).

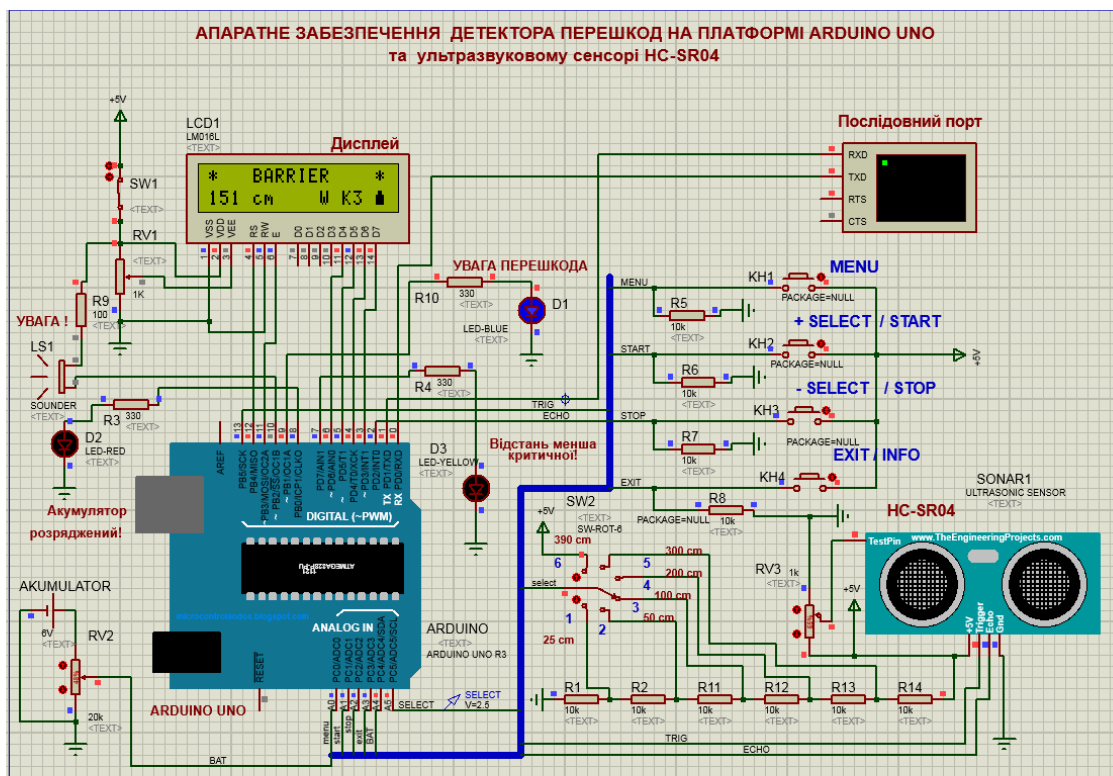


Рис. 4.17. Симуляція роботи детектора в середовищі Proteus ISIS при активації режиму KN

Функціонування в режимі КЗ: Як показано на Рис. 4.18, при скороченні дистанції до перешкоди (об'єкта) нижче встановленої межі (менше 100 см) активується протокол попередження. Система реагує вмиканням жовтого індикатора D3 та генерацією звукового сигналу через випромінювач LS1. Одночасно з цим на верхньому рядку РК-дисплея з'являється попереджувальне повідомлення "ALARM!!!".

Аналогічні дані передаються через послідовний інтерфейс і відображаються у вікні термінала (Рис. 4.19) у форматі: "ALARM !!! LMAX=100 L=99 см". Це дозволяє здійснювати віддалений моніторинг параметрів системи в режимі реального часу.

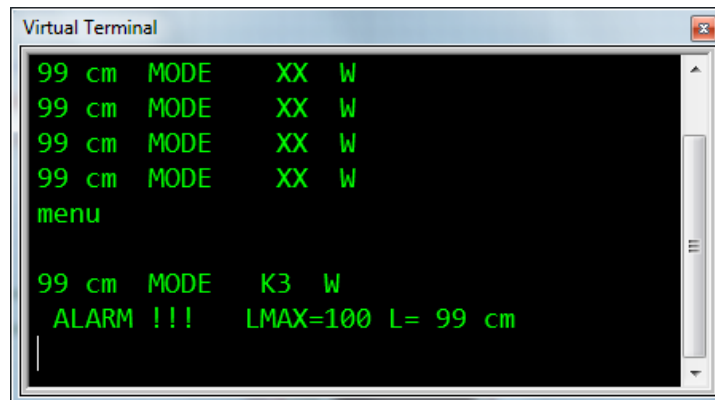


Рис. 4.19. Відображення діагностичних та попереджувальних повідомлень у вікні віртуального термінала

Коли дистанція до перешкоди (об'єкта) збільшується і виходить за межі критичного діапазону, але перешкода все ще залишається в полі зору сенсора, система переходить у режим очікування. При цьому активується синій світлодіод D1 з індикацією "УВАГА ПЕРЕШКОДА". Пристрій продовжує безперервний моніторинг відстані, оновлюючи відповідні показники на РК-дисплеї.

Аналіз функціонування детектора в режимі програмування пам'яті (MP) здійснюється на основі параметра ALARM DISTANCE. Для активації цього алгоритму необхідно в головному меню MODE KN/MP обрати опцію MP, після чого в спеціалізованому підменю ALARM DISTANCE задати необхідний поріг критичної відстані (Рис. 4.20, 4.21).

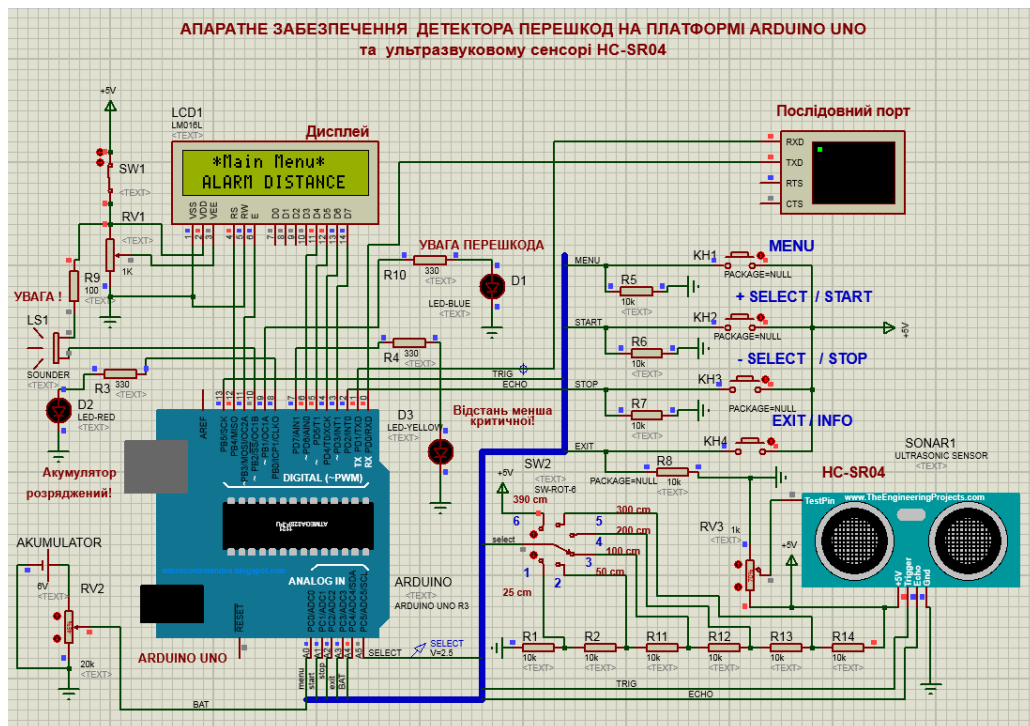


Рис. 4.20. Візуалізація процесу конфігурування порогу спрацювання у підменю “ALARM DISTANCE”

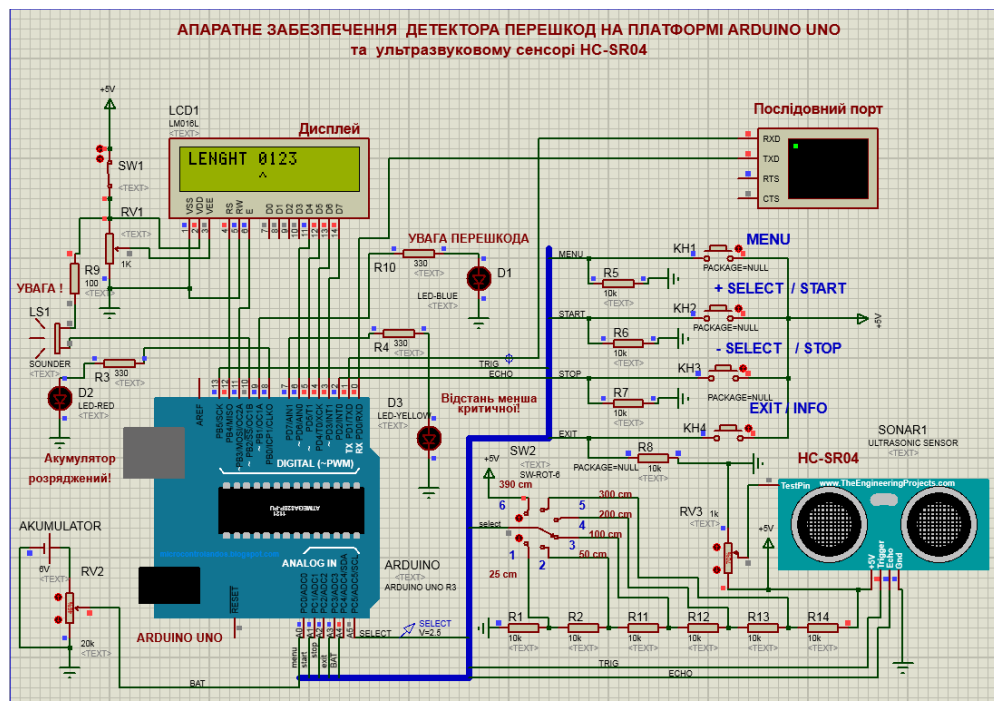


Рис. 4.21. Результати імітаційного моделювання пристрою виявлення перешкод у Proteus ISIS

Підменю “ALARM DISTANCE” призначене для задання порогу критичної дистанції. Редагування окремих розрядів параметра LENGTH (відстань) виконується кнопками “+SELECT/START” та “-SELECT/STOP”.

Для навігації між розрядами (переміщення курсора) використовується клавіша “MENU”, а завершення роботи з налаштуваннями здійснюється натисканням “EXIT/INFO”. Усі встановлені дані автоматично фіксуються в енергонезалежній пам’яті (EEPROM) мікроконтролера. Підсумком конфігурування є такі параметри:

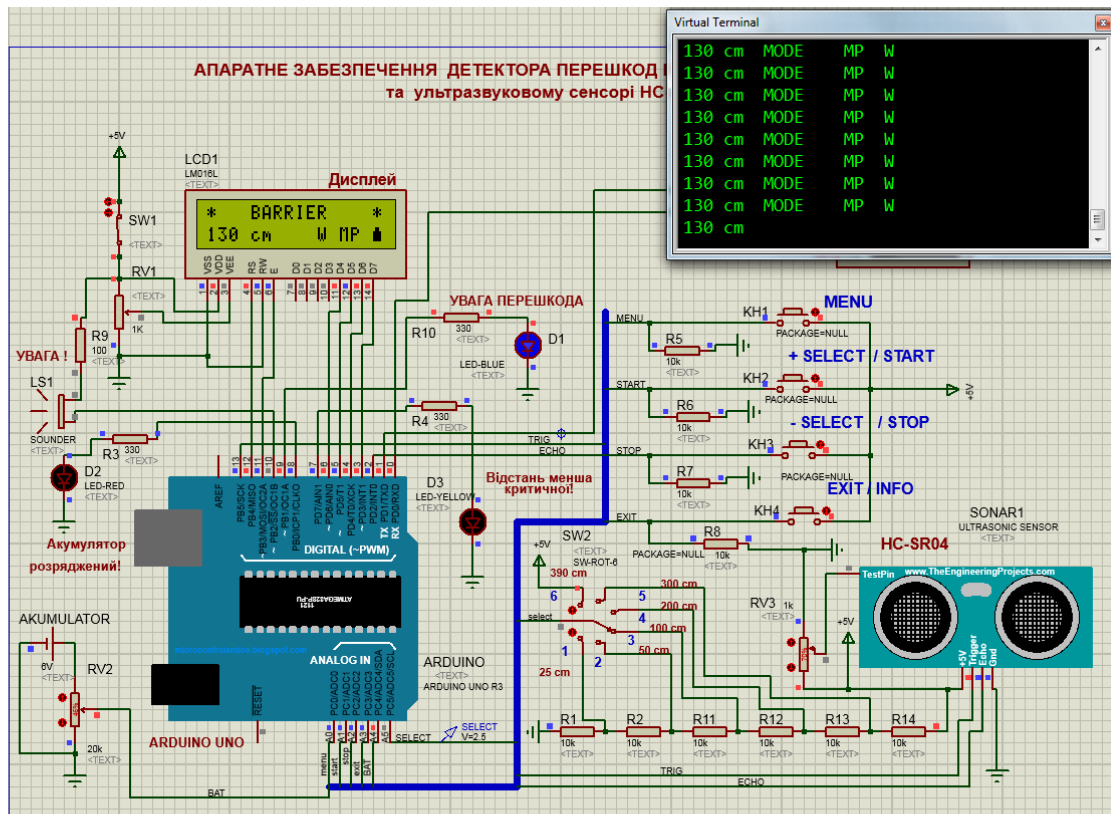


Рис. 4.22. Результати симуляції детектора перешкод у середовищі Proteus

ISIS. Функціонування в режимі MP: здійснення моніторингу граничної відстані до об’єкта відповідно до параметрів, внесених у пам’ять пристрою

Пристрій передбачає два ключові стани функціонування: активний моніторинг (режим “W”), під час якого ультразвуковий датчик здійснює пошук перешкод, та енергоефективний стан (режим “S”), що вимикає сенсор для зменшення споживання живлення.

Перехід пристрою в режим очікування здійснюється за допомогою кнопки КН3 “+SELECT/STOP”. Після її натискання активний моніторинг припиняється, а на рідкокристалічному дисплеї (РКД) з’являється відповідний індикатор “S”. У цьому стані функція вимірювання відстані вимикається (Рис. 4.23 та 4.24).

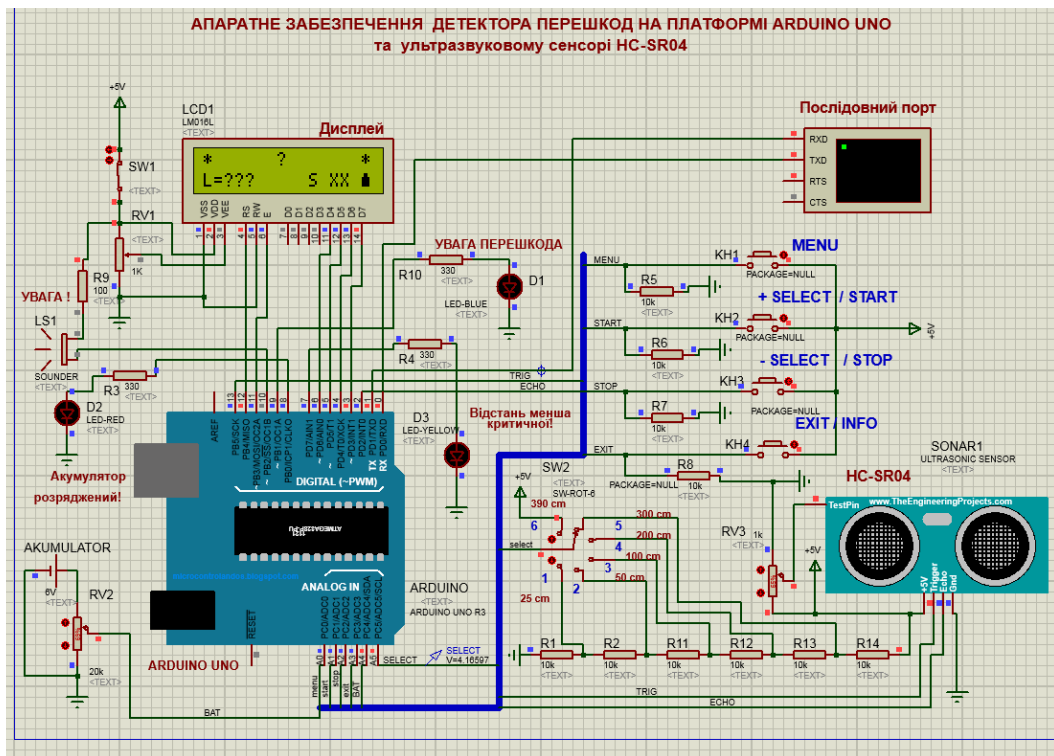


Рис. 4.23. Результати симуляції функціонування детектора в середовищі Proteus ISIS. Відображено перехід системи у стан “S” – енергоефективний режим очікування

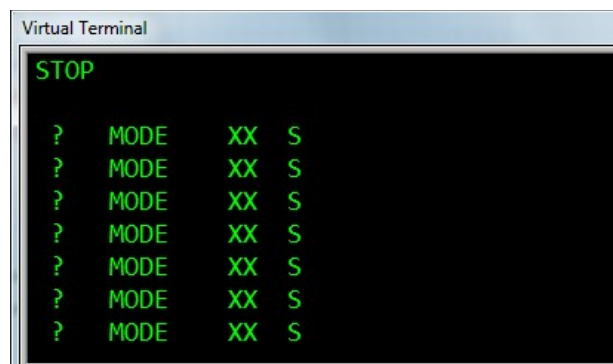


Рис. 4.24. Відображення службових даних на віртуальному терміналі.

Система перебуває у стані “S” (режим очікування)

Активація робочого стану здійснюється натисканням кнопки КН2 “+SELECT/START”. Після переходу прилад відображає на рідкокристалічному дисплеї символ “W”, що свідчить про початок процесу сканування простору. Система починає вимірювати дистанцію до перешкоди.

Доступ до параметрів детектора відкритий у обох станах системи (як у “W”, так і в “S”). Для виклику цієї інформації використовується кнопка КН4 “EXIT/INFO” (Рис. 4.25, 4.26). При цьому через послідовний інтерфейс

передаються дані про встановлені пороги критичних відстаней для режимів MP та KN, а також відображається поточний статус пристрою ("CURRENT

Аналогічні дані на короткий час дублюються на РКД, зокрема:

- MP DISTANCE — значення контрольної відстані, збережене в пам'яті пристрою;
- KN DISTANCE — ліміт відстані, заданий за допомогою перемикача SW2;
- CURRENT MODE — поточний активований режим моніторингу

```

130 cm  MODE   K3  W
130 cm  MODE   K3  W
130 cm  MODE   K3  W
*****  SETUP INFORMATION  *****
MP DISTANCE= 51 cm
KN DISTANCE= 100 cm
CURRENT MODE KN
*****
129 cm
  
```

Рис. 4.25. Вивід службової інформації про поточні налаштування детектора

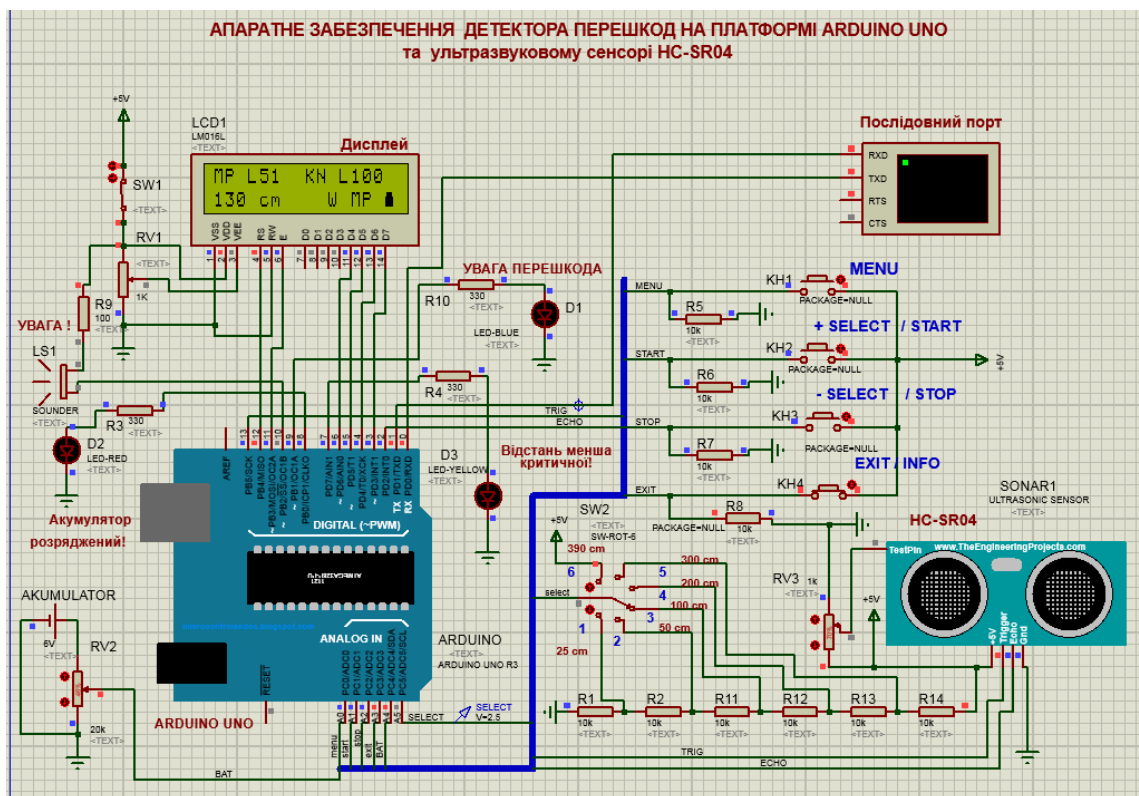


Рис. 4.26. Процес симуляції детектора в Proteus ISIS. На дисплей виведено інформаційне вікно з поточними налаштуваннями пристрою

У системі встановлено такі граничні параметри: для режиму MP критична дистанція (MP L51) зафіксована на позначці 51 см, тоді як для

режиму KN (KN L100) цей показник становить 100 см. Особливістю алгоритму МР є можливість прецизійного налаштування моніторингу об'єктів із кроком в один сантиметр. Допоміжна інформація слугує для верифікації коректності поточної конфігурації приладу. У разі необхідності коригування експлуатаційних параметрів детектора користувач може скористатися відповідним інтерфейсом. Перехід до меню налаштувань виконується шляхом натискання функціональної клавіші КН1 “MENU”. Система підтримує відображення дистанції до перешкоди (об'єкта) у різних одиницях вимірювання: крім сантиметрів, доступні метри та дюйми. Вибір бажаного формату відображення виконується у відповідному розділі меню:

DISTANCE FORMAT – спеціалізоване підменю, призначене для вибору одиниць вимірювання, у яких інформація про відстань відображатиметься на екрані РКД (доступні варіанти: сантиметри, метри або дюйми), Рис. 4.27.

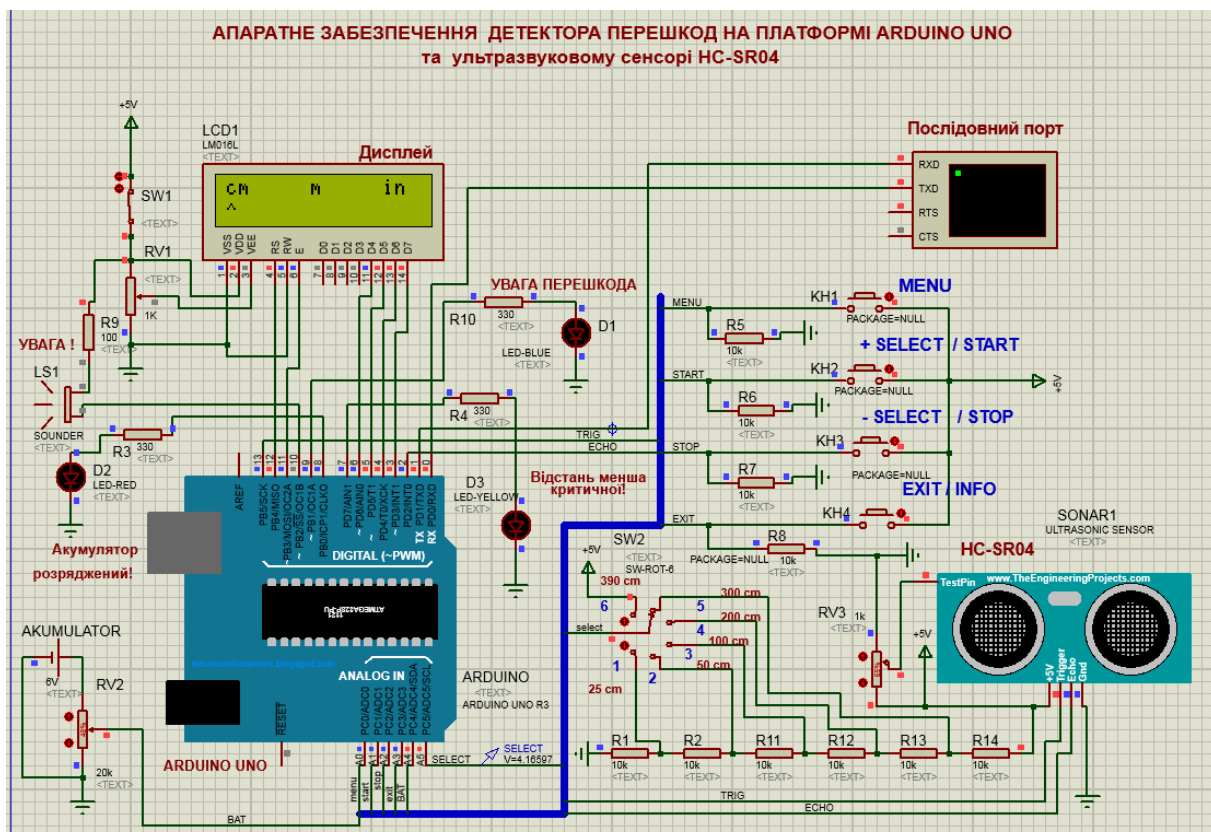
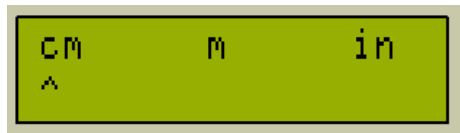
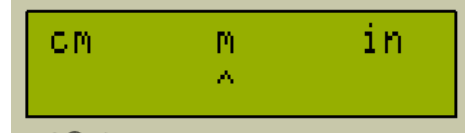


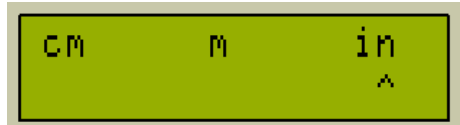
Рис. 4.27. Демонстрація інтерфейсу для вибору одиниць виводу дистанції до перешкоди



Задаємо відстань в сантиметрах



Задаємо відстань в метрах



Задаємо відстань в дюймах

Рис. 4.28. Визначення одиниць виводу дистанції до перешкоди

Для навігації по пунктах меню використовуються клавіші “+SELECT/START” та “-SELECT/STOP”, які забезпечують переміщення курсора. Обраний тип одиниць вимірювання фіксується натисканням кнопки “MENU”, а повернення до попереднього рівня інтерфейсу здійснюється за допомогою клавіші “EXIT/INFO”. Приклади відображення дистанції в різних форматах наведено на наступних ілюстраціях:



відстань в дюймах



ця ж відстань в сантиметрах



відстань в метрах

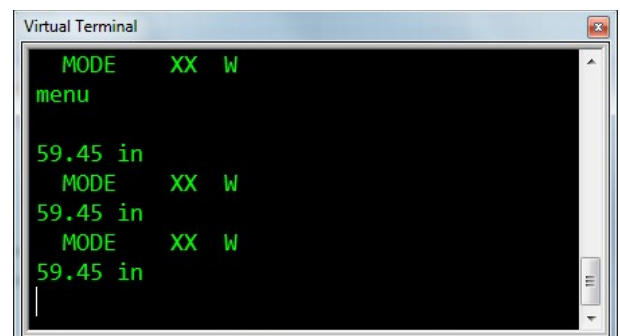
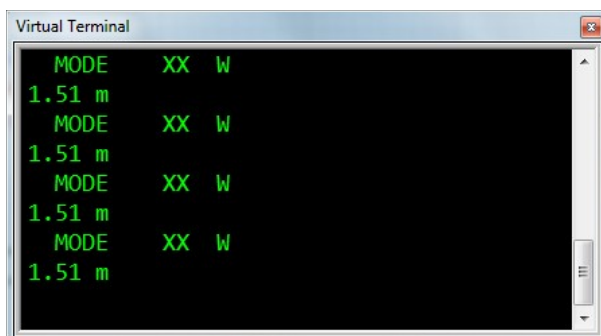


Рис. 4.29. Приклади візуалізації виміряної дистанції в обраних системах вимірювання

З метою оптимізації енергоспоживання в активному режимі детектування передбачено вимикач SW1, який дозволяє деактивувати живлення рідкокристалічного дисплея. Крім того, архітектура системи включає функціонал моніторингу рівня заряду акумуляторної батареї.

У разі падіння напруги на акумуляторі нижче рівня 5,5 В, система активує режим оповіщення про критичний розряд. На рідкокристалічному дисплеї (координати 1,15) з'являється піктограма порожнього гальванічного елемента, що супроводжується миготливою індикацією червоного світлодіода “Акумулятор розряджений” (Рис. 4.30).

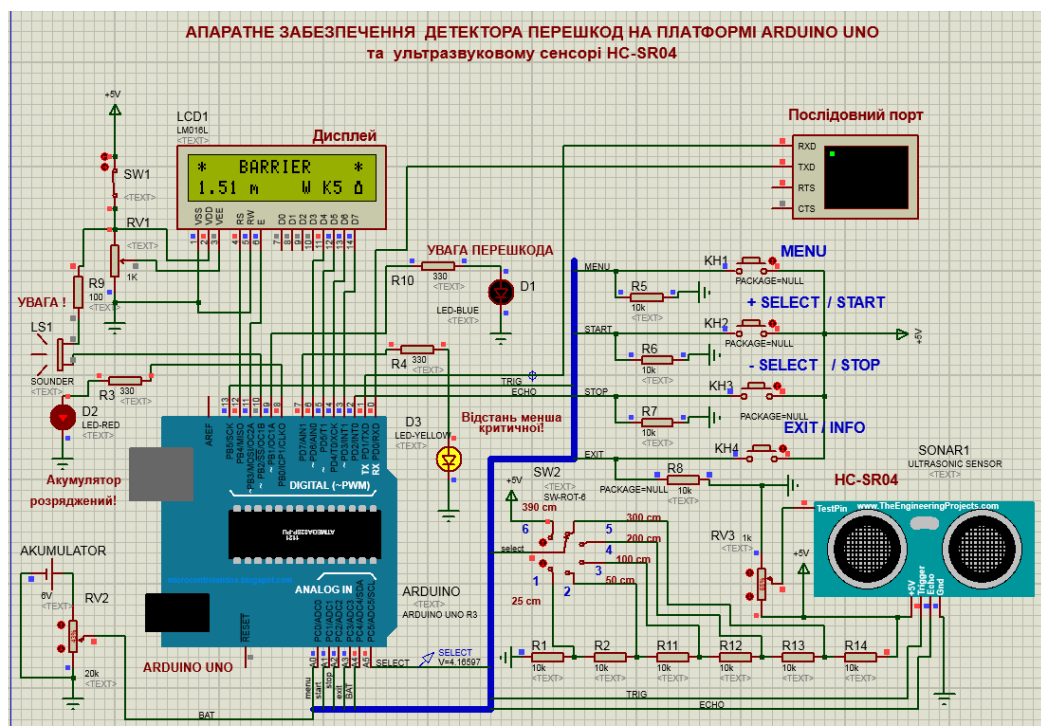


Рис. 4.30. Моделювання стану критичного розряду джерела живлення в Proteus ISIS.

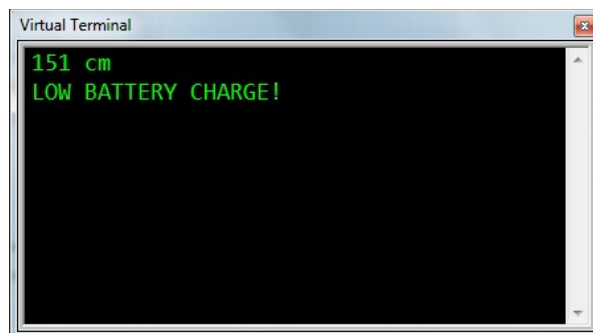


Рис. 4.31. Реєстрація події критичного розряду батареї у віртуальному терміналі

Паралельно з візуальною індикацією, система передає діагностичне повідомлення “LOW BATTERY CHARGE!” у віртуальний термінал, що дозволяє дистанційно моніторити стан живлення пристрою (Рис. 4.31).

З метою оптимізації енергоспоживання системи під час активного моніторингу перешкод передбачено вимикач SW1, який дозволяє деактивувати живлення рідкокристалічного дисплея.

ВИСНОВКИ

Результатом кваліфікаційної роботи є створення апаратного та програмного забезпечення цифрового пристрою, що забезпечує автономне уникнення перешкод дроном. Система розроблена на базі мікроконтролера Arduino та ультразвукового дальноміра HC-SR04 для ефективного маневрування дрона в обмеженому просторі та умовах низької видимості (задимленості).

Функціонал розробленого детектора охоплює ідентифікацію перешкод та обчислення дистанції до них з подальшою трансляцією результатів через послідовний інтерфейс. У разі порушення встановлених лімітів безпечної відстані система активує комплекс світло-звукової сигналізації. Користувачеві доступний гнучкий інтерфейс керування для корекції робочих режимів та вхідних величин (зокрема одиниць виміру та порогових значень наближення). Для запобігання втраті даних при відключенні живлення всі параметри фіксуються в енергонезалежній пам'яті (EEPROM), а стабільність автономної роботи контролюється вбудованим модулем моніторингу заряду батареї.

Пристрій оснащено символьним РК-екраном для моніторингу режимів роботи, виявлених перешкод та поточних налаштувань. Система керування підтримує два типи регулювання контрольної відстані: швидке (за допомогою механічного перемикача) та високоточне (через інтерфейс меню). Окрему увагу приділено енергозбереженню — користувач може вимикати дисплей для подовження часу автономної роботи.

Розроблений детектор перешкод є економічно вигідним рішенням, що за функціональним наповненням перевершує існуючі аналоги. Ключовими особливостями пристрою є вбудована система керування через меню, підтримка декількох робочих режимів та автоматичне збереження налаштувань у незалежній пам'яті. Окрім візуальної та звукової індикації, що базується на програмних алгоритмах, пристрій підтримує вивід даних через послідовний порт для інтеграції з іншими системами.

При проектуванні пристрою обрано сучасну архітектуру, побудовану на базі мікроконтролера ATmega328 (Arduino Uno). Вимірювання відстані до об'єктів здійснюється за допомогою ультразвукового модуля HC-SR04, а взаємодія з користувачем реалізована через символьний дисплей стандартної конфігурації 16x2.

Для реалізації детектора перешкод було виконано повний цикл проектування: від розробки принципової схеми до її віртуального тестування в пакеті Proteus. Програмне забезпечення для керування пристроєм створено на мові C в інструментарії Arduino IDE. Алгоритмічна база дозволяє ефективно використовувати ресурси мікроконтролерів AVR для виконання поставлених задач.

Розроблено спеціалізовані програмні бібліотеки для взаємодії з ультразвуковим датчиком HC-SR04, а також реалізовано систему адаптивного меню для керування параметрами пристрою.

Процес функціонування детектора було успішно змодельовано в середовищі Proteus ISIS. Отримані результати підтвердили повну відповідність розроблених програмних бібліотек та загальної логіки керування технічному завданню. Віртуальна апробація довела працездатність пристрою в різних експлуатаційних режимах.

У процесі роботи над проектом поглиблено теоретичні знання та здобуто практичний досвід у сфері проектування цифрових мікропроцесорних систем. Зокрема, опановано методику побудови алгоритмів керування та специфіку налагодження програмного забезпечення для пристроїв на базі платформи Arduino.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Massimo Banzi, Michael Shiloh. Getting Started With Arduino: The Open Source Electronics Prototyping Platform. – Published by Make Community, LLC; 4th ed., 2022. – p. 285.
2. Sarful Hassan. Arduino Programming Essentials: A Practical Handbook for Makers, Engineers, and Students. – Independently published, 2024. – p. 308.
3. Margolis M. Arduino Cookbook. 2nd Ed. O'Reilly Media Inc., 2012. 724 p.
4. Williams E. Make: AVR Programming. Learning to Write Software for Hardware. Maker Media, Inc, 2014. 472 p.
5. Barrett S. F., Pack D. J. Atmel AVR Microcontroller Primer: Programming and Interfacing. Morgan & Claypool Publishers, 2008. 198 p.
6. Padrue J. C Programming for Microcontrollers. SmileyMicros, 2005. 300 p.
7. Norman Dunbar. Arduino Software Internals: A Complete Guide to How Your Arduino Language and Hardware Work Together (Maker Innovations Series). – Published by Apress (Kindle Edition), 2nd ed., 2024. – p. 717.
8. Jonathan Bartlett. Electronics for Beginners: A Practical Introduction to Schematics, Circuits, and Microcontrollers. – Published by Apress, 1st ed., 2020. – p. 537.
9. Michael Stone. The Complete Guide to Arduino Programming: Simple and Effective Methods to Learn Arduino Programming. – Published by Deal Michael David, 2023. – p. 98.
10. Jeremy Blum. Exploring Arduino: Tools and Techniques for Engineering Wizardry. – Published by Wiley, 2nd ed., 2019. – p. 512.
11. Elecia White. Making Embedded Systems: Design Patterns for Great Software. – Published by O'Reilly Media, 2nd ed., 2024. – p. 425.
12. Benjamin Spahic. Arduino Without Prior Knowledge: Create your own first project within 7 days (Become an Engineer Without Prior Knowledge). – Kindle Edition, 2020. – p. 113.

13. Harley Pebley. Hacker's Field Guide: Arduino Foundations: Basic electronics and software for success with Arduino projects. – Kindle Edition, 2021. – p. 106.
14. Blum Richard. Arduino Programming in 24 Hours, Sams Teach Yourself. – Sams Publishing; 1st ed., 2014. – p. 434.
15. ATmega328P Automotive Microcontrollers : Datasheet. Microchip Technology Inc. URL: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Data-sheet.pdf.
16. Hitachi HD44780U (LCD-II) : Dot Matrix Liquid Crystal Display Controller/Driver. Datasheet. URL: <https://circuitdigest.com/sites/default/files/HD44780U.pdf>.
17. Arduino IDE : Software. Official Website. URL: <https://www.arduino.cc/en/Main/Software>.
18. Proteus VSM : Labcenter Electronics. URL: <https://www.labcenter.com/downloads/>.
19. HC-SR04 Ultrasonic Sensor : Complete Guide for Arduino. URL: <https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04>.
20. Arduino Tutorials : Official Documentation. URL: <https://www.arduino.cc/en/Tutorial/HomePage>.

ДОДАТКИ

Додаток 1

Фрагмент програмної реалізації алгоритму роботи детектора перешкод (пристрою для уникнення дроном перешкод в замкнутих та задимлених приміщеннях)

```

/*****
* Проект: Цифровий пристрій для уникнення перешкод дроном в замкнутих та
задимлених приміщеннях на базі МК ATmega328P
* Призначення: Робота в обмеженому просторі та умовах задимлення
* Платформа: Arduino/ATmega328P (16 MHz)
* Датчик: Ultrasonic HC-SR04
*****/

#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include <EEPROM.h>
// Визначення портів підключення сенсора
#define TRIGGER_PIN 13
#define ECHO_PIN 2
// Конфігурація дисплея
const int RS = 12, EN = 11, D4 = 6, D5 = 5, D6 = 4, D7 = 3;
LiquidCrystal display(RS, EN, D4, D5, D6, D7);
// Користувацькі символи стану АКБ
byte battery_full[8] = {0xE, 0xE, 0x1F, 0x1F, 0x1F, 0x1F, 0x1F, 0x1F};
byte battery_empty[8] = {0xE, 0xE, 0x1B, 0x11, 0x11, 0x11, 0x1F, 0x1F};
// Призначення пінів периферії
const int BEEP_PIN = 10;
const int LED_LOW_BAT = 8; // Червоний
const int LED_DETECTED = 9; // Синій (перешкода виявлена)
const int LED_CRITICAL = 7; // Жовтий (небезпечна відстань)
const int BAT_VOLT_PIN = A4; // Моніторинг напруги
const int SELECTOR_PIN = A5; // Вибір дистанції
// Клавіатура керування
const int KEY_MENU = A0; const int KEY_UP = A1; const int KEY_DOWN = A2;
const int KEY_EXIT = A3;
// Глобальні змінні стану
float current_voltage = 0; volatile long raw_battery_val = 0; volatile byte selector_index = 1;
unsigned int obstacle_range = 0; unsigned int limit_distance = 400; unsigned int saved_limit = 100;
unsigned long manual_threshold = 51;
byte op_mode = 1; // 1: Work, 0: Stop
byte control_type = 0; // 0: OFF, 1: KN (Manual), 2: MP (EEPROM)
byte unit_format = 1; // 1: cm, 2: m, 3: in
// Масиви для інтерфейсу
const char* ctrl_labels[] = {"KN ", " MP ", " OFF"};
const char* unit_labels[] = {"cm ", " m ", "in "};
unsigned int range_presets[7] = {0, 25, 50, 100, 200, 300, 390};
// --- Допоміжні функції ---
void triggerSignal() {
    pinMode(BEEP_PIN, OUTPUT);
    analogWrite(BEEP_PIN, 127); // 50% duty cycle
    delay(1000); analogWrite(BEEP_PIN, 0); pinMode(BEEP_PIN, INPUT);
}

```

```

void flashLed(int pin, int duration = 1000) {
    digitalWrite(pin, HIGH); delay(duration); digitalWrite(pin, LOW); }
void checkPowerStatus() {
    int sensorValue=analogRead(BAT_VOLT_PIN); raw_battery_val=(sensorValue*100L)/1024;
    display.setCursor(15, 1);
    if (raw_battery_val >= 55) { // 5.5V threshold
        display.print(char(0)); } else {
        Serial.println(F("CRITICAL: BATTERY LOW!"));
        display.print(char(1));flashLed(LED_LOW_BAT); }
}
void updateSelector() {
    int val = analogRead(SELECTOR_PIN); byte level = byte((val * 50) / 1024);
    if (level >= 49) selector_index = 6;
    else if (level >= 41) selector_index = 5;
    else if (level >= 33) selector_index = 4;
    else if (level >= 25) selector_index = 3;
    else if (level >= 16) selector_index = 2;
    else selector_index = 1;
}
// --- Функції роботи з пам'яттю ---
void saveLongToEEPROM(int addr, long val) {
    for (int i = 0; i < 4; i++) { EEPROM.write(addr + i, (val >> (24 - i * 8)) & 0xFF);
    } }
long loadLongFromEEPROM(int addr) {
    long res = 0;
    for (int i = 0; i < 4; i++) { res = (res << 8) + EEPROM.read(addr + i); }
    return res; }
// --- Системне меню ---
void configureControlMode() {
    display.clear();
    int cursor_pos = 0;
    for (int i = 0; i < 3; i++) {
        display.setCursor(i * 5, 0); display.print(ctrl_labels[i]); }
    while (true) {
        display.setCursor(cursor_pos, 1); display.print("^");

        if (digitalRead(KEY_UP) == HIGH) {
            display.setCursor(cursor_pos, 1); display.print(" ");
            cursor_pos = (cursor_pos >= 10) ? 0 : cursor_pos + 5;
            delay(200);
        }
        if (digitalRead(KEY_MENU) == HIGH) {
            control_type = (cursor_pos == 0) ? 1 : (cursor_pos == 5 ? 2 : 0);
            EEPROM.write(1, control_type); return; }
        if (digitalRead(KEY_EXIT) == HIGH) return;
    } }
// --- Основні цикли ---
void setup() {
    DDRD |= 0x78; DDRB |= 0x18; // Налаштування портів
    pinMode(TRIGGER_PIN,OUTPUT); pinMode(ECHO_PIN,INPUT);
    pinMode(LED_LOW_BAT,OUTPUT); pinMode(LED_DETECTED, OUTPUT); pinMode(LED_CRITICAL,
    OUTPUT);
    display.createChar(0, battery_full); display.createChar(1, battery_empty);

```

```

control_type = EEPROM.read(1); manual_threshold = loadLongFromEEPROM(10);
Serial.begin(9600); display.begin(16, 2); display.setCursor(0, 0);
display.print(F("-SCANNER ACTIVE-")); triggerSignal(); display.clear();
}

void loop() {
  if (digitalRead(KEY_MENU) == HIGH) {
    delay(200); }
  // Робота з ультразвуком
  if (op_mode == 1) {
    digitalWrite(TRIGGER_PIN, LOW); delayMicroseconds(2);
    digitalWrite(TRIGGER_PIN, HIGH); delayMicroseconds(10);
    digitalWrite(TRIGGER_PIN, LOW);
    long duration = pulseIn(ECHO_PIN, HIGH);
    obstacle_range = duration / 58; // Розрахунок у см
  } else { obstacle_range = 999; }
  // Логіка сповіщення
  unsigned int active_limit = (control_type == 1) ? range_presets[selector_index] :
    (control_type == 2 ? manual_threshold : 0);
  display.setCursor(0, 0);
  if (obstacle_range < 400) {
    display.print(F("DIST: ")); display.print(obstacle_range); display.print(F("cm "));
    flashLed(LED_DETECTED, 100);

    if (obstacle_range <= active_limit && control_type != 0) {
      display.setCursor(0, 1);
      display.print(F("!!! DANGER !!! "));
      digitalWrite(LED_CRITICAL, HIGH);
      triggerSignal();
    } else {
      digitalWrite(LED_CRITICAL, LOW);
    }
  } else {
    display.print(F("SEARCHING... "));
  }
  checkPowerStatus();
  updateSelector();
  delay(200);
}

```