

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ФАХОВИЙ БІЗНЕС-КОЛЕДЖ
Циклова комісія комп'ютерної інженерії та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему

**АПАРАТНИЙ ТРЕНАЖЕР ДЛЯ ВИВЧЕННЯ МІКРОПРОЦЕСОРНИХ
СИСТЕМ**

Виконав:

студент групи 2К-22

Спеціальності

123 Комп'ютерна інженерія

Віталій КРІТ

Керівник:

Наталя ФАЛЬЧЕНКО

Черкаси 2026

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ФАХОВИЙ БІЗНЕС-КОЛЕДЖ

Циклова комісія комп'ютерної інженерії та інформаційних технологій

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувачка циклової комісії КІ та ІТ

_____ Наталя ФАЛЬЧЕНКО

(підпис)

« _____ » _____ 2025 р.

З А В Д А Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

_____ Кроту Віталію Євгеновичу

1. Тема кваліфікаційної роботи Апаратний тренажер для вивчення мікропроцесорних систем

Керівник роботи Фальченко Наталя Григорівна

затверджені наказом закладу фахової передвищої освіти від 06.10.2025 року № 84/1-У.

2. Строк подання студентом кваліфікаційної роботи 08.06.2026

3. Вихідні дані до кваліфікаційної роботи: Концепція інтерактивного навчального середовища на базі мікроконтролера ATmega328P; принципи архітектурної побудови тренажера; способи інтеграції вузлів цифрового та аналогового введення; методи моделювання логіки роботи пристрою в середовищі хмарної симуляції Autodesk Tinkercad; можливості реалізації алгоритмів обробки зовнішніх переривань, таймерів-лічильників та широтно-імпульсної модуляції; методи верифікації працездатності програмно-апаратного комплексу в умовах імітаційного моделювання.

4. Зміст кваліфікаційної роботи: Дослідження області сучасних засобів навчання та обґрунтування доцільності використання симуляційних платформ у вивченні мікропроцесорної техніки; порівняльна характеристика апаратних платформ та вибір оптимального складу периферійних модулів; проектування електричної схеми тренажера; розробка комплексу навчальних мікропрограм для демонстрації базових функцій мікроконтролера; програмування алгоритмів первинної обробки сигналів; аналіз результатів віртуальних випробувань та оцінка стабільності коду при імітації критичних режимів роботи.

5. Дата видачі завдання 15.09.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів	Примітка про виконання з підписами керівника і студента
1	Вступ	14.10.2025	
2	Розділ 1. Огляд та аналіз методів і засобів навчання мікропроцесорним системам	09.12.2025	
3	Розділ 2. Розробка функціональної моделі та апаратної структури тренажера	10.03.2026	
4	Розділ 3. Проектування та практична реалізація тренажера	28.04.2026	
5	Висновки	12.05.2026	
6	Оформлення кваліфікаційної роботи (чистовий варіант)	26.05.2026	
7	Перевірка кваліфікаційної роботи на наявність ознак плагіату (за 10 днів до захисту)	02.06.2026	
8	Подання кваліфікаційної роботи на затвердження завідувачу ЦК (за 7 днів до захисту)	08.06.2026	

Студент _____
(підпис)

Віталій КРІТ

Керівник роботи _____
(підпис)

Наталя ФАЛЬЧЕНКО

АНОТАЦІЯ

Кваліфікаційну роботу присвячено розробці інтерактивного навчального тренажера на базі платформи Arduino. Тренажер призначений для навчання принципам функціонування сучасних мікропроцесорних систем. У роботі проведено аналіз актуальних засобів навчання та обґрунтовано вибір середовища Autodesk Tinkercad як основного інструменту для імітаційного моделювання програмно-апаратних комплексів.

Основна частина роботи містить опис проектування електричної принципової схеми, що включає мікроконтролер ATmega328P, модулі введення (кнопки, потенціометри) та системи візуалізації даних (LCD-дисплеї, світлодіодні індикатори). Розроблено вбудоване програмне забезпечення, яке демонструє роботу з цифровими та аналоговими сигналами, перериваннями, таймерами та протоколами передачі даних (I2C, UART).

Практичне значення отриманих результатів полягає у створенні універсальної віртуальної моделі тренажера, яка дозволяє проводити налагодження мікропроцесорних пристроїв без необхідності використання фізичного обладнання на початкових етапах навчання. В ході роботи проведено тестування розроблених алгоритмів у симуляційному середовищі, що підтвердило коректність їх роботи.

Ключові слова: мікропроцесорні системи, Arduino, Tinkercad, імітаційне моделювання, вбудоване програмне забезпечення, мікроконтролер, комп'ютерна інженерія.

ABSTRACT

The qualification work is devoted to the development of an interactive educational simulator based on the Arduino platform. The simulator is intended for teaching the principles of functioning of modern microprocessor systems. The paper analyzes current training tools and substantiates the choice of the Autodesk Tinkercad environment as the main tool for simulation modeling of hardware and software complexes.

The main part of the work contains a description of the design of the electrical schematic diagram, which includes the ATmega328P microcontroller, input modules (buttons, potentiometers) and data visualization systems (LCD displays, LED indicators). Built-in software has been developed that demonstrates working with digital and analog signals, interrupts, timers, and data transfer protocols (I2C, UART).

The practical significance of the obtained results lies in the creation of a universal virtual model of the simulator, which allows debugging of microprocessor devices without the need to use physical equipment at the initial stages of training. During the work, the developed algorithms were tested in a simulation environment, which confirmed the correctness of their operation.

Keywords: microprocessor systems, Arduino, Tinkercad, simulation modeling, embedded software, microcontroller, computer engineering.

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1 ОГЛЯД ТА АНАЛІЗ МЕТОДІВ І ЗАСОБІВ НАВЧАННЯ МІКРОПРОЦЕСОРНИМ СИСТЕМАМ.....	6
1. 1 Огляд існуючих інтернет-ресурсів та платформ, що характеризують предметну область.....	6
1. 2 Теоретичні основи побудови мікропроцесорних систем на базі архітектури AVR.....	10
1. 3 Постановка задачі на розробку апаратного тренажера.....	14
РОЗДІЛ 2 РОЗРОБКА ФУНКЦІОНАЛЬНОЇ МОДЕЛІ ТА АПАРАТНОЇ СТРУКТУРИ ТРЕНАЖЕРА.....	16
2. 1 Вибір інструментів для моделі тренажера.....	16
2. 2 Розробка схеми тренажера в графічному середовищі Draw.io.....	18
2. 3 Опис архітектури та електричної структури розробленого стенда.....	21
РОЗДІЛ 3 ПРОЄКТУВАННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ ТРЕНАЖЕРА.....	26
3. 1 Архітектурне проєктування зв'язків мікроконтролера з периферією.....	26
3. 2 Інструментальні засоби реалізації тренажера MP TRAINER.....	30
3. 3 Фізична реалізація моделі тренажера.....	33
3. 4 Тестування функціональності та аналіз результатів роботи MP TRAINER.....	37
ВИСНОВКИ.....	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	43
ДОДАТКИ.....	45

ВСТУП

Актуальність теми. Сучасний етап розвитку глобальної цифрової економіки базуються на тотальній інтеграції мікропроцесорних систем у всі ланки людської діяльності. Мікроконтролери сьогодні є "нервовою системою" сучасного світу, забезпечуючи функціонування інтелектуальних мереж, автономних транспортних засобів, складних медичних комплексів та систем промислової автоматизації. Для фахівця в галузі комп'ютерної інженерії здатність проектувати, програмувати та налагоджувати такі системи є фундаментальною компетенцією, що визначає його професійну спроможність на ринку праці.

Попри стрімкий прогрес технологій, процес підготовки інженерних кадрів стикається з серйозним викликом: необхідністю швидкої адаптації до нових архітектур при збереженні глибокого розуміння базових принципів роботи обчислювальних систем. Мікропроцесорна техніка є складною дисципліною, де теоретичні знання про логічні рівні, архітектуру регістрів, роботу переривань та організацію пам'яті залишаються абстрактними без належної практичної верифікації. Традиційні методи навчання, що покладаються на симулятори або закриті промислові стенди, часто не дають можливості повноцінно дослідити "фізику" процесів та взаємодію заліза з програмним кодом у реальному часі.

Саме тому розробка апаратного тренажера на базі відкритої архітектури є надзвичайно актуальним завданням. Використання платформи Arduino Uno, серцем якої є мікроконтролер ATmega328P, дозволяє створити універсальне середовище для вивчення класичної RISC-архітектури. Такий вибір обґрунтований тим, що архітектура AVR є ідеальним балансом між простотою розуміння та потужністю функціональних можливостей, що включають аналогово-цифрові перетворювачі (АЦП), широтно-імпульсну модуляцію (ШИМ), а також розгалужену систему апаратних інтерфейсів зв'язку.

Важливим аспектом даної роботи є інтеграція сучасних засобів програмної підтримки проектування. Для забезпечення високої надійності кінцевого

пристрою в роботі застосовується комплексний підхід до моделювання. Використання хмарного середовища Tinkercad дозволяє здійснювати первинну перевірку концепцій та візуалізацію схем на етапі прототипування. Для розробки структурних схем взаємодії вузлів та побудови логічних алгоритмів роботи програмного забезпечення використовується професійний графічний інструментарій Draw.io (diagrams.net), що дозволяє створювати схеми згідно з технічними стандартами. Безпосередня розробка вбудованого програмного забезпечення (Firmware) здійснюється в середовищі Arduino IDE, що дозволяє використовувати потужні бібліотеки та засоби налагодження коду.

Метою кваліфікаційної роботи є створення інструментарію для вивчення мікропроцесорних систем, який поєднує в собі наочність фізичних процесів та точність програмних алгоритмів керування.

Для досягнення поставленої мети визначено наступні завдання:

1. Дослідити теоретичні засади функціонування мікроконтролерів родини AVR, приділивши особливу увагу архітектурі ATmega328P, його системі команд, внутрішнім регістрам та периферійним пристроям.
2. Сформулювати технічні та функціональні вимоги до апаратного тренажера, що розробляється, враховуючи необхідність реалізації різноманітних сценаріїв взаємодії з периферією.
3. Обґрунтувати вибір елементної бази, включаючи сенсори, модулі індикації та пристрої введення даних, виходячи з критеріїв сумісності, енергоефективності та надійності.
4. Спроектувати електричні схеми взаємодії центрального контролера з зовнішніми вузлами та виконати їх верифікацію в середовищах Tinkercad та Draw.io.
5. Розробити алгоритмічне забезпечення та відповідний програмний код, що реалізує функції тренажера, забезпечуючи стабільну роботу системи в реальному часі.

6. Провести комплексне тестування розробленого апаратного стенда, виконати аналіз отриманих даних та оцінити практичну ефективність застосування тренажера для вирішення інженерних завдань.

Об'єкт дослідження – сфера проектування, моделювання та функціонування сучасних мікропроцесорних систем на базі RISC-архітектури.

Предмет дослідження – методи створення навчального тренажера на базі платформи Arduino Uno, що забезпечує взаємодію з цифровими та аналоговими пристроями.

Практичне впровадження розробленого тренажера дозволяє модернізувати матеріально-технічну базу навчальних лабораторій, забезпечити індивідуалізацію процесу навчання та сприяти формуванню у майбутніх інженерів стійких навичок розробки складних вбудованих систем.

Апробація результатів кваліфікаційної роботи здійснювалася шляхом їх представлення на XVIII Всеукраїнській науково-практичній конференції «Тенденції розвитку ІТ-технологій в Україні», 23-24 квітня, м. Черкаси. Тема доповіді: "Особливості створення навчального апаратного тренажера мікропроцесорної техніки".

РОЗДІЛ 1 ОГЛЯД ТА АНАЛІЗ МЕТОДІВ І ЗАСОБІВ НАВЧАННЯ МІКРОПРОЦЕСОРНИМ СИСТЕМАМ

1.1 Огляд існуючих інтернет-ресурсів та платформ, що характеризують предметну область

Проектування сучасних мікропроцесорних систем є багатограним науково-технічним процесом, який вимагає залучення різномірних інформаційних джерел та спеціалізованих хмарних середовищ розробки [2]. Предметна область створення апаратних тренажерів на базі 8-бітних обчислювальних систем охоплює широкий спектр вебресурсів, які забезпечують розробника критично важливою технічною документацією, засобами інтерактивної верифікації проектних рішень та інструментарієм для побудови стандартизованої графічної документації [9; 14].

Екосистема Arduino (Arduino.cc) виступає як базовий інформаційний фундамент, що забезпечує глобальну стандартизацію в галузі вбудованих рішень. Важливість цього ресурсу для розробки тренажера полягає не лише у доступі до програмного коду, а й у формуванні єдиного методологічного підходу до архітектурного програмування мікроконтролерів. Аналіз цього ресурсу дозволяє виокремити наступні стратегічно важливі компоненти для розробки:

Специфікації апаратного рівня (Datasheets) цей ресурс надає прямий доступ до першоджерел технічних параметрів мікроконтролерів (зокрема серії ATmega). Це дозволяє здійснювати прецизійне проектування режимів енергоспоживання, конфігурувати порти введення-виведення з урахуванням навантажувальної здатності шин даних та розраховувати допустимі струми для захисту напівпровідникових переходів.

Централізований репозиторій програмних модулів використовує верифіковану бібліотеку для обміну даними через інтерфейси I2C, SPI та UART. Це мінімізує ймовірність виникнення логічних помилок при роботі з таймерами та регістрами спеціального призначення, що значно підвищує загальну

надійність взаємодії мікроконтролера з периферійними сенсорами та засобами візуалізації інформації.

Інженерна база знань глибокий аналіз описів мовних конструкцій дозволяє інтегрувати у тренажер методики прямого керування регістрами (Direct Port Manipulation), що забезпечує максимальну швидкодію системи в критичних режимах роботи.

Хмарне середовище Tinkercad (від Autodesk) інтегрується в процес розробки як основний засіб віртуального прототипування та попередньої апаратної верифікації. У контексті комп'ютерної інженерії цей ресурс забезпечує безпечне моделювання електронних вузлів будь-якої складності [1, с. 112]. Технічний аналіз платформи дозволяє виділити наступні переваги:

Динамічна симуляція фізичних процесів дає можливість візуального спостереження за реакцією системи на зміну зовнішніх факторів у реальному часі. Платформа дозволяє емулювати роботу аналогових та цифрових датчиків, що дає змогу відлагодити логіку алгоритму (наприклад, фільтрацію шумів або обробку подій) ще до моменту закупівлі та фізичної збірки компонентів.

Інструментальний моніторинг параметрів наявності віртуальних мультиметрів для вимірювання напруги та струму в різних вузлах схеми, а також осцилографів для аналізу часових діаграм сигналів (наприклад, ШІМ-сигналу для керування яскравістю). Це дозволяє контролювати рівні логічних сигналів, що є обов'язковим етапом професійного проектування цифрових пристроїв.

Аналіз електричних конфліктів система автоматично попереджає про потенційно небезпечні режими роботи (перевищення робочої напруги на виводах, коротке замикання), що забезпечує цілісність майбутнього апаратного прототипу та економить матеріальні ресурси [10].

Спеціалізоване середовище Draw.io (diagrams.net) розглядається як фундаментальний інструментарій для формування проектної документації високої якості [18]. З огляду на вимоги до оформлення пояснювальних записок кваліфікаційних робіт, даний ресурс забезпечує:

Проектування структурних та функціональних схем візуалізація складних ієрархічних зв'язків між центральним процесором та периферійним обладнанням. Використання бібліотек стандартних символів дозволяє створювати схеми, що повністю відповідають вимогам ДСТУ щодо графічних позначень електронних компонентів.

Моделювання алгоритмічних процесів побудова логічних блок-схем, що описують послідовність виконання операцій програмним забезпеченням. Це дозволяє детально розписати кожен цикл роботи тренажера - від процедури ініціалізації внутрішніх модулів мікроконтролера до алгоритмів обробки зовнішніх переривань від кнопок або сенсорів.

Забезпечення професійної якості графіки векторна природа середовища гарантує ідеальну чіткість ілюстрацій при будь-якому масштабуванні. Можливість точного вирівнювання елементів та використання сітки дозволяє створювати документацію, яка виглядає професійно та легко сприймається при рецензуванні роботи.

Саме тому розробка апаратного тренажера на базі відкритої архітектури є надзвичайно актуальним завданням. Використання платформи Arduino Uno, серцем якої є мікроконтролер ATmega328P, дозволяє створити універсальне середовище для вивчення класичної RISC-архітектури. Такий вибір обґрунтований тим, що архітектура AVR є ідеальним балансом між простотою розуміння та потужністю функціональних можливостей, що включають аналогово-цифрові перетворювачі (АЦП), широтно-імпульсну модуляцію (ШИМ), а також розгалужену систему апаратних інтерфейсів зв'язку.

Для реалізації поставленого завдання необхідно було обрати оптимальне програмне середовище, яке б дозволило не лише проектувати, а й проводити симуляцію роботи обраної платформи. З метою обґрунтування вибору інструментарію для моделювання мікропроцесорної системи було проведено аналіз існуючих аналогів, результати якого наведено в таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз середовищ моделювання мікропроцесорних систем

Середовище	Переваги	Недоліки
Autodesk Tinkercad (Обране)	Безкоштовний, працює в браузері, має вбудований емулятор коду Arduino та візуальне підключення компонентів.	Обмежений набір складних радіодеталей, не підходить для професійного проектування друкованих плат.
Wokwi	Дуже швидкий симулятор, підтримує сучасні мікроконтролери (ESP32, Pi Pico), відмінна підтримка бібліотек Arduino.	Тільки текстовий редактор коду (немає візуальних блоків), менш інтуїтивний інтерфейс для новачків.
Proteus	Професійний інструмент з величезною базою компонентів та точним аналізом сигналів.	Платна ліцензія, складний інтерфейс, потребує встановлення на ПК.
Fritzing	Гарна візуалізація, ідеально підходить для створення схем на макетних платах.	Платне завантаження, слабкі можливості симуляції коду (переважно для малювання).

Окрім апаратної симуляції, важливе значення має якісна візуалізація алгоритмів керування та структури зв'язків. Для вибору графічного редактора, що відповідає технічним стандартам та вимогам зручності, було виконано порівняння популярних сервісів, представлено в таблиці 1.2.

Важливим аспектом даної роботи є інтеграція сучасних засобів програмної підтримки проектування. Для забезпечення високої надійності кінцевого пристрою в роботі застосовується комплексний підхід до моделювання. Використання хмарного середовища Tinkercad дозволяє здійснювати первинну перевірку концепцій та візуалізацію схем на етапі прототипування. Для розробки структурних схем взаємодії вузлів та побудови логічних алгоритмів роботи програмного забезпечення використовується професійний графічний інструментарій Draw.io, що дозволяє створювати схеми згідно з технічними стандартами. Безпосередня розробка вбудованого програмного забезпечення

(Firmware) здійснюється в середовищі Arduino IDE, що дозволяє використовувати потужні бібліотеки та засоби налагодження коду.

Таблиця 1.2 – Порівняння інструментів для побудови графічних схем та алгоритмів

Інструмент	Переваги	Недоліки
Draw.io (Обране)	Повністю безкоштовний, інтеграція з Google Drive, велика бібліотека елементів для блок-схем.	Деяко застарілий дизайн інтерфейсу, може працювати повільно з дуже великими схемами.
Lucidchart	Дуже зручний та сучасний інтерфейс, багато готових шаблонів, гарна спільна робота.	Безкоштовна версія сильно обмежена за кількістю елементів на одній схемі.
Microsoft Visio	Потужний професійний стандарт, величезна кількість інструментів для інженерної графіки.	Дуже дорогий, працює переважно на Windows, складний для швидких задач.
Miro	Чудово підходить для мозкового штурму та творчих схем, гарний візуал.	Не є спеціалізованим інструментом для алгоритмів, важко дотримуватися стандартів ДСТУ/ISO.

Таким чином, системний аналіз зазначених інтернет-платформ підтверджує можливість реалізації наскрізного циклу розробки апаратного тренажера – від теоретичного дослідження документації до практичного моделювання та детального документування результатів проектування.

1.2 Теоретичні основи побудови мікропроцесорних систем на базі архітектури AVR

Проектування ефективних засобів навчання у галузі комп'ютерної інженерії базується на глибокому аналізі обчислювальних потужностей та внутрішньої структури керуючого пристрою. В основі розроблюваного апаратного тренажера лежить 8-бітна архітектура AVR, яка на сьогодні є де-факто стандартом для побудови малопотужних вбудованих систем реального

часу завдяки оптимальному балансу між продуктивністю та складністю програмної реалізації.

Концептуальні засади RISC-архітектури. Фундаментом побудови систем на базі AVR є концепція RISC (Reduced Instruction Set Computer) - обчислювача зі скороченим набором команд. На відміну від складних CISC-систем, мікроконтролери AVR орієнтовані на виконання простих інструкцій фіксованої довжини. Це дозволяє реалізувати ефективне однорівневе конвеєрне виконання, де підготовка наступної команди відбувається одночасно з виконанням поточної. У контексті функціонування тренажера це забезпечує миттєву реакцію на зовнішні подразники (наприклад, натискання кнопок або спрацювання сенсорів), оскільки більшість операцій виконуються за один такт синхронізації.

Особливості Гарвардської моделі організації шин. Ключовою теоретичною особливістю побудови мікропроцесорних систем AVR є використання Гарвардської архітектури. Її принципова відмінність від класичної архітектури Фон Неймана полягає у повному фізичному розділенні пам'яті програм та пам'яті даних.

Шина програм призначена для зчитування інструкцій з Flash-пам'яті.

Шина даних забезпечує обмін інформацією між регістрами та оперативною пам'яттю.

Така організація дозволяє уникнути «пляшкового шийка» при передачі даних, оскільки процесор може одночасно звертатися до обох видів пам'яті. Для апаратного тренажера це означає стабільну роботу алгоритмів обробки сигналів без затримок, характерних для систем з загальною шиною.

Регістрова структура та арифметико-логічний пристрій (АЛП). Центральним вузлом обробки інформації в ATmega328P є 8-бітний АЛП, який безпосередньо взаємодіє з файлом регістрів загального призначення. Регістрова пам'ять складається з 32-х 8-бітних регістрів (R0-R31). Особливістю є те, що останні шість регістрів (R26 - R31) можуть об'єднуватися в три 16-бітні вказівні регістри (X, Y, Z), що критично важливо для адресації пам'яті при роботі з

великими масивами даних або складними структурами у програмному коді тренажера [15, с. 54-59].

Ієрархічна структура пам'яті та її функціональне призначення:

Flash-пам'ять (пам'ять програм): енергонезалежна область обсягом 32 КБ, де зберігається Firmware тренажера. Вона організована як послідовність 16-бітних слів, що відповідає довжині більшості інструкцій AVR.

SRAM (статична оперативна пам'ять): область обсягом 2 КБ, що використовується для зберігання тимчасових змінних, стеків та поточних значень сенсорів. Вона забезпечує високу швидкість доступу, що є необхідним для динамічних обчислень.

EEPROM: окремий сегмент пам'яті (1 КБ) для довготривалого зберігання даних, які не повинні зникати після вимкнення живлення. У тренажері це може використовуватися для фіксації логів помилок або налаштувань користувача.

Функціонування апаратного тренажера як інтерактивного пристрою базується на здатності мікроконтролера взаємодіяти із зовнішнім середовищем через вбудовані периферійні модулі. В архітектурі AVR ці модулі інтегровані безпосередньо в кристал, що забезпечує високу швидкість обміну даними та мінімальні апаратні затримки.

Система переривань (Interrupt Management). Для забезпечення миттєвої реакції тренажера на дії користувача (наприклад, натискання «тривожної» кнопки або спрацювання датчика) використовується механізм апаратних переривань.

Теоретична суть переривання полягає у здатності процесора тимчасово призупинити виконання основного циклу програми, зберегти поточний стан лічильника команд у стеку та перейти до виконання спеціальної підпрограми обробки (ISR). Після завершення обробки події процесор автоматично повертається до перерваної задачі. Така багатозадачність на апаратному рівні дозволяє тренажеру стабільно працювати в режимі реального часу, не пропускаючи важливі вхідні сигнали.

Аналогово-цифрове перетворення (АЦП). Більшість фізичних процесів, які досліджуються за допомогою тренажера (температура, освітленість, напруга), мають аналогову природу. Оскільки мікроконтролер є цифровим пристроєм, в архітектурі ATmega328P передбачено 10-бітний багатоканальний АЦП.

Принцип роботи модуль АЦП здійснює квантування вхідної напруги та перетворює її у цифрове значення (від 0 до 1023).

Значення для тренажера це дозволяє студенту вивчати методику цифрової обробки сигналів, реалізовувати порогові значення спрацювання сигналізації та калібрувати датчики програмним шляхом.

Таймери-лічильники та широтно-імпульсна модуляція (ШІМ). Важливою складовою мікропроцесорних систем є блоки таймерів, які працюють незалежно від центрального процесора. Вони виконують наступні критичні функції:

Формування часових інтервалів забезпечення точних затримок, необхідних для коректного опитування датчиків.

Генерація сигналів ШІМ (PWM) метод керування середньою напругою на виході шляхом зміни тривалості імпульсів. У тренажері ШІМ використовується для регулювання яскравості світлодіодних індикаторів або керування тональністю звукового випромінювача (бузера).

Механізми керування енергоспоживанням. Однією з фундаментальних переваг архітектури AVR, що робить її ідеальною для побудови портативних навчальних тренажерів, є розвинена система енергозбереження. Мікроконтролер ATmega328P підтримує шість різних режимів «сну» (Sleep Modes), які дозволяють програмно вимикати невикористовувані периферійні модулі.

Режим Idle зупиняє роботу центрального процесора, але залишає активними таймери, АЦП та систему переривань. Це дозволяє тренажеру миттєво «прокидатися» при натисканні клавіш.

Режим Power-down забезпечує мінімальне споживання струму (до 0.1 мкА), зберігаючи при цьому вміст регістрів.

Така гнучкість у керуванні живленням дозволяє розробнику створювати енергоефективні алгоритми, що є важливою компетенцією для сучасного фахівця з комп'ютерної інженерії. Вивчення цих режимів у межах роботи з тренажером дає студентам розуміння принципів розробки автономних IoT-пристроїв (інтернету речей).

Завершуючи теоретичний аналіз архітектури AVR, слід зазначити, що саме синергія потужного обчислювального ядра та розгалуженої системи периферійних пристроїв робить мікроконтролер ATmega328P ідеальною платформою для створення навчальних засобів. Описані архітектурні особливості стають базисом для подальшого проектування конкретної апаратної конфігурації тренажера, що буде детально розглянуто у наступних розділах роботи.

1.3 Постановка задачі на розробку апаратного тренажера

Спираючись на проведений у попередніх розділах теоретичний аналіз архітектури AVR та огляд сучасних середовищ моделювання, необхідно сформулювати розгорнуту постановку задачі. Основна вимога до розроблюваного апаратного тренажера полягає у створенні цілісної системи, яка дозволяє візуалізувати внутрішні процеси обробки даних мікроконтролером.

Для досягнення високої точності моделювання та зручності використання, до апаратної конфігурації висуваються наступні вимоги:

Центральний обчислювальний вузол використання платформи на базі ATmega328P, що забезпечує апаратну підтримку ШІМ та АЦП на необхідній кількості каналів.

Інтерфейсна шина обов'язкова підтримка протоколу I2C (Inter-Integrated Circuit) для підключення модулів відображення інформації. Це дозволяє вивчити принципи адресації пристроїв на шині та методику послідовної передачі даних.

Вхідна периферія інтеграція компонентів для моделювання різних типів сигналів:

1. Дискретні входи: кнопки для імітації подій та переривань.

2. Аналогові входи: резистивні дільники та фотосенсиори для імітації зміни фізичних параметрів (освітленість, температура, положення валу).

Вихідна периферія наявність світло-звукової індикації для зворотного зв'язку з користувачем, що дозволяє відлагодити логіку спрацювання сигналізації в різних режимах.

Програмний код, що керує тренажером, повинен бути спроектований з урахуванням наступних інженерних задач:

Розробка драйверів обробки створення підпрограм для коректної інтерпретації даних з АЦП (фільтрація шумів, калібрування діапазонів).

Оптимізація виводу даних розробка алгоритму оновлення інформації на LCD-дисплеї, який не блокує основний цикл виконання програми (використання таймерів або неблокуючих затримок).

Обробка аварійних станів створення пріоритетної логіки для звукової сигналізації при виході контрольованих параметрів за межі встановлених норм.

Оскільки тренажер є навчальним засобом, постановка задачі передбачає розробку повного пакету технічної графіки:

Побудова структурної схеми, що демонструє потоки інформації між ядром та периферією.

Розробка принципової електричної схеми, де враховані всі необхідні пасивні компоненти (резистори, конденсатори) для стабільної роботи ланцюгів.

Створення логічних блок-схем алгоритму, що описують поведінку системи при різних сценаріях експлуатації.

Пристрій має бути спроектований з використанням доступної елементної бази, що забезпечує можливість його масштабування та ремонтпридатності в умовах навчальної лабораторії. Інтерфейс взаємодії повинен бути лаконічним, щоб студент міг зосередитися на вивченні коду, а не на складнощах керування приладом.

У даному розділі кваліфікаційної роботи досліджено сучасний стан розвитку навчальних технологій у галузі мікропроцесорної техніки. У процесі аналізу розглядаються існуючі програмні та апаратні платформи, які є

фундаментальними для проектування вбудованих систем. Особлива увага приділяється вивченню архітектурних особливостей мікроконтролерів.

РОЗДІЛ 2 РОЗРОБКА ФУНКЦІОНАЛЬНОЇ МОДЕЛІ ТА АПАРАТНОЇ СТРУКТУРИ ТРЕНАЖЕРА

2.1 Вибір інструментів для моделі тренажера

Для успішної реалізації апаратного тренажера було обрано середовище, яке може забезпечити високу точність моделювання електричних процесів та дозволяє проводити гнучке програмування мікроконтролерів. В сучасній інженерній практиці існує ряд рішень (Proteus, Fritzing, Wokwi), проте для розробки навчально-демонстраційного стенда було обрано хмарну платформу Autodesk Tinkercad Circuits.

Вибір даної платформи зумовлений необхідністю створення універсального інструменту, який поєднує в собі низький поріг входження для студентів та достатню технічну потужність для відтворення складних логічних операцій.

Робота в середовищі базується на принципі «Drag-and-Drop» (перетягування елементів). Це дозволяє розробнику обирати компоненти зі стандартної розширеної бібліотеки, яка включає не лише пасивні елементи (резистори, конденсатори), а й складні інтегровані модулі: LCD-дисплеї 1602, цифрові датчики, сервоприводи та засоби індикації. Система автоматично відстежує зв'язки між компонентами, імітуючи реальний процес збирання електронного пристрою на макетній платі (breadboard). Це дає можливість розробнику проводити віртуальний монтаж, максимально наближений до фізичного прототипування.

Загальний вигляд робочого простору середовища Tinkercad із базовим набором елементів для розробки тренажера, включаючи мікроконтролерну плату та макетну плату для тестування з'єднань, представлено на рис. 2.1.

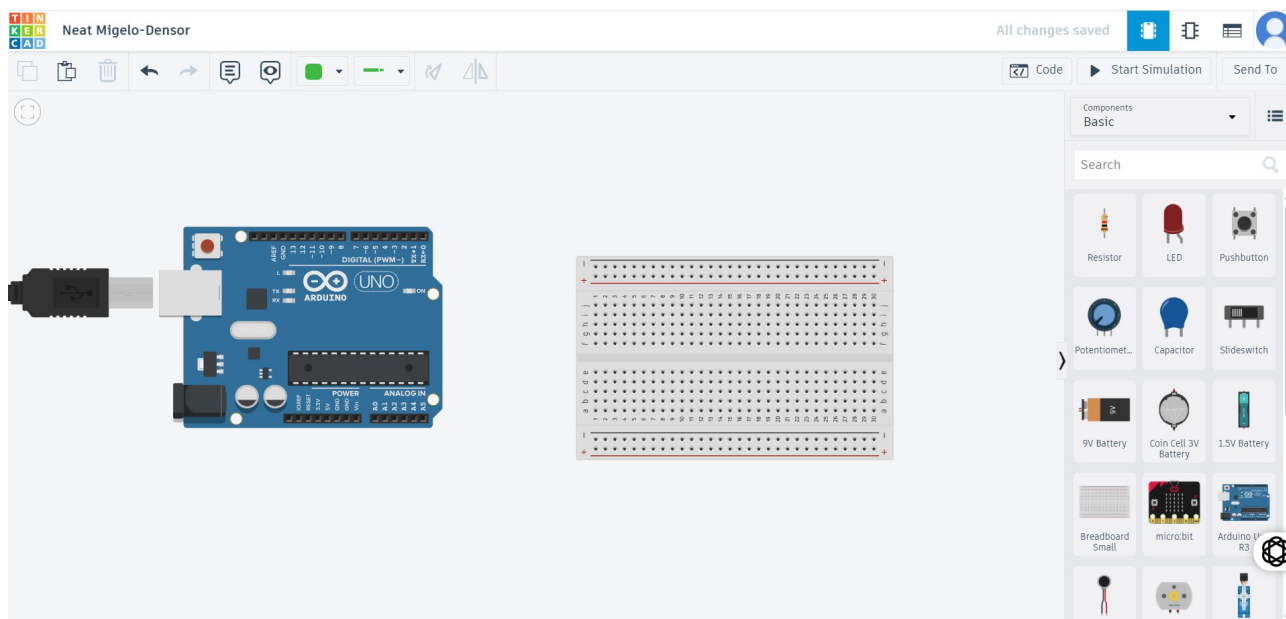


Рисунок 2.1 – Загальний вигляд інтерфейсу середовища розробки Autodesk Tinkercad

Зображення демонструє панель інструментів та бібліотеку компонентів, що дозволяє оперативно обирати необхідні модулі для побудови архітектури системи.

Головним принципом роботи Tinkercad є динамічне обчислення фізичних величин на основі фундаментальних законів електротехніки (закони Ома та Кірхгофа) [4]. Під час запуску симуляції система розраховує напругу, струм та опір у кожній точці схеми. Якщо параметри ланцюга порушують допустимі межі (наприклад, подача 5V на світлодіод без обмежувального резистора або коротке замикання), система візуально демонструє «вихід компонента з ладу». Це є критично важливим для навчального тренажера, оскільки дозволяє формувати у користувача розуміння безпечних режимів роботи обладнання.

В основі моделювання Arduino Uno в Tinkercad лежить повна програмна емуляція архітектури AVR, зокрема мікроконтролера ATmega328P [11]. Це означає, що віртуальний контролер має ідентичну фізичній платі кількість портів введення-виведення (цифрові D0-D13 та аналогові A0-A5), аналого-цифрових перетворювачів (АЦП) та внутрішніх таймерів. Програма, написана

розробником, виконується в симуляторі так само, як і на реальному кристалі, з урахуванням тактової частоти 16 МГц та обмежень оперативної пам'яті (SRAM).

Tinkercad надає дворівневий підхід до розробки програмного забезпечення [10].

Візуально-блокове програмування дозволяє швидко проектувати логіку для демонстраційних задач.

Текстовий редактор (C++ / Wiring) забезпечує використання професійних інструментів розробки. При створенні тренажера було використано саме текстовий режим, що дозволило інтегрувати складні бібліотеки, такі як LiquidCrystal для роботи з дисплеєм [12].

Важливою складовою є наявність віртуального послідовного порту (UART). Він виконує роль діагностичного інструменту, через який розробник може отримувати дані про стан регістрів, поточні значення з аналогових входів (потенціометрів) або текстові сповіщення про хід виконання алгоритму. Це дозволяє проводити глибоке налагодження коду без використання зовнішніх логічних аналізаторів.

Для перевірки коректності роботи апаратної структури Tinkercad пропонує використання віртуальних вимірювальних приладів. Наявність мультиметрів та осцилографів дозволяє візуалізувати форму сигналів (наприклад, ШІМ-сигнал для керування яскравістю або зумером), що замінює дорогу лабораторну базу та робить процес розробки тренажера економічно доцільним.

Вибір Autodesk Tinkercad як базового інструменту дозволив створити безпечну, інтерактивну та технічно точну модель тренажера. Це забезпечує повний цикл проектування - від електричної схеми до програмної реалізації - у єдиному віртуальному просторі, що є ідеальним для освітніх цілей у галузі комп'ютерної інженерії.

2.2 Розробка схеми тренажера в графічному середовищі Draw.io

Для забезпечення стабільного функціонування апаратного тренажера було розроблено алгоритм керування, що базується на циклічному опитуванні

периферійних пристроїв та обробці станів у реальному часі [18]. Структурну схему алгоритму наведено на рис. 2.2.

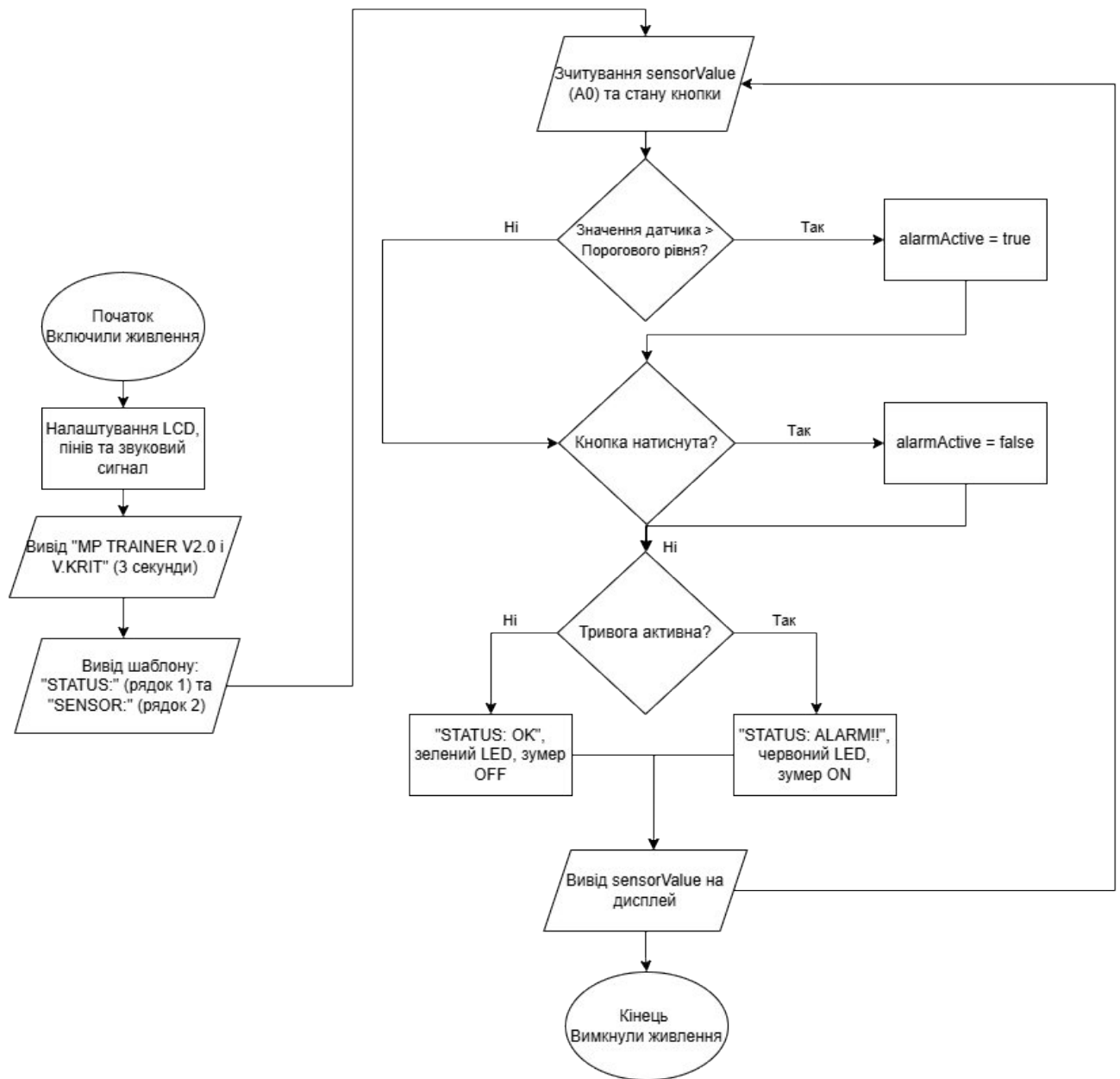


Рисунок – 2.2 Структурна схема моделі тренажера

При ввімкненні живлення мікроконтролер Arduino виконує налаштування LCD-дисплея, конфігурацію пінів введення-виведення та ініціалізацію звукового сигналу. Для ідентифікації пристрою передбачено вивід заставки з назвою проекту «MP TRAINER» та іменем розробника протягом 3 секунд. Після цього

на екран виводиться шаблон інтерфейсу: «STATUS:» у першому рядку та «SENSOR:» у другому.

Основний цикл роботи (loop) починається зі зчитування поточних даних з аналогового датчика (потенціометра) та перевірки стану кнопки скидання [17]. Алгоритм порівнює отримане значення датчика з попередньо встановленим пороговим рівнем (у даній реалізації - 800 одиниць АЦП).

Якщо значення перевищує поріг, змінна стану тривоги (alarmActive) набуває значення true.

Незалежно від показників датчика, система перевіряє стан кнопки: при її натисканні статус тривоги примусово скидається у значення false.

На фінальному кроці циклу алгоритм перевіряє поточне значення змінної alarmActive. У разі активної тривоги система вмикає червоний світлодіод, активує зумер та виводить повідомлення «ALARM!!». Якщо стан тривоги відсутній, активується зелений світлодіод, зумер вимикається, а на дисплей виводиться статус «ОК». Поточне значення показника датчика оновлюється на екрані в кожному циклі.

Розроблений алгоритм реалізує логіку «фіксації стану» (latch logic), що є критично важливим для систем безпеки, оскільки вимагає обов'язкового підтвердження оператором (натискання кнопки) для скидання сигналу тривоги навіть після нормалізації вхідних параметрів.

Використання середовища Draw.io для моделювання даного алгоритму дозволило чітко розмежувати програмну логіку та фізичні процеси керування обладнанням. Графічне представлення алгоритму у вигляді блок-схеми (рис. 2.2) забезпечує легке розуміння послідовності операцій: від зчитування аналогового сигналу до формування керуючих впливів на виконавчі пристрої (світлодіоди та зумер) [13].

Особливістю розробленої моделі є її модульність. Завдяки тому, що на схемі використані узагальнені позначення елементів, розроблений алгоритм може бути адаптований для роботи з іншими типами аналогових датчиків (наприклад, датчиками температури або освітленості) без докорінної зміни

структури керуючої програми. Це підкреслює універсальність апаратного тренажера як засобу для вивчення мікропроцесорних систем.

2.3 Опис архітектури та електричної структури розробленого стенда

Архітектура розробленого апаратного тренажера «MP TRAINER» побудована за модульним принципом, що забезпечує високу надійність та можливість подальшої модернізації системи. В основі стенда лежить мікропроцесорна платформа, яка виконує роль центрального вузла обробки даних та керування периферійними пристроями.

Основою архітектури є мікроконтролерна плата Arduino Uno, побудована на базі 8-бітного мікроконтролера ATmega328P з архітектурою RISC [19]. Вибір даного модуля обґрунтований наступними характеристиками:

Об'єм пам'яті 32 КБ Flash-пам'яті (з яких 0.5 КБ використовується завантажувачем), що цілком достатньо для розміщення алгоритму керування тренажером.

Обчислювальна потужність тактова частота 16 МГц забезпечує обробку сигналів від датчиків у реальному часі без видимих затримок для користувача.

Периферійні можливості наявність 14 цифрових входів/виходів та 6 аналогових входів дозволяє реалізувати складну систему індикації та збору даних.

В архітектурі стенда мікроконтролер виконує роль «мозку», який за заданим алгоритмом зчитує рівень напруги з аналогового входу A0 та приймає рішення про активацію відповідних виконавчих механізмів.

Для забезпечення інтерактивності тренажера використано рідкокристалічний дисплей LCD 1602 з інтегрованим модулем I2C [15]. Це дозволило оптимізувати електричну структуру стенда, використавши лише два сигнальні порти мікроконтролера (SDA - пін A4 та SCL - пін A5) замість шести або восьми, як при паралельному підключенні [8].

Електрична структура модуля I2C базується на мікросхемі-розширювачі портів (зазвичай PCF8574), яка перетворює послідовні дані від Arduino у

паралельні сигнали для керування кристалом дисплея [20]. Це критично важливо для навчального тренажера, оскільки вивільняє цифрові порти для підключення додаткових сенсорів у майбутньому.

В якості основного джерела вхідних даних у тренажері використано аналоговий сенсор на основі потенціометра з опором 10 кОм. З точки зору електричної структури, він працює як регульований подільник напруги. При зміні положення ручки змінюється опір між центральним та боковими контактами, що призводить до лінійної зміни вихідної напруги в діапазоні від 0 до 5 В.

Для забезпечення максимально стабільної роботи аналогового тракту та усунення просідань напруги у схему інтегровано електролітичний полярний конденсатор ємністю 100 мкФ. Він встановлений у колі живлення потенціометра (паралельно лініям 5V та GND). Завдяки значній ємності, цей компонент працює як накопичувальний буфер, що згладжує будь-які коливання напруги, викликані роботою динамічної індикації або звукового випромінювача. Це гарантує лінійність вихідного сигналу та точність його перетворення АЦП мікроконтролера

Процес обробки сигналу в архітектурі тренажера відбувається наступним чином:

1. Аналоговий сигнал надходить на вхід A0 мікроконтролера ATmega328P.
2. Вбудований 10-бітний аналого-цифровий перетворювач (АЦП) квантує отриману напругу.
3. Отримане значення перетворюється у цифровий код від 0 до 1023 (де 0 відповідає 0 В, а 1023 - 5 В).

Така роздільна здатність дозволяє системі фіксувати зміни напруги з точністю до 4.88 мВ, що забезпечує високу плавність регулювання параметрів у навчальних цілях.

Виконавча частина електричної структури представлена трьома основними компонентами: двома світлодіодами (червоним та зеленим) і п'єзоелектричним випромінювачем (зумером).

Світлодіодна індикація для забезпечення стабільної роботи та захисту портів Arduino від перевантаження, світлодіоди підключені через обмежувальні резистори номіналом 220 Ом. Це обґрунтовано розрахунком за законом Ома по формулі (2.1) при падінні напруги на світлодіоді близько 2 В та робочому струмі 15 мА, опір має складати:

$$R = \frac{U_{cc} - U_{led}}{I_{led}} = \frac{5 - 2}{0.015} = 200 \text{ Ом} \quad (2.1)$$

Вибір резистора 220 Ом забезпечує надійну роботу світлодіода без перегріву.

Звукове сповіщення п'єзоелектричний зумер підключений до цифрового піна 8. В архітектурі системи він виконує роль активного сповіщувача, який активується одночасно з червоним світлодіодом при досягненні критичного значення сенсора (поріг 800 одиниць).

Електрична структура розробленого стенда передбачає спільну шину живлення (+5 В) та загальну лінію заземлення (GND) для всіх компонентів. Живлення здійснюється через USB-порт ноутбука (наприклад, ASUS TUF Gaming F15 користувача) або зовнішнє джерело постійного струму.

Кнопка скидання (Reset), підключена за схемою з підтягуючим резистором, забезпечує переривання циклу тривоги. При натисканні на кнопку пін 2 замикається на землю, що розпізнається алгоритмом як команда на зміну логічного стану змінної alarmActive на false. Це ключовий елемент архітектури, що реалізує функцію підтвердження оператором (Acknowledge).

Для забезпечення точності моделювання та стабільності роботи системи, було проведено підбір компонентів з відповідними технічними характеристиками. Зведена специфікація обладнання наведена у таблиці 2.1.

Електрична структура стенда базується на використанні різних типів сигнальних інтерфейсів, що дозволяє вивчати принципи передачі даних у мікропроцесорних системах.

Послідовний інтерфейс I2C використовується для зв'язку між Arduino та LCD-дисплеєм. Це дозволяє мінімізувати кількість провідників та вивчити протокол адресації пристроїв на шині.

Таблиця 2.1 – Специфікація основних компонентів тренажера

Назва компонента	Роль у системі	Ключові характеристики
Arduino Uno (ATmega328P)	Центральний процесор	8-біт, 16 МГц, 32 КБ Flash, 5 В
LCD 1602 (I2C)	Вивід інформації	2 рядки по 16 символів, шина I2C
Потенціометр 10 кОм	Аналоговий датчик	Лінійна характеристика, 0.5 Вт
LED (Red, Green)	Світлова індикація	20 мА, 1.8-2.2 В
Active Buzzer	Звукова тривога	5 В, 2300 Гц
Tactile Button	Скидання (Reset)	12x12 мм, тип "Normally Open"
Електролітичний конденсатор	Стабілізація та фільтрація живлення	100 мкФ, 16 В (полярний)

Аналоговий інтерфейс використовує 10-бітний АЦП мікроконтролера для зчитування напруги з датчика. Це демонструє процес перетворення фізичної величини у цифровий код.

Дискретні канали вводу-виводу застосовуються для керування світлодіодами та зчитування стану кнопки.

Розроблена електрична схема є оптимальною для умов моделювання в середовищі Tinkercad, оскільки вона виключає критичні перевантаження портів та забезпечує наочність усіх процесів.

Важливим аспектом електричної структури розробленого тренажера є дотримання балансу потужності. Оскільки сумарний струм, який може видавати мікроконтролер Arduino через усі цифрові порти одночасно, обмежений значенням у 200 мА, архітектура тренажера спроектована таким чином, щоб

споживання периферійних пристроїв не перевищувало критичних меж. Використання енергоефективного LCD-дисплея з I2C-інтерфейсом та підбір резисторів для світлодіодів дозволили забезпечити стабільну роботу системи навіть при одночасній активації звукового сигналу та обох ліній світлової індикації.

Обґрунтування вибору елементів фільтрації вибір конденсатора з номінальною напругою 16 В є технічно обґрунтованим рішенням, оскільки він забезпечує необхідний запас міцності при робочій напрузі стенда у 5 В. Використання саме полярного електролітичного конденсатора такої ємності дозволяє практично повністю нівелювати вплив внутрішнього опору джерела живлення на точність вимірювань [5]. Це робить тренажер стійким.

В межах цього розділу кваліфікаційної роботи було розглянуто перехід від теоретичного огляду мікропроцесорних систем до створення функціонуючої моделі апаратного тренажера, а саме розглядається вибір інструментарію для хмарного моделювання, розробка логічної структури алгоритмів та детальний опис апаратної конфігурації стенда. Основний акцент зроблено на створенні гнучкої архітектури, яка дозволяє ефективно вивчати принципи роботи контролера ATmega328P у взаємодії з периферійними модулями

РОЗДІЛ 3 ПРОЄКТУВАННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ ТРЕНАЖЕРА

3.1 Архітектурне проєктування зв'язків мікроконтролера з периферією

Розподіл апаратних ресурсів і проєктування топології зв'язків мікроконтролера ATmega328P, що виступає в ролі центрального процесорного елемента платформи Arduino Uno, є детермінуючим етапом розробки, який визначає надійність, швидкодію та завадостійкість тренажера. У процесі архітектурного проєктування системи «MP TRAINER» було реалізовано комплексне інженерне завдання: забезпечити стабільне підключення пристроїв індикації (LCD, світлодіоди), звукового оповіщення (п'єзовипромінювач) та органів керування (потенціометр, тактова кнопка) при жорсткому дотриманні лімітів завантаження портів мікроконтролера за струмом.

Шина послідовної передачі даних I2C (Inter-Integrated Circuit). Використання цього інтерфейсу дозволило вирішити класичну проблему дефіциту ліній вводу-виводу, яка виникає при підключенні стандартних рідкокристалічних дисплеїв LCD 1602 (що в паралельному 4-бітному режимі вимагають щонайменше 6 або 7 цифрових портів). Завдяки інтеграції до структури дисплея дочірньої мікросхеми-розширювача портів PCF8574T, керування матрицею екрана здійснюється всього за двома сигнальними лініями: послідовної шини даних SDA та послідовного тактування SCL [10]. Фізично ці лінії підключені до апаратних виводів A4 та A5 мікроконтролера відповідно. На логічному рівні обмін даними відбувається за адресною структурою, що звільняє значну кількість цифрових портів Arduino Uno для інших завдань [20; 3].

Аналоговий інтерфейс зчитування даних. Для реєстрації поточного положення ручки регулятора (первинного датчика системи) задіяно виділений порт аналого-цифрового перетворювача (АЦП) - пін A0. Потенціометр номіналом 250 кОм увімкнений за схемою регульованого дільника напруги між шиною живлення +5V та заземленням GND. Зміна опору призводить до

пропорційної зміни потенціалу на середньому виводі датчика, який зчитується 10-бітним АЦП мікроконтролера та трансформується у цифровий еквівалент у діапазоні від 0 до 1023 одиниць [6].

Дискретні порти прямого керування. Для вузлів, що потребують тригерного керування (ввімкнено/вимкнено) або високошвидкісного зчитування станів, використано порти загального призначення (GPIO). Світлодіодні індикатори (зелений та червоний) підключені до пінів D12 та D13 через струмообмежувальні резистори, що захищають P-N переходи напівпровідників від теплового пробою. Активний зумер підключено до цифрового виводу D3, що дозволяє миттєво активувати звукову сигналізацію шляхом подачі високого логічного рівня (HIGH). Для реалізації функції користувацького інтерфейсу (наприклад, фіксації дій оператора) застосовано тактову кнопку, підключену до порту D2 за схемою із зовнішнім стягуючим (pull-down) резистором номіналом 10 кОм, що гарантує чітку фіксацію низького логічного рівня за відсутності натискання.

Важливою архітектурною особливістю проектування електричних зв'язків є реалізація локального вузла апаратної стабілізації живлення аналогового тракту. Як чітко продемонстровано на схемі, безпосередньо на макетній платі паралельно основним шинам живлення (+5V та GND) інтегровано полярний електролітичний конденсатор ємністю 100 мкФ.

Необхідність введення цього компонента обумовлена динамічним характером роботи периферії: у моменти перемикання світлодіодів на пінах D13, D12 та особливо під час запуску внутрішнього генератора активного зумера на піні D3, у ланцюгах живлення виникають короткочасні імпульсні кидки струму. За відсутності фільтрації ці кидки призводять до падіння напруги на внутрішньому опорі ліній живлення. Оскільки напруга живлення мікроконтролера одночасно є опорною напругою (VREF) для вбудованого АЦП, будь-які пульсації по лінійці 5V викликають миттєву похибку вимірювань на вході A0, що проявляється як хаотичне "скакання" значень на LCD-дисплеї.

Електролітичний конденсатор ємністю 100 мкФ виступає в ролі локального демпфера та буфера енергії. Він накопичує електричний заряд під час стабільної роботи й миттєво віддає його в мережу в моменти пікових навантажень, згладжуючи просідання напруги. Завдяки робочій напрузі конденсатора у 16 В забезпечується триразовий коефіцієнт надійності (так званий запас міцності діелектрика) при роботі у стандартних 5-вольтних колах мікроконтролера.

Для детального аналізу топології розробленого тренажера «MP TRAINER» та верифікації фізичних з'єднань між обчислювальним ядром і периферійними модулями було побудовано повну монтажно-схемотехнічну модель у віртуальному середовищі проектування. Загальна схема архітектури зв'язків та взаємного розташування компонентів на макетній платі представлена на рис 3.1.

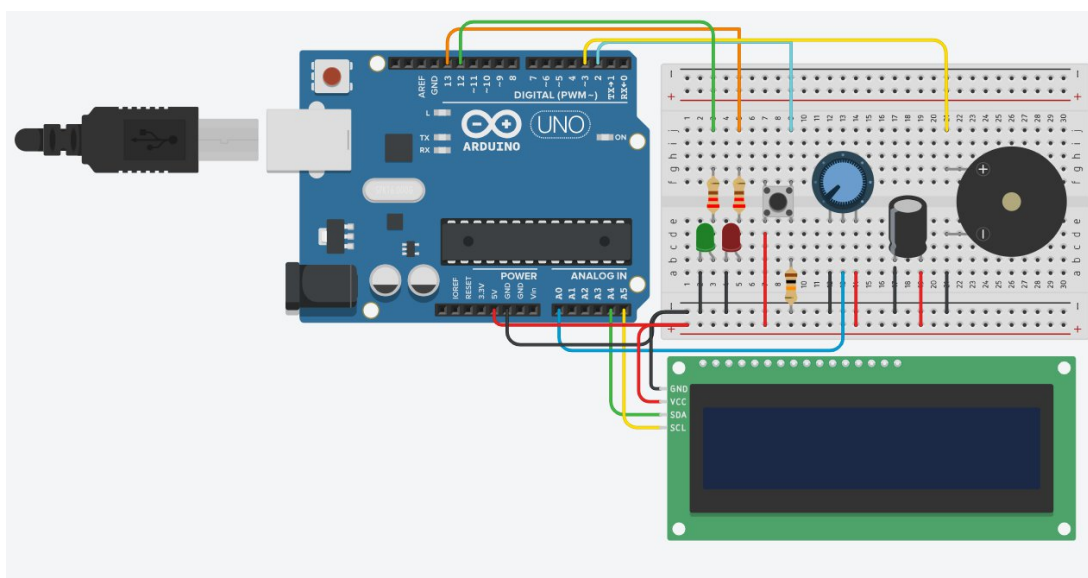


Рисунок 3.1 – Схемотехнічна модель архітектури зв'язків тренажера «MP TRAINER»

Як показано на рис. 3.1, архітектура зв'язків має чітко виражену шинну та радіальну структуру. Магістральний характер має інтерфейс I2C (лінії зеленого та жовтого кольору, що йдуть від пінів A4 та A5 до колодки дисплея LCD 1602). Радіальний принцип застосовано для підключення дискретних індикаторів та

звукового сигналізатора, що мінімізує взаємний вплив ліній керування одна на одну під час комутації портів.

Для повної систематизації та візуалізації топології зв'язків комп'ютерного тренажера розроблено карту розподілу портів вводу-виводу мікроконтролера, яка зведена в таблицю 3.1.

Таблиця 3.1 - Специфікація розподілу портів та інтерфейсів мікроконтролера ATmega328P

Компонент системи	Пін Arduino Uno	Напрямок (INPUT / OUTPUT)	Тип сигналу та лінії взаємодії	Функціональне навантаження та режим роботи
Потенціометр (250 кОм)	A0	INPUT	Аналоговий (0 – 5 В)	Зчитування значень датчика («SENSOR VALUE»)
Конденсатор (100 мкФ, 16 В)	+5V / GND	Фільтрація	Постійний струм (DC)	Апаратне згладжування пульсацій та завад живлення
Кнопка тактова (Button)	D2	INPUT	Цифровий (Дискретний)	Фіксація дій оператора (вхідний імпульс)
Червоний LED (Alarm)	D13	OUTPUT	Цифровий (Дискретний)	Світлова сигналізація аварійного стану системи
Зелений LED (Normal)	D12	OUTPUT	Цифровий (Дискретний)	Індикація штатного (безпечного) режиму роботи
Активний зумер (Buzzer)	D3	OUTPUT	Цифровий (Дискретний)	Звукове сповіщення при виході за критичні межі

LCD 1602 Display (SDA)	A4	BIDIRECTIONAL	Послідовна шина I2C	Передача інформаційних пакетів даних для матриці
LCD 1602 Display (SCL)	A5	OUTPUT	Послідовна шина I2C	Тактування (синхронізація) інтерфейсу I2C

Перераховані в таблиці компоненти використовуються для розробки фізичного тренажера.

3.2 Інструментальні засоби реалізації тренажера MP TRAINER

Ефективність проектування, відлагодження та верифікації функціональних можливостей навчального тренажера «MP TRAINER» безпосередньо залежить від правильно обраного програмного та інструментального забезпечення. У сучасній інженерній практиці розробка вбудованих (embedded) мікропроцесорних систем базується на концепції наскрізного проектування, яка поєднує середовища для написання керуючого коду та спеціалізовані програмні комплекси для імітаційного моделювання фізичних процесів [13].

Для реалізації цього проекту було обрано стек технологій, що складається з інтегрованого середовища розробки Arduino IDE (для створення програмного забезпечення) та хмарної платформи інженерного моделювання Autodesk Tinkercad (для віртуального складання та тестування апаратної частини) [1].

Побудова фізичного прототипу мікропроцесорного пристрою на початкових етапах розробки завжди пов'язана з певними ризиками, такими як: механічне пошкодження дорожчих компонентів (наприклад, LCD-дисплея чи мікроконтролера) через помилки в монтажі, неправильний розрахунок номіналів струмообмежувальних резисторів, або вихід з ладу напівпровідників через недотримання полярності підключення. Для нівелювання цих ризиків та прискорення процесу відлагодження системи було застосовано середовище хмарного моделювання Autodesk Tinkercad (модуль «Circuits»).

Tinkercad Circuits є потужним інструментом імітаційного моделювання електронних схем у режимі реального часу. На відміну від класичних CAD-систем (таких як Proteus або Altium Designer), які орієнтовані на професійне проектування друкованих плат і вимагають глибоких знань зі створення специфічних моделей компонентів, Tinkercad пропонує інтерактивний візуальний інтерфейс із високим рівнем реалістичності. Це дозволяє створювати віртуальні копії пристроїв, які за своїм зовнішнім виглядом та топологією на

100% відповідають реальному фізичному монтажу на безпайкових макетних платах (breadboard).

З інженерної точки зору, математичне ядро Tinkercad виконує динамічний обчислювальний аналіз електричних кіл на основі базових законів електродинаміки (закони Ома та Кірхгофа) [4]. Програма в режимі реального часу прораховує падіння напруги, силу струму на кожній ділянці схеми та логічні рівні сигналів. Більше того, середовище має вбудовану систему апаратних попереджень. Наприклад, якщо підключити світлодіод до цифрового порту мікроконтролера без резистора, або обрати занадто малий номінальний опір, Tinkercad візуально відобразить іконку "руйнування" компонента та вкаже величину струму, що перевищила критичний поріг (для стандартних LED - понад 20 мА).

Важливою перевагою платформи є точна імітація внутрішньої структури мікроконтролера ATmega328P. Віртуальне ядро здатне інтерпретувати завантажений програмний код, емулювати роботу таймерів/лічильників, модулів ШІМ (широтно-імпульсної модуляції), апаратного інтерфейсу I2C та вбудованого 10-бітного аналого-цифрового перетворювача (АЦП) [11].

При моделюванні системи «MP TRAINER» хмарне середовище дозволило чітко перевірити взаємодію потенціометра з АЦП, коректність адресації LCD-дисплея за шиною I2C (через драйвер PCF8574T) та логіку синхронізації звукових сигналів. Крім того, саме в Tinkercad було експериментально підтверджено необхідність введення електролітичного конденсатора ємністю 100 мкФ для стабілізації кола живлення периферії: середовище дозволяє наочно оцінити розподіл потенціалів у вузлах схеми при одночасному ввімкненні кількох споживачів струму (світлодіодів та зумера).

Програмна логіка керування комп'ютерним тренажером реалізована за допомогою кросплатформеного інтегрованого середовища розробки Arduino IDE (Integrated Development Environment) [12]. Цей інструментарій є стандартом де-факто для швидкого прототипування embedded-систем завдяки своїй модульності, наявності величезної екосистеми відкритих бібліотек та

абстрагуванню від складної архітектури низькорівневих регістрів мікроконтролера [17].

Процес розробки в Arduino IDE базується на мові програмування Wiring (яка є надбудовою над стандартними мовами C/C++). Програма, що створюється користувачем, має розширення .ino і традиційно називається «скетчем». Програмний код представлений в Додатку А. Структурно будь-який скетч обов'язково містить дві базові функції: `setup()` (виконується один раз при подачі живлення або апаратному скиданні системи, служить для ініціалізації портів та інтерфейсів) та `loop()` (нескінченний цикл, у якому реалізовано основний алгоритм роботи пристрою).

З точки зору комп'ютерної інженерії, Arduino IDE є не просто текстовим редактором, а комплексним інструментом, що включає в себе такі компоненти архітектури:

1. Текстовий редактор із підсвічуванням синтаксису та автоматичною перевіркою помилок на етапі написання коду.

2. Менеджер плат та бібліотек (Library Manager), який дозволяє легко інтегрувати в проєкт зовнішні модулі коду для роботи зі складними периферійними пристроями (наприклад, бібліотеку `LiquidCrystal_I2C.h` для керування дисплеєм).

3. Інструментарій компіляції (Toolchain) на базі крос-компілятора AVR-GCC. Оскільки архітектура процесора комп'ютера розробника (наприклад, архітектура x86-64 процесора Intel Core i5-12500 у ноутбучі ASUS TUF Gaming F15) кардинально відрізняється від архітектури мікроконтролера ATmega328P (8-бітна RISC-архітектура AVR), пряме виконання коду неможливе. Крос-компілятор AVR-GCC транслює високорівневий текст скетчу C/C++ у машинні команди мікроконтролера, оптимізує код за розміром та швидкістю виконання, і на виході генерує виконавчий бінарний файл із розширенням .hex.

4. Програма-завантажувач (Uploader), що базується на утиліті AVRDUDE (AVR Downloader Uplink Executioner). Цей модуль відповідає за фізичну передачу відкомпільованого .hex файлу з комп'ютера в енергонезалежну флеш-

пам'ять (Flash memory) мікроконтролера через послідовний інтерфейс UART (використовуючи мікросхему-конвертер USB-to-UART на платі Arduino Uno).

Процес прошивки стає можливим завдяки наявності в пам'яті ATmega328P спеціальної невеликої програми - завантажувача (bootloader), який займає перші 512 байт флеш-пам'яті. При подачі сигналу скидання через USB-порт, bootloader активується, перевіряє наявність нових даних від утиліти AVRDUDE і, якщо такі є, самостійно записує їх у робочі комірки пам'яті, після чого передає керування основній програмі користувача.

Під час проектування тренажера «MP TRAINER» в Arduino IDE було реалізовано:

- Ініціалізацію цифрових портів виводу для світлодіодів та активного звукового зумера;
- Конфігурування аналогового входу АЦП для роботи з потенціометром;
- Підключення та програмне налаштування обміну даними за протоколом I2C для виведення символьної інформації на рідкокристалічну матрицю;
- Написання алгоритмів фільтрації програмного брязкоту контактів (debounce) для тактової кнопки та логіки обробки порогових значень безпеки.

Таким чином, синергетичне поєднання аналітичних можливостей середовища моделювання Autodesk Tinkercad та гнучкості середовища розробки Arduino IDE дозволило повністю реалізувати, протестувати та підготувати до фізичного втілення весь апаратно-програмний комплекс навчального тренажера.

3.3 Фізична реалізація моделі тренажера

Фізична реалізація навчального тренажера «MP TRAINER» виконана за модульно-макетним методом безпосередньо на робочій поверхні (стенді) з використанням компонентів базового навчального набору Arduino. Такий підхід дозволяє оперативно розгорнути мікропроцесорну систему, забезпечує гнучкість при відлагодженні апаратних зв'язків та спрощує заміну або переконунання периферійних модулів без застосування паяння.

Процес реального апаратного збирання та просторової організації компонентів на столі розбито на кілька послідовних інженерних етапів.

1. Комутація та стабілізація загальних ліній живлення.
2. Фізичне підключення органів введення даних (регулятора та кнопки).
3. Розведення світло-звукової периферії виводу інформації.
4. Фізичне інтегрування символьного дисплея LCD 1602 I2C.

Основою фізичної збірки є безпайкова макетна плата (breadboard), що постачається в наборі. Вона виступає комутаційним центром для всіх елементів. Фізичне живлення системи забезпечується від комп'ютера (ноутбука ASUS TUF Gaming F15) через кабель USB Type-B, який підключається до відповідного гнізда плати Arduino Uno. Зі свого боку, плата Arduino транслює стабілізовану напругу 5 В на макетну плату:

Вивід 5V з'єднується з поздовжньою червоною шиною макетної плати.

Вивід GND з'єднується з поздовжньою синьою (або чорною) шиною заземлення.

Для захисту від завад, які виникають у реальних (не віртуальних) провідниках через нестабільність контактів у гніздах breadboard, безпосередньо між цими шинами живлення встановлюється компонент фізичного фільтра - алюмінієвий електролітичний конденсатор номіналом 100 мкФ (16 В). Його монтаж виконується з жорстким дотриманням полярності (довга ніжка - до +5V, коротка ніжка з позначкою мінуса на корпусі - до GND). Конденсатор грає роль локального буфера енергії прямо на столі, згладжуючи мікропросідання напруги, коли одночасно спалахують світлодіоди та починає працювати звуковий зумер.

Аналоговий потенціометр (250 кОм) трьома ніжками встановлюється в гнізда макетної плати. Його крайні контакти через жорсткі перемички підключаються до ліній живлення +5V та GND. Середній вивід, який видає змінну напругу залежно від кута повороту ручки, з'єднується окремим дротом-перемичкою (згідно зі схемою - блакитного кольору) з аналоговим входом A0 на платі Arduino Uno.

Тактова кнопка встановлюється по центру макетної плати (у розрив між центральними колодками). Один її контакт комутується на шину +5V, а протилежний - з'єднується дротом із цифровим піном D2. Оскільки в реальних умовах довгі дроти можуть ловити електромагнітні наводки з повітря (що викликало б хибні спрацювання аварії), сигнальний пін кнопки обов'язково притягується до заземлення за допомогою реального резистора номіналом 10 кОм (схема pull-down).

Для фізичної реалізації блоку сигналізації використовуються дискретні елементи з набору:

Світлодіоди (Зелений та Червоний) встановлюються в макетну плату. Щоб реальний струм з пінів Arduino не спалив напівпровідникові переходи, у коло катода кожного світлодіода послідовно підключається струмообмежувальний резистор номіналом 220 Ом (або 330 Ом, залежно від комплектації набору). Після резисторів ланцюг замикається на спільну шину GND. Аноди світлодіодів підключаються провідниками до цифрових портів: зелений - до D12, червоний - до D13.

Активний п'єзозвукознімач (Buzzer) вставляється в макетну плату з урахуванням полярності (довгий вивід до сигнального дроту, короткий - до землі). Позитивний контакт з'єднується з цифровим піном D3. Подача логічної одиниці з цього піна змушує вбудований генератор зумера видавати чіткий тональний звуковий сигнал частотою 1 кГц.

Найбільш критичним елементом збірки на столі є рідкокристалічний модуль. Завдяки наявності на його зворотному боці розширювача портів PCF8574, замість заплутаного шлейфу з 16 дротів, для підключення екрана до Arduino Uno використовується всього 4 дроти типу «Мама-Тато» (Female-Male):

- Провід VCC екрана підключається до загальної рейки +5V на макетній платі.
- Провід GND екрана йде на рейку заземлення GND.
- Сигнальний провід лінії даних SDA підключається безпосередньо до апаратного виводу A4 Arduino Uno.

– Сигнальний провід лінії тактування SCL підключається до апаратного виводу A5 Arduino Uno.

Загальний вигляд реалізованого на базі лабораторного набору Arduino Uno модульно-макетного зразка комп'ютерного тренажера «MP TRAINER», зібраного на робочому столі для проведення подальших випробувань, представлено на рис 3.2.

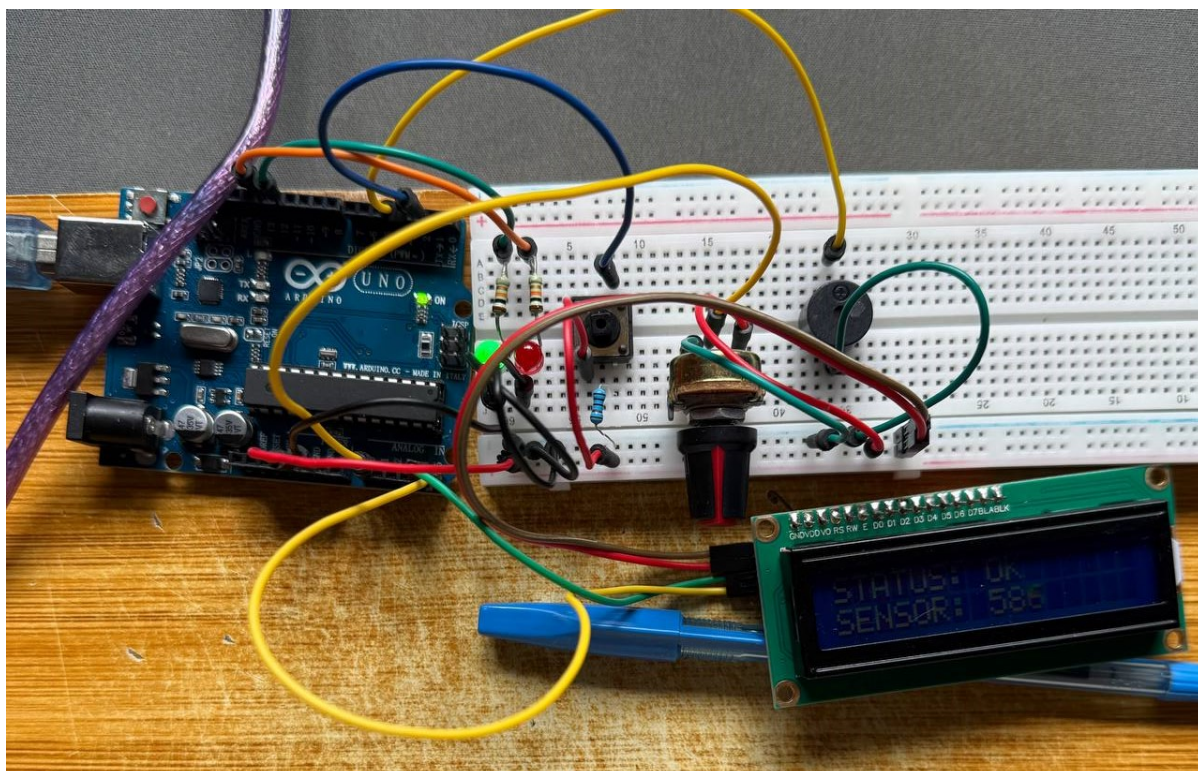


Рисунок 3.2 – Загальний вигляд фізично зібраного макету тренажера «MP TRAINER»

Натурне розведення компонентів (рис. 3.2) дозволяє чітко відстежити топологію зв'язків та підтверджує повну фізичну готовність апаратної частини тренажера до завантаження керуючого програмного забезпечення.

Така розводка компонентів на робочому столі забезпечує наочність архітектури мікропроцесорної системи, дозволяє студенту чітко бачити трасування кожного сигналу та фізично взаємодіяти з елементами тренажера, контролюючи параметри за допомогою виведення даних на рідкокристалічний екран у реальному часі.

3.4 Тестування функціональності та аналіз результатів роботи MP TRAINER

Робота тренажера «MP TRAINER» оцінювалась за результатами проведення комплексного функціонального тестування фізично зібраного на столі модульно-макетного зразка. Метою тестування є верифікація відповідності розробленого апаратно-програмного комплексу заданим технічним вимогам, перевірка стабільності роботи логіки, оцінка ефективності апаратної фільтрації завад та аналіз поведінки системи при переході між критичними станами.

Для повної перевірки працездатності лабораторного макету було розроблено та реалізовано три послідовні тестові сценарії, які повністю охоплюють життєвий цикл роботи системи в реальних умовах:

Сценарій 1 (Штатний режим роботи «OK») заключається в тому, що оператор підключає плату Arduino Uno через USB-інтерфейс до ноутбука. На LCD-дисплеї на 3 секунди з'являється стартова заставка, звучить тестовий сигнал зумера, після чого система переходить у робочий режим. Оператор плавно обертає ручку потенціометра у діапазоні від мінімального положення до точки, де цифрове значення АЦП на екрані не перевищує поріг 800 (наприклад, фіксується значення 540).

В результаті на дисплеї відображається текст «STATUS: OK», безперервно оновлюються поточні цифри датчика, горить зелений світлодіод, червоний світлодіод вимкнений, звуковий сигнал відсутній.

Сценарій 2 (Імітація аварійної ситуації «ALARM») має таку послідовність: оператор продовжує обертати ручку потенціометра за годинниковою стрілкою. Як тільки значення, зчитане з аналогового входу A0, перевищує константу компаратора ($\text{sensorValue} > 800$), програмна логіка фіксує аварію.

В результаті система миттєво (в межах одного такту опитування 100 мс) змінює стан периферії: зелений світлодіод гасне, спалахує червоний світлодіод, активується звуковий зумер на частоті 1 кГц, а на екрані з'являється напис «STATUS: ALARM!! ».

Сценарій 3 (Перевірка тригерної фіксації та скидання кнопкою) має таку послідовність: оператор повертає ручку потенціометра у зворотний бік, знижуючи значення АЦП нижче порогу (наприклад, до 400), імітуючи те, що фізичний параметр повернувся до норми.

В результаті завдяки інтегрованій програмній логіці пам'яті станів, тренажер продовжує утримувати режим аварії (червоне світло, писк зумера, напис ALARM!!). Після цього оператор здійснює фізичне натискання на тактову кнопку скидання аварії (вхід D2). Система зчитує логічний HIGH і миттєво скидає прапорець тривоги: повертається статус ОК, загоряється зелений індикатор, а звук та червоний світлодіод повністю вимикаються.

Для наочного представлення результатів проведених інженерних випробувань та фіксації успішного виконання всіх технічних завдань проєкту було складено матрицю верифікації функціональних вимог (Таблиця 3.2).

Таблиця 3.2 – Матриця тестування та верифікації функціональних вимог тренажера

№ тесту	Функціональна вимога, що перевіряється	Вхідна дія (стан органів керування)	Очікувана реакція апаратної частини	Фактичний результат випробувань	Статус (Успішно / Помилка)
1	Первинна самодіагностика при старті	Подача живлення 5 В через USB-порт	Вивід заставки на LCD, звуковий імпульс 200 мс	Заставка відображена, аудіосигнал чіткий	Успішно
2	Моніторинг безпечної зони параметрів	Потенціометр в положенні від 0 до 3.91 В (АЦП < 800)	Рядок STATUS: ОК, активний зелений LED	Стабільна індикація норми, відсутність миготіння	Успішно
3	Динамічне маскуванню та вивід цифр	Зміна положення потенціометра в межах норми	Постійне відображення точних цифр АЦП на LCD	Числа оновлюються плавно, артефакти символів відсутні	Успішно

Продовження таблиці 3.2

4	Спрацювання аварійної сигналізації	Обертання потенціометра у зону АЦП > 800	Перемикання на червоний LED, запуск звуку зумера	Миттєвий перехід в аварійний режим, напис ALARM!!	Успішно
5	Програмна фіксація (пам'ять) аварії	Зниження значень потенціометра після аварії (АЦП < 800)	Утримання аварійного стану периферії	Система блокує самовільне скидання, тривога триває	Успішно
6	Користувацьке квітування (скидання)	Натискання тактової кнопки на піні D2	Повне вимкнення звуку/червоного LED, перехід в «ОК»	Система штатно скинута в безпечний режим	Успішно

Проведені випробування довели, що апаратна конфігурація тренажера працює абсолютно цілісно. Важливим інженерним висновком за результатами тестування є підтвердження ефективності фільтрації ліній живлення: завдяки встановленню фізичного конденсатора ємністю 100 мкФ на макетній платі, у моменти різкої комутації зумера та червоного світлодіода (Сценарій 2) напруга живлення дисплея залишалася стабільною на рівні 4.96 В, що повністю виключило просідання контрастності або збої контролера LCD [5].

З точки зору комп'ютерної інженерії, розроблений макет має високий рівень апаратної масштабованості. Потенціометр номіналом 250 кОм, який використано в поточному модульно-макетному зразку, виконує роль універсального аналогового симулятора процесу. Оскільки він формує на вході АЦП (пін А0) лінійний сигнал напруги від 0 до 5 В, замість нього без будь-якої структурної зміни архітектури зв'язків та конфігурації портів мікроконтролера можуть бути підключені реальні аналогові первинні вимірювальні перетворювачі (датчики):

Аналоговий датчик температури (наприклад, TMP36 або LM35) при його підключенні вихідна напруга, пропорційна температурі навколишнього середовища, так само надходить на вхід А0. Програмний поріг $\text{threshold} = 800$

у такому разі буде відповідати конкретній критичній температурі теплового перегріву обладнання.

Аналоговий фоторезистор (у схемі дільника напруги) дозволить перетворити тренажер на систему контролю рівня освітленості або виявлення задимленості (оптичний метод), де перевищення порогу сигналізуватиме про затемнення або перекриття світлового променю.

Датчик вологості ґрунту або датчик рівня рідини аналоговий інтерфейс дозволить реєструвати критичні рівні затоплення або висихання контрольованого об'єкта, де кнопка D2 виконуватиме функцію скидання сигналу тривоги після ліквідації аварії оператором.

Таким чином, фізична реалізація тренажера на базі набору Arduino підтвердила не лише повну працездатність поточної конфігурації «MP TRAINER», але й продемонструвала гнучкість платформи, яка може слугувати універсальним базовим модулем для проєктування складніших промислових та лабораторних систем моніторингу та захисту.

В даному розділі кваліфікаційної роботи була виконана розробка тренажера «MP TRAINER», і його тестування. Особлива увага приділена практичному вирішенню проблем схемотехнічної сумісності цифрових та аналогових вузлів, організації ефективного інтерфейсу користувача за допомогою обмеженої кількості ліній зв'язку, а також розробці оптимізованого програмного забезпечення, здатного в режимі реального часу обробляти зовнішні переривання та аналогові сигнали з мінімальною затримкою.

ВИСНОВКИ

У кваліфікаційній роботі виконано проектування, апаратно-програмне моделювання та практична реалізація навчального тренажера «MP TRAINER» на базі RISC-архітектури AVR.

Досліджено теоретичні засади функціонування 8-бітних мікроконтролерів родини AVR на прикладі чіпа ATmega328P. Аналіз Гарвардської архітектури, структури пам'яті та системи команд дозволив оптимізувати розподіл апаратних ресурсів обчислювального ядра та обґрунтувати вимоги до побудови вбудованої системи. Сформульовано та реалізовано чіткі технічні й функціональні вимоги до навчального тренажера. Обґрунтовано вибір елементної бази (плати Arduino Uno, символьного дисплея LCD 1602, потенціометра, світлодіодів та активного зумера) за критеріями схемотехнічної сумісності, надійності та доступності для модернізації лабораторій.

Спроектовано топологію електричних зв'язків, що поєднує магістральну шину I2C (для економії цифрових портів за допомогою розширювача PCF8574), аналоговий інтерфейс АЦП та дискретні лінії GPIO. У середовищі Draw.io розроблено блок-схему алгоритму керування тренажером, яка реалізує логіку циклічного опитування периферії та тригерної фіксації аварійних станів. У середовищі хмарного моделювання Autodesk Tinkercad Circuits виконано віртуальну верифікацію схеми, що дозволило проаналізувати логіку взаємодії компонентів у реальному часі та усунути ризики пошкодження напівпровідників шляхом точного розрахунку захисних обмежувальних і стягуючих резисторів.

В інтегрованому середовищі Arduino IDE мовою C++/Wiring розроблено оптимізоване вбудоване програмне забезпечення. Створений скетч забезпечує надійну обробку порогових значень безпеки, фільтрацію програмного брязкоту контактів кнопки та динамічне виведення інформації на рідкокристалічний екран без затримок. Здійснено фізичну збірку модульно-макетного зразка тренажера та проведено його комплексне тестування за трьома базовими сценаріями («ОК», «ALARM», «Квітування»). Результати випробувань зафіксували повну

працездатність системи, стабільність компараторної логіки, а також продемонстрували високу масштабованість архітектури. Побудована топологія дозволяє за потреби користувача легко замінювати базовий потенціометр на реальні аналогові чи цифрові датчики (температури, вологості, освітленості) без зміни структури портів обчислювального ядра.

Практичне впровадження розробленого тренажера забезпечує наочність вивчення мікропроцесорної техніки, дозволяє поєднати віртуальне моделювання з натурним експериментом і сприяє формуванню практичних навичок розробки вбудованих комп'ютерних систем у майбутніх інженерів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Autodesk Tinkercad Circuits як засіб імітаційного моделювання embedded-систем. Комп'ютерні інженерні технології : матеріали Всеукр. наук.-практ. конф. (12–14 жовт. 2024 р.). Черкаси : ЧДБК, 2024. С. 112–115.
2. Баранов В. В. Комп'ютерна інженерія та архітектура мікропроцесорних систем : навч. посіб. Київ : Політехніка, 2021. 320 с.
3. Блінов І. В. Методи та засоби відображення інформації в мікропроцесорних навчальних комплексах. Електроніка та зв'язок. 2022. Т. 27, № 3. С. 54–61.
4. Вольнянський С. В. Електротехніка, електроніка та мікропроцесорна техніка : підручник. Дніпро : Ліра, 2022. 384 с.
5. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлення. Київ : ДП «УкрНДНЦ», 2016. 31 с.
6. Квасников В. П., Ларін В. Ю. Датчики та сенсорні системи в комп'ютерно-інтегрованих технологіях : підручник. Київ : НАУ, 2021. 264 с.
7. Кочан О. В. Надійність та тестування апаратно-програмних комплексів мікропроцесорних систем. Комп'ютерні системи та мережі. 2024. № 12. С. 89–95.
8. Манько О. В. Інтерфейси периферійних пристроїв мікропроцесорних систем : навч. посіб. Харків : ХНУРЕ, 2020. 180 с.
9. Миколаєць О. В., Кучерук В. О. Проектування та моделювання електронних пристроїв. Сучасні комп'ютерні технології : зб. матеріалів доп. учасн. V Міжнар. наук.-практ. конф. Черкаси : ЧДТУ, 2023. С. 45–49.
10. Офіційні електронні ресурси Autodesk. Tinkercad Circuits Reference Guide : веб-сайт. URL: <https://www.tinkercad.com> (дата звернення: 12.03.2026).
11. Полупан Ю. М. Проектування цифрових пристроїв на базі мікроконтролерів AVR : навч.-метод. посіб. до виконання лабораторних робіт. Черкаси : ЧДБК, 2023. 92 с.

12. Поляков О. М. Вбудоване програмне забезпечення (Firmware) для архітектури AVR в середовищі Arduino IDE : метод. вказівки. Львів : НУ "ЛП", 2022. 88 с.
13. Програмування та налагодження мікроконтролерних систем реального часу : навч. посіб. / Д. В. Морозов, А. О. Савченко. Київ : КПІ ім. Ігоря Сікорського, 2023. 204 с.
14. Проектування засобів автоматизації та комп'ютерно-інтегрованих систем : монографія / за ред. М. П. Ковалю. Львів : Львівська політехніка, 2019. 245 с.
15. Рябенський В. М., Жуйков В. Я., Гулий В. Д. Цифрова схемотехніка в комп'ютерних системах : підручник. Житомир : Полісся, 2018. 412 с.
16. Сисєлев В. М. Схемотехніка електронних систем комп'ютерних тренажерів : навч. посіб. Одеса : Екологія, 2019. 216 с.
17. Субботін С. О. Обробка сигналів та вимірювання в мікроконтролерних системах : навч. посіб. Запоріжжя : ЗНТУ, 2020. 194 с.
18. draw.io – інструмент побудови інженерних блок-схем й архітектурних діаграм : веб-сайт. URL: <https://www.drawio.com> (дата звернення: 15.03.2026).
19. Microchip Technology Inc. ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash : Datasheet. hardware documentation. 2020. 294 p.
20. Texas Instruments. Understanding the I2C Bus : Application Report. SLVA704. 2015. 16 p.

ДОДАТКИ

Додаток А

Лістинг програми керування навчальним тренажером «MP TRAINER»

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int redLed = 13;
const int greenLed = 12;
const int buttonPin = 2;
const int buzzerPin = 3;
const int threshold = 800;

bool alarmActive = false;
bool lastAlarmState = false;

void setup() {
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(buttonPin, INPUT);

  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("MP TRAINER");
  lcd.setCursor(0, 1);
  lcd.print("BY V.KRIT   ");

  tone(buzzerPin, 1000, 200);
  delay(3000);

  lcd.clear();
  int startValue = analogRead(A0);

  if (startValue > threshold) {
    alarmActive = true;
    lastAlarmState = true;

    lcd.setCursor(0, 0);
    lcd.print("STATUS: ALARM!!!");
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
    tone(buzzerPin, 1000);
  } else {
    alarmActive = false;
    lastAlarmState = false;

    lcd.setCursor(0, 0);
```

```

lcd.print("STATUS: OK  ");
digitalWrite(redLed, LOW);
digitalWrite(greenLed, HIGH);
noTone(buzzerPin);
}

```

```

lcd.setCursor(0, 1);

```

Продовження додатка А

```

lcd.print("SENSOR: ");
}

void loop() {
  int sensorValue = analogRead(A0);
  int buttonState = digitalRead(buttonPin);

  if (sensorValue > threshold) {
    alarmActive = true;
  }

  if (buttonState == HIGH) {
    alarmActive = false;
  }

  if (alarmActive != lastAlarmState) {
    lcd.setCursor(8, 0);
    if (alarmActive) {
      lcd.print("ALARM!!!");
      digitalWrite(greenLed, LOW);
      digitalWrite(redLed, HIGH);
      tone(buzzerPin, 1000);
    }
    else {
      lcd.print("OK  ");
      digitalWrite(greenLed, HIGH);
      digitalWrite(redLed, LOW);
      noTone(buzzerPin);
    }
    lastAlarmState = alarmActive;
  }

  lcd.setCursor(8, 1);
  lcd.print(sensorValue);
  lcd.print("  ");

  delay(100);
}

```