

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмований генератор випробувальних сигналів на базі мікроконтролера
ATMega328 для перевірки логічних мікросхем

Назва теми

КВРКІ 22024.40.02.40 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Виконав: студент III курсу, група KI2c-22-2


Підпис


Володимир МАРКОВ
Ініціали, прізвище

Керівник


Підпис, дата

Василь СТЕЦЮК
Ініціали, прізвище

Нормоконтролер


Підпис, дата

Тетяна КИСІЛЬ
Ініціали, прізвище

До захисту допускаю:
зав. кафедри комп'ютерної
інженерії та інформаційних
систем


Підпис

Ольга ПАВЛОВА
Ініціали, прізвище

«9» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Володимиру МАРКОВУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмований генератор випробувальних сигналів на базі мікроконтролера ATmega328 для перевірки логічних мікросхем

Керівник проекту (роботи) Василь СТЕЦІЮК, старший викладач

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2025 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Огляд існуючих рішень проектування та тестування логічних мікросхем

Проектування програмованого генератора випробувальних сигналів на базі мікроконтролера ATmega328 для перевірки логічних мікросхем

Програмно-апаратна реалізація програмованого генератора випробувальних сигналів на базі мікроконтролера ATmega328 для перевірки логічних мікросхем

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Архітектура контролера

Схема підключення пристрою

Результати функціонування пристрою

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КІС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КІС		

7. Дата видачі завдання « 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2025	виконано
4	Робота над розділом 2 – вибір компонентів для проектування програмованого генератору випробувальних сигналів на базі мікроконтролера ATmega328	01.04.2025	виконано
5	Робота над розділом 3 – проектування програмованого генератору випробувальних сигналів на базі мікроконтролера ATmega328 для перевірки логічних мікросхем	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Керівник роботи

Підпис

Підпис

Володимир МАРКОВ

Ініціали, прізвище

Василь СТЕЦЮК

Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмований генератор випробувальних сигналів на базі мікроконтролера АТМega328 для перевірки логічних мікросхем».

Автор роботи: Володимир МАРКОВ.

Керівник роботи: Василь СТЕЦЮК.

Пояснювальна записка: 60 с., 18 рис., 11 табл., 3 дод., 60 джерел.

Графічна частина: 3 креслення.

МІКРОКОНТРОЛЕР, АРДУІНО, ГЕНЕРАТОР СИГНАЛІВ, ЛОГІЧНІ
ВЕНТИЛІ, ШАБЛОН СИГНАЛІВ.

Метою даної роботи є перевірка логічних мікросхем різних типів за допомогою програмованого генератора випробувальних сигналів на базі мікроконтролера АТmega328.

Об'єктом дослідження є цифрові логічні мікросхеми та їх типові комбінації, що застосовуються в електронних пристроях керування, обробки сигналів та автоматики.

Предметом дослідження є програмно-апаратні методи автоматизованого формування шаблонів логічних сигналів та контролю правильності роботи логічних елементів з використанням мікроконтролера Arduino.

Практичне значення полягає в тому, що програмований генератор випробувальних сигналів на базі мікроконтролера АТmega328 є універсальним інструментом для перевірки справності логічних мікросхем, що дозволяє подавати на їх входи типові та користувацькі сигнали з різною частотою, послідовністю та логікою, що значно спрощує процес діагностики.



Підпис студента

30.05.2025

Дата

ЗМІСТ

ВСТУП	3
1 ОГЛЯД ПРЕДМЕТНОЇ ГАЛУЗІ	5
1.1 Аналіз існуючих рішень	5
1.2 Логічні вентиля.....	11
1.3 Висновки до першого розділу.....	20
2 ПРОЄКТУВАННЯ ПРОГРАМОВАНОГО ГЕНЕРАТОРА ВИПРОБУВАЛЬНИХ СИГНАЛІВ НА БАЗІ МІКРОКОНТРОЛЕРА АТМЕГА328	21
2.1 Методи перевірки мікросхем	21
2.2 Основні характеристики мікроконтролера АТМega328	31
2.3 Моделювання схеми генератора логічних шаблонів.....	34
2.4 Висновки до другого розділу	41
3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ГЕНЕРАТОРА ВИПРОБУВАЛЬНИХ СИГНАЛІВ	43
3.1 Опис реалізації модулів апаратного та програмного забезпечення програмно-технічного засобу.....	43
3.1.1 Апаратне забезпечення	44
3.1.2 Програмне забезпечення	48
3.2 Опис процесу створення програмованого генератора випробувальних сигналів.....	50
3.3 Висновки до третього розділу.....	62
ВИСНОВКИ	63
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	66
ДОДАТОК А	73
ДОДАТОК Б	74
ДОДАТОК В	75

				КвРКІ. 22024.40.02.40 ПЗ				
Зм.	Арк.	№докум.	Підпис	Дата	Програмований генератор випробувальних сигналів на базі мікроконтролера АТМega328 для перевірки логічних мікросхем	Літера	Аркуш	Аркушів
Виконав		МАРКОВ	<i>М.С.Р.</i>	20.6.25		у	2	71
Перевір.		СТЕЦЮК	<i>С.Т.</i>		Тояснювальна записка	ХНУ КІ2с-22-2		
Н.контр.		Гетяна КИСІЛЬ	<i>Г.К.</i>	20.6.25				
Затвер.		Ольга ПАВЛЮВА	<i>О.П.</i>	20.6.25				

ВСТУП

У сучасній електроніці логічні мікросхеми широко використовуються в системах автоматизації, цифрових пристроях, мікроконтролерах та інших цифрових системах. Надійність і правильне функціонування таких мікросхем є критично важливими, особливо в умовах розробки, налагодження та технічного обслуговування складних електронних пристроїв. Одним з основних інструментів для перевірки та діагностики логічних елементів є генератор випробувальних сигналів.

Актуальність дослідження полягає в тому, що більшість існуючих генераторів мають обмежений функціонал, фіксовані шаблони сигналів або високу вартість. Водночас розвиток мікроконтролерної техніки дозволяє створити прості, гнучкі та економічно вигідні рішення для генерації різноманітних цифрових сигналів. Програмований генератор на базі мікроконтролера ATmega328 є доступним та ефективним інструментом, який дозволяє реалізувати різні режими подачі сигналів, швидко перемикає шаблони, а також інтегрувати засоби відображення інформації та керування.

Розробка такого пристрою дозволяє не лише вдосконалити процес тестування цифрових схем, але й слугує хорошою навчальною базою для вивчення принципів цифрової електроніки, мікроконтролерного програмування та симуляційного моделювання. У зв'язку з цим проєкт програмованого генератора випробувальних сигналів є актуальним як з практичної, так і з освітньої точки зору.

Метою даної роботи є перевірка логічних мікросхем різних типів за допомогою програмованого генератора випробувальних сигналів на базі мікроконтролера ATmega328.

Об'єктом дослідження є цифрові логічні мікросхеми та їх типові комбінації, що застосовуються в електронних пристроях керування, обробки сигналів та автоматики.

					КвРКІ. 22024.40.02.40 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

Предметом дослідження є програмно-апаратні методи автоматизованого формування шаблонів логічних сигналів та контролю правильності роботи логічних елементів з використанням мікроконтролера Arduino.

Практичне значення полягає в тому, що програмований генератор випробувальних сигналів на базі мікроконтролера ATmega328 є універсальним інструментом для перевірки справності логічних мікросхем, що дозволяє подавати на їх входи типові та користувацькі сигнали з різною частотою, послідовністю та логікою, що значно спрощує процес діагностики.

Завдяки гнучкості програмування користувач може змінювати шаблони сигналів без потреби втручання в апаратну частину, що скорочує час підготовки до випробувань і підвищує ефективність технічного обслуговування. Вбудовані кнопки керування та LCD-дисплей забезпечують зручний інтерфейс користувача, а модульна структура дозволяє легко адаптувати пристрій до різних задач.

Генератор може використовуватись у лабораторіях технічних навчальних закладів, у малих сервісних центрах, дослідницьких лабораторіях і приватних майстернях, де немає можливості використовувати дорогі професійні генератори.

					КвРКІ. 22024.40.02.40 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ОГЛЯД ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз існуючих рішень

Мікроконтролер – це компактний обчислювальний блок, інтегрований на одному кристалі, або пристрій, який об'єднує мікропроцесор, пам'ять та периферійні інтерфейси в одному корпусі [1].

Існують різні типи мікроконтролерів, включаючи сімейства мікроконтролерів Intel 8051, PIC, MSP, ARM та AVR. Ці мікроконтролери містять низку периферійних пристроїв, подібних до тих, що використовуються в комп'ютерах, включаючи компоненти введення та виведення, пам'ять, таймери, послідовні з'єднання для передачі даних та програмовані інтерфейси. Мікроконтролери знаходять застосування у вбудованому програмному забезпеченні та автономній роботі машин, охоплюючи офісне обладнання, електроінструменти, медичні прилади, системи керування, пульти дистанційного керування та електронні гаджети [2].

Arduino – це програмована плата, що працює на платформі з відкритим вихідним кодом. Вона широко застосовується у проєктах різного рівня складності, від простих до високотехнологічних у сфері технічної творчості, прототипування та систем автоматизації [3]. Екосистема Arduino включає як зручні програмні, так і апаратні компоненти. Програма Arduino IDE підтримує програмування на C та C++, дотримуючись певних рекомендацій [4].

Хоча існують різні альтернативні мікроконтролери серед доступних плат розробки, Arduino здобуває популярність завдяки своїй доступності, зрозумілому програмному середовищу та інклюзивній архітектурі апаратного забезпечення. Незважаючи на появу конкуруючих варіантів, існують додаткові фактори, що сприяють постійній популярності Arduino серед його широкої бази користувачів [5].

Управління різноманітними альтернативами апаратного забезпечення з відкритим кодом може бути складним завданням.

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 5
Зм.	Арк.	№ докум.	Підпис	Дата		

Arduino розробила різноманітний асортимент із понад 100 апаратних елементів, що охоплює різноманітні пропозиції, такі як плати, захисні екрани, носії, комплекти та додаткові аксесуари [6]. Ця велика колекція продуктів підкреслює зобов'язання Arduino задовольняти широкий спектр потреб користувачів та вимог проектів. З роками їхній каталог розширився, включивши низку опцій, які дозволяють аматорам, розробникам та професіоналам ефективно втілювати свої ідеї в життя, використовуючи універсальні та адаптивні апаратні рішення. Різні типи плат Arduino представлені на рисунку 1.1.



Рисунок 1.1 – Різні типи плат Arduino [6]

Arduino охоплює три окремі сімейства апаратного забезпечення: Nano, MKR та Classic. Кожне сімейство пропонує унікальний набір плат, розроблених для задоволення потреб різних проектів.

Сімейство Nano характеризується компактними розмірами та включає низку плат, таких як Nano Every, варіант початкового рівня, а також більш просунуті моделі, такі як Nano 33 BLE Sense та Nano RP2040, які можуть похвалитися додатковими функціями для різноманітних застосувань [7].

Сімейство MKR представляє колекцію плат, щитків та носіїв, які можна комбінувати для створення інноваційних проєктів. Це сімейство дозволяє користувачам використовувати поєднання апаратного забезпечення між різними компонентами, що дозволяє розробляти цікаві проєкти.

У сімействі Classic можна знайти такі плати, як добре відома Arduino Uno, а також перші класичні моделі, такі як Leonardo та Micro.

Під час вибору плати мікроконтролера ключову роль у визначенні її придатності для конкретного застосування відіграють певні характеристики. Такі фактори, як можливості мікропроцесора, обсяг пам'яті, параметри введення та виведення, а також фізичні розміри, впливають на вибір відповідної конструкції.

Наприклад, плата Lilypad виготовлена спеціально для електронного текстилю, оснащена сенсорними платами, які ідеально підходять для носимих пристроїв, і навіть їх можна прати для додаткової зручності. Коли проєкти потребують великої кількості входів і виходів, доцільно розглядати використання плат, таких як Arduino Due або Arduino Mega, які мають до 54 цифрових ліній вводу/виводу. Це робить їх ефективним вибором для складних систем, що потребують масштабного підключення [8].

Для більших програм або складних проєктів, що потребують значного обсягу пам'яті, можуть бути корисними такі плати, як Arduino M0 на базі ARM Cortex-M0+, яка має 256 КБ ПЗП та 32 КБ оперативної пам'яті.

Екосистема Arduino також пропонує спеціалізовані програми. Для аудіопроектів MKR Zero пропонує спеціальні функції. Arduino YN працює в парі з дистрибутивом Linux Linino OS, інтегруючи можливості Wi-Fi. Arduino Ethernet REV 3, заснований на платі Uno має підключення Ethernet та зчитувач карт micro SD, що робить його придатним для виконання мережевих завдань [9].

Для застосувань, що включають макетні плати, Arduino Nano пропонує компактну альтернативу [10]. Хоча там нема роз'єму живлення постійного струму та він має менший порт USB порівняно з Arduino Uno, Arduino Nano особливо добре підходить для проєктів, що включають схеми на макетних платах. Це

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

робить його гарним вибором для людей, які вже знайомі з електронікою, та тих, хто часто використовує макетні плати у своїй роботі.

Такий широкий асортимент плат дозволяє розробникам узгодити вимоги своїх проєктів з оптимальним апаратним рішенням, що робить Arduino універсальною платформою для інновацій.

Така властивість як адаптивність, дозволяє Arduino бути універсальним інструментом, що виходить за межі галузі, пропонуючи інноваційні концепції та практичні рішення для різноманітних застосувань.

Arduino можна застосовувати у різних сферах, таких як різні сектори оборонної галузі [11], медичне застосування [12], застосування в протезуванні та ортопедії [13] та освіти [14]. Але, його функціональність може бути обмеженою.

Розглянемо більш детально сфери застосування (таблиця 1.1):

1. Освітні проєкти. Arduino широко використовується у школах, коледжах та університетах для вивчення основ електроніки, програмування, схемотехніки та робототехніки.

Наприклад для моделювання системи світлофору, побудови макету «розумного будинку», вимірювання температури та виведення на LCD-дисплей, тощо.

2. Автоматизація та контроль. Arduino здатне автоматизувати різні процеси, від керування освітленням до промислових механізмів.

Наприклад, для автоматичного керування освітленням за допомогою датчика руху, для поливу рослин за вологістю ґрунту, для ідстеження рівня води в резервуарі.

3. Інтернет речей (IoT). За допомогою модулів Wi-Fi (ESP8266, ESP32) або Ethernet Arduino може підключатися до Інтернету та обмінюватися даними з хмарними сервісами.

Наприклад, віддалений моніторинг температури або вологості, онлайн-сигналізація або трекінг об'єктів, веб-інтерфейс для керування пристроями вдома.

4. Робототехніка. Arduino є «мозком» багатьох саморобних роботів і

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

автоматизованих систем.

Наприклад, робот, що слідує за лінією, об'їзд перешкод за допомогою ультразвукових датчиків, робоманіпулятори з керуванням по Bluetooth.

5. Збір і обробка даних. Завдяки датчикам Arduino може використовуватися для збору та аналізу різноманітних фізичних параметрів.

Наприклад, для погодніх станцій (збір даних щодо температури, вологості, атмосферного тиску), для пульсометра або детектора серцевого ритму, для вимірювання рівня освітлення або шуму.

6. Мистецтво та інтерактивні інсталяції. Arduino часто застосовують у цифровому мистецтві для створення реактивних інсталяцій, освітлення або звуку. Наприклад, світлодіодні панелі, що реагують на звук, інтерактивна скульптура, що змінюється залежно від наближення людини, світломузика.

7. Медичні та біомедичні пристрої. Хоча Arduino не сертифікований для медичного використання, він широко застосовується у прототипуванні медичних рішень. Наприклад, портативний пульсоксиметр, монітор тиску або температури пацієнта, пристрої нагадування про прийом ліків.

8. Тестери та генератори сигналів. Arduino здатне генерувати цифрові шаблони або зчитувати логічні сигнали, що корисно для перевірки цифрових мікросхем. Наприклад, генератор тестових сигналів для логічних мікросхем, побудова логічного аналізатора або осцилографа, сканування контактів у макетних платах.

Arduino має низку суттєвих переваг, які зробили її популярною серед студентів, інженерів і розробників прототипів.

По-перше, вона є відкритою апаратною платформою, що дозволяє вільно модифікувати, вдосконалювати та виготовляти власні варіації.

По-друге, Arduino має просте середовище програмування, яке базується на мові C/C++, що значно спрощує початок роботи для новачків. Бібліотеки та велика спільнота забезпечують легкий доступ до готових рішень, прикладів і підтримки. Крім того, вартість плати є низькою, що дозволяє створювати доступні

					КвРКІ. 22024.40.02.40 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

Кінець таблиці 1.1

Напрямок застосування	Типові компоненти	Приклади проєктів	Рівень складності
Мистецтво/Інсталяції	RGB-світлодіоди, мікрофони, датчики руху	Інтерактивне підсвічування, звукові візуалізації	Середня
Медичні прототипи	Датчики пульсу, температури, екрани	Пульсометр, нагадування про ліки	Висока
Тестери/генератори	Цифрові виходи, ZIF-сокети, дисплеї	Генератор логічних шаблонів, тестер мікросхем	Висока

1.2 Логічні вентиля

Логічні вентиля (ЛВ) – є життєво важливими елементами, які керують нашим цифровим всесвітом в епоху сучасних технологій. Ці передові технічні засоби відіграють важливу роль у контролі двійкових даних, які складають основу сучасних комп'ютерних технологій. Логічні вентиля є основними елементами цифрових схем, що використовуються в електронних пристроях, відповідальних за виконання логічних операцій, таких як І, АБО та НІ, на основі вхідних сигналів, що приймаються [15].

Логічні вентиля, як базові елементи цифрових схем, мають широкий спектр застосувань у різних галузях. Їх здатність виконувати логічні операції над вхідними сигналами робить їх необхідними в різних галузях, таких як інформатика, телекомунікації, електроніка та автоматизація.

Однак, зі збільшенням кількості логічних вентилів в схемі з'являються обмеження, такі як енергоспоживання та тепловиділення, що призводить до труднощів у підтримці загальної продуктивності системи. Їх здатність швидко

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

маніпулювати двійковими даними, отриманими за допомогою логічних операцій, формує основи цифрової схемотехніки.

За допомогою комбінації вентилів будуються суматори, тригери, регістри, мультиплексори, дешифратори та інші функціональні вузли мікропроцесорів. Архітектори цифрових систем проєктують схеми з мільйонів логічних елементів, що дозволяє реалізовувати багатоядерні процесори, графічні прискорювачі та системи на кристалі (SoC). Кожен логічний рівень – це одиниця або нуль, і завдяки строгій логіці цих сигналів, комп'ютерна система може приймати, обробляти і зберігати великі обсяги інформації. Процесори, які керують смартфонами, автомобілями, медичними пристроями та промисловими роботами – все це результат логічного проєктування на основі вентилів.

Сучасні засоби автоматизованого проєктування (EDA) допомагають інженерам формувати цифрові архітектури, перевіряти їхню функціональність і синтезувати у вигляді мікросхем. Логічні вентиля залишаються фундаментом будь-якої цифрової системи, незважаючи на зростаючу складність кінцевих продуктів.

Таким чином, інженери можуть проєктувати складні комп'ютерні архітектури, які забезпечують наш технологічний прогрес за допомогою комбінації та конфігурації різноманітних логічних вентилів.

Логічні вентиля, як базові елементи цифрових схем, мають широку колекцію застосувань у різних галузях. Їх здатність виконувати логічні операції над вхідними сигналами робить їх необхідними в різних галузях, таких як інформатика, телекомунікації, електроніка та автоматизація. Однак зі збільшенням кількості логічних вентилів у схемі з'являються обмеження, такі як споживання енергії та тепловиділення, що призводить до труднощів у підтримці загальної продуктивності системи. Логічні вентиля найбільше використовуються в цифрових системах з багатовимірними функціональними можливостями, питання забезпечення сумісності та інтероперабельності з іншими компонентами та підсистемами є важливим, а також, з розвитком технологій, існує постійна потреба

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

у збільшенні швидкості роботи логічних вентилів [15].

Отже, усунення недоліків у швидкодії є постійною проблемою.

У дослідженні авторів [16] згадуються різні типи логічних вентилів (NOT, OR, AND, NAND, NOR, XOR, XNOR) над якими був проведений ряд експериментів.

Вентиль NAND є універсальним вентиляем, який можна використовувати для побудови різних дуже великих інтегральних схем, таких як АБО, І-НІ, І, АБО та XOR. Вдосконалення конструкції вентилів NAND може призвести до покращення продуктивності систем НВІС.

Інверторний вентиль, який також називають NOT є базовим компонентом в електронних схемах. Йому потрібен на один вхід логічний 0 або 1 і він видає інвертований логічний вихід 1 або 0 відповідно.

Вентилі АБО, І також є базовим компонентом в електронних схемах. У математиці вентиль представляє процес множення, що означає, що на вихід вентилів впливають вхідні дані. Вентилі мають один вихід, який залежить від двох або більше входів, як того вимагає проектування схеми.

Розглянемо основні типи логічних вентилів з таблицями істиностей.

Логічний вентиль AND (І) - це базовий елемент цифрової логіки, який видає логічну «1» на виході лише тоді, коли всі його входи також знаходяться в стані «1». В іншому випадку результатом буде логічний «0». Такий вентиль реалізує кон'юнкцію, тобто операцію логічного множення (таблиця 1.2). Наприклад, у схемах безпеки або автоматизації, де два або більше сенсорів мають одночасно подати сигнал, щоб відбулась дія (наприклад, відкриття дверей при виявленні людини і введенні правильного коду).

Логічний вентиль OR (АБО) - видає логічну «1» на виході, якщо хоча б один із входів перебуває в стані «1». Вихід буде «0» лише тоді, коли всі входи дорівнюють нулю. Цей вентиль реалізує логічну операцію диз'юнкції (таблиця 1.3).

Наприклад, у сигналізаційних системах, якщо спрацює будь-який датчик

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

(руху, диму чи температури), на виході OR з'явиться «1», і система виконає відповідну реакцію.

Таблиця істинності вентиля OR показує, що лише комбінація (0, 0) дає «0», всі інші – «1».

Таблиця 1.2 – Таблиця істинності логічного вентиля AND (І)

Входи		Вихід
A	B	$C=A*B$
0	0	0
0	1	0
1	0	0
1	1	1

Таблиця 1.3 – Таблиця істинності логічного вентиля OR (АБО)

Входи		Вихід
A	B	$C=A*B$
0	0	0
0	1	1
1	0	1
1	1	1

Логічний елемент NOT (НІ), також називається інвертор – це один з найпростіших логічних вентилів, який реалізує логічне заперечення.

Його основним завданням є змінити логічний стан вхідного сигналу на протилежний. Тобто, якщо на вхід подається логічна «1» (високий рівень напруги), то на виході буде логічний «0» (низький рівень), і навпаки (таблиця 1.4).

Цей вентиль має лише один вхід і один вихід. Він є незамінним у цифрових схемах, оскільки інверсія сигналу потрібна майже у кожній складній логічній

комбінації. Наприклад, у схемах керування, де необхідно активувати дію при відсутності сигналу, інвертор використовується для створення зворотного ефекту. У процесорній логіці інвертори застосовуються в побудові суматорів, тригерів, логіки умовних переходів та логіки адресації. Фізично елемент NOT може бути реалізований на транзисторах, наприклад, у вигляді інвертора на одному n-p-n транзисторі. В сучасних інтегральних схемах, інвертори входять до складу більш складних логічних елементів або працюють у парі з іншими вентилями.

Інвертори також відіграють важливу роль у побудові елементів пам'яті, таких як бістабільні тригери, оскільки їх здатність «повертати» сигнал назад критично важлива для збереження стану. Крім того, вони використовуються для формування затримок у сигналі, генерації тактових імпульсів або створення генераторів імпульсів у цифрових генераторах.

Таблиця 1.4 – Таблиця істинності логічного вентиля NOT (НІ)

A	A'
0	1
1	0

Логічний вентиль NAND (НЕ-І) – це комбінаційний логічний вентиль, який виконує заперечення результату логічної операції «І» (AND). Іншими словами, вихід цього елемента буде логічною «1» у всіх випадках, крім того, коли всі його входи одночасно є одиницями, тоді на виході з'явиться логічна «0» (таблиця 1.5). Саме тому він і має назву «НЕ-І» (тобто інверсія AND).

Елемент NAND є надзвичайно важливим у цифровій схемотехніці, оскільки належить до так званих функціонально повних логічних елементів. Це означає, що за допомогою лише NAND-вентилів можна реалізувати будь-яку іншу логічну функцію: NOT, AND, OR, XOR та інші. Саме тому мікросхеми, побудовані винятково на NAND-елементах, здатні реалізовувати повноцінні логічні обчислення.

Фізично NAND-елементи будуються на основі транзисторних ключів (наприклад, у CMOS-технології з використанням пар P- та N-канальних MOSFET-транзисторів). Завдяки своїм електричним характеристикам, NAND-вентилі мають низьке енергоспоживання та високу швидкодію, що дозволяє використовувати їх у швидких цифрових пристроях.

Таблиця 1.5 – Таблиця істинності логічного вентиля NAND (НЕ-І)

Входи		Вихід
A	B	$C=(A*B)$
0	0	1
0	1	1
1	0	1
1	1	0

Найпоширенішою мікросхемою, що реалізує NAND, є 74LS00, яка містить чотири двовходові NAND-вентилі. Такі мікросхеми широко використовуються в навчальних лабораторіях, у побудові схем керування, регістрів, суматорів, мультиплексорів, тригерів і навіть повноцінних процесорних вузлів.

У цифрових автоматах і логічних контролерах NAND застосовується для реалізації умовних операцій: наприклад, якщо не всі сигнали відповідають умові, то активувати певну дію. Ця гнучкість робить NAND-вентиль основою для побудови надійних і простих логічних пристроїв.

Логічний вентиль NOR (НЕ-АБО) – це комбінаційний логічний елемент, який виконує інверсію результату логічної операції OR (АБО). Тобто, NOR видає логічну «1» тільки тоді, коли всі його входи дорівнюють логічному «0». В усіх інших випадках, коли хоча б один із входів має логічну «1», результатом буде логічна «0» (таблиця 1.6). Саме тому його називають НЕ-АБО, або інверсія OR.

До основних характеристик відносяться наступні:

- має два або більше входів (найчастіше — два);

- видає логічну одиницю тільки при повній відсутності активних сигналів;
- позначається в схемах заокругленим трикутником із кружечком на виході (позначення інверсії);
- у таблиці істинності – це точна протилежність OR.

Таблиця 1.6 – Таблиця істинності логічного вентиля NOR (НЕ-АБО)

Входи		Вихід
A	B	$C=(A+B)'$
0	0	1
0	1	0
1	0	0
1	1	0

NOR, як і NAND, є функціонально повним елементом, тобто з його допомогою можна побудувати будь-яку логічну функцію (NOT, AND, OR, XOR тощо). Застосовується в логіці обнулення, затримках, контролі виключення, генераторах імпульсів, тригерних схемах (наприклад, RS-тригер на основі двох NOR). NOR є важливою частиною багатьох мікропроцесорних архітектур та арифметико-логічних пристроїв.

Типовою мікросхемою є 74LS02, що містить чотири незалежні двовходові NOR-вентилі. У CMOS- або TTL-технологіях NOR-вентиль реалізується з використанням комбінації транзисторів, які забезпечують необхідну логічну поведінку, зокрема, інверсію логіки АБО.

Логічний вентиль NOR – це не просто інверсія OR, а складова для цифрових систем. Його здатність реагувати виключно на повну відсутність сигналу робить його цінним у схемах контролю та безпеки, де важливо активувати функцію лише тоді, коли жоден із сигналів не подано.

Вентиль XOR (виключне АБО) є одним із базових логічних елементів у цифровій електроніці. Його головна функція полягає в тому, щоб повертати

логічну одиницю (1) лише в тому випадку, коли вхідні сигнали відрізняються. Іншими словами, результат буде 1, якщо один із входів є 1, а інший є 0. Якщо обидва входи однакові, то результат 0 (таблиця 1.7). XOR виконує логічну нерівність між двома (або більше) сигналами. У побутовому розумінні це подібне до «виключного вибору»: тільки один, але не обидва. XOR можна реалізувати апаратно (на рівні мікросхем, наприклад, за допомогою серії 74LS86) або програмно, використовуючи логічні вирази.

Таблиця 1.7 – Таблиця істинності логічного вентиля XOR (виключне АБО)

Входи		Вихід
A	B	$C=(A\oplus B)$
0	0	0
0	1	1
1	0	1
1	1	0

У цифрових схемах XOR часто використовується в арифметичних логічних пристроях (АЛП), зокрема в суматорах (adders) для обчислення суми двох бітів. Також застосовується в шифруванні.

Якщо сигнал зашифрувати за допомогою XOR і того самого ключа, розшифрування виконується повторним застосуванням XOR. XOR корисний у порівнянні даних (наприклад, перевірка на відмінність), а також у контрольних сумах і CRC.

Характерними властивостями є те, що порядок входів не має значення, та будь-яке число XOR із самим собою дає 0. У схемах XOR зручно демонструється за допомогою двох світлодіодів: світиться лише тоді, коли рівно один із перемикачів увімкнений.

XNOR (Виключне НЕ-АБО, іноді називається еквівалентним вентилям) – це логічний вентиль, який є інверсією вентиля XOR. Тобто, XNOR дає логічну

одиницю (1), коли всі входи однакові, і логічний нуль (0) тоді, коли входи різні (таблиця 1.8).

Таблиця 1.8 – Таблиця істинності логічного вентиля XNOR (виключне НЕ-АБО)

Входи		Вихід
A	B	$C = \overline{(A \oplus B)}$
0	0	1
0	1	0
1	0	0
1	1	1

Властивостями є симетричний вентиль. Унікальною особливістю виявлення рівності є $XNOR = 1$ тоді й тільки тоді, коли входи рівні. Добре підходить для порівняння бітів: якщо A дорівнює B, то результат = 1.

У цифровій логіці XNOR використовується для контролю помилок, виявлення збігів, побудови компараторів. Також застосовується в шифруванні, арифметичних операціях, логічних контролерах. У пристроях перевірки парності (наприклад, у передаванні даних через UART або I2C) XNOR може бути використаний для перевірки, чи чітна кількість одиниць у наборі бітів.

У схемах на Arduino вентиль XNOR можна реалізувати програмно як частину алгоритму перевірки. Для візуалізації можна підключити два перемикачі до входів, і LED до виходу: світлодіод світиться тоді, коли обидва перемикачі в однаковому положенні (обидва ON або обидва OFF).

Таким чином, XNOR є логічним вентиляем, який забезпечує високий рівень на виході при рівності входів, і широко використовується у задачах порівняння, виявлення парності, кодування/декодування та в математичних блоках цифрових схем.

Завдяки цим елементарним логічним операціям можливо реалізувати складні цифрові схеми: суматори, компаратори, шифратори, пам'ять, контролери та мікропроцесори. Усі логічні вентиля активно використовуються в проектуванні цифрових систем, зокрема з мікроконтролерами Arduino та іншими платформами. Їх правильне комбінування дозволяє створювати пристрої, які керують, обробляють та передають інформацію у сучасному світі електроніки.

1.3 Висновки до першого розділу

У межах розділу 1 проведено огляд мікроконтролерної платформи Arduino та основних логічних вентилів. В результаті проведеного огляду, доведено, що Arduino є доступним, гнучким і функціональним інструментом для швидкої розробки та тестування електронних систем, що дає змогу створювати як навчальні, так і промислові прототипи. Платформа має широку апаратну підтримку, велику кількість бібліотек і активну спільноту, що спрощує роботу навіть початківцям. Особливе значення має її здатність керувати виводами для подачі або зчитування логічних сигналів, що робить її ідеальною для роботи з логічними мікросхемами. Водночас логічні вентиля виконують базові функції обробки двійкової інформації, які лежать в основі роботи будь-якої цифрової схеми. Зокрема, вентиля AND, OR, NOT, NAND, NOR, XOR та XNOR дозволяють реалізовувати логічні операції, необхідні для побудови арифметичних пристроїв, контролерів, тригерів та логічних процесорів. Їх комбінації створюють складні схеми, що забезпечують обробку, зберігання та передачу інформації.

Проведений огляд показав, що Arduino дозволяє не лише керувати логічними рівнями, але й виводити інформацію на дисплеї, взаємодіяти з кнопками, сенсорами та іншими пристроями. Крім того, Arduino дозволяє моделювати та перевіряти різні шаблони подачі сигналів, що є корисним для аналізу поведінки логічних мікросхем у реальному часі.

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ПРОГРАМОВАНОГО ГЕНЕРАТОРА ВИПРОБУВАЛЬНИХ СИГНАЛІВ НА БАЗІ МІКРОКОНТРОЛЕРА ATMEGA328

2.1 Методи перевірки мікросхем

Методи перевірки мікросхем є важливою частиною процесу розробки, виготовлення та експлуатації електронних пристроїв. Основна мета таких перевірок – це виявлення несправностей, дефектів виготовлення або порушень у логічній роботі мікросхеми. Тестування інтегральних мікросхем (ІС) необхідне для того, щоб забезпечення їх функціонування та довговічність, оскільки ІС є критично важливими компонентами в сучасній електроніці. Для тестування інтегральних схем часто використовують цифрові тестери, що стають все більш складними з розвитком технологій.

Існують різні методи для перевірки мікросхем, які можна умовно поділити на апаратні й програмні, ручні та автоматизовані. Вибір методу залежить від типу мікросхеми, доступного обладнання та вимог до точності.

Розглянемо найпоширеніші підходи:

1. Метод функціонального тестування (Functional Testing). Перевіряється відповідність вхідних і вихідних сигналів відповідно до логічної таблиці істинності. Наприклад, для мікросхеми AND на входи подаються всі можливі комбінації 0 і 1, а на виході фіксується результат. Такий підхід добре підходить для простих логічних елементів [17].

2. Метод тестування з використанням генератора шаблонів. Програмований генератор (наприклад, на базі Arduino) подає на входи мікросхеми певні шаблони логічних сигналів, а результат аналізується вручну або автоматично. Це дозволяє імітувати типові режими роботи [18].

3. Тестування в логічному аналізаторі. Логічний аналізатор записує і візуалізує цифрові сигнали на виводах мікросхеми, дозволяючи проаналізувати їх у часовому контексті. Дає змогу виявити помилки синхронізації або нестабільні

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

стани [19].

4. Метод циклічного перемикання входів при якому створюється цикл, у якому логічні входи мікросхеми змінюються послідовно, а результати порівнюються з еталонними. Такий підхід зручно реалізувати у віртуальному середовищі (Tinkercad, Proteus) [20].

5. Перевірка на стенді з візуальними індикаторами (світлодіодами) – це простий спосіб візуального контролю роботи ІС. Логічні стани на виходах мікросхеми виводяться через резистори на світлодіоди, які змінюють стан відповідно до сигналів.

6. Використання симуляторів (Proteus, Tinkercad, Multisim). Віртуальні середовища дозволяють моделювати поведінку мікросхем без фізичного збирання схем. Це ефективно на етапі проектування або навчання.

7. Метод самотестування (Built-in Self-Test, BIST). У складніших мікросхемах (ПЛІС, мікроконтролери) можуть бути вбудовані модулі самоперевірки. Це дозволяє виявляти несправності автоматично при запуску пристрою.

8. Метод інверсного аналізу (Fault Injection). В систему штучно вводяться помилки або змінюються сигнали, щоб перевірити, як вона на них реагує. Це корисно для оцінки надійності логіки.

Ці методи можуть комбінуватись, наприклад: генератор сигналів + візуальний індикатор + вивід на LCD або послідовний порт для аналізу.

Ефективна перевірка мікросхем є критично важливим етапом у забезпеченні надійності та працездатності електронних систем. Кожен метод має свої переваги та області застосування: ручні тести зручні для навчання та прототипування, тоді як автоматизовані системи забезпечують швидкість і точність у масовому виробництві. Особливе місце займає використання Arduino та генераторів тестових сигналів, які дозволяють створювати гнучкі та доступні засоби перевірки логіки мікросхем. Загалом, правильний вибір методу тестування залежить від складності мікросхеми, вимог до точності та обсягів перевірки. Комплексне

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

застосування різних методів забезпечує високу якість тестування, сприяє виявленню дефектів ще на ранніх етапах і гарантує стабільну роботу електронних пристроїв у реальних умовах експлуатації.

Порівняльні характеристики методів перевірки логічних мікросхем наведені в таблиці 2.1.

Таблиця 2.1 – Методи перевірки логічних мікросхем

Метод	Переваги	Недоліки	Складність реалізації	Тип застосування
Функціональне тестування	Висока точність, проста логіка	Потрібно вручну подавати всі комбінації	Низька	Ручне, навчальне
Генератор шаблонів (Arduino)	Автоматизація, повторюваність, просте розширення	Обмежена кількість пінів, потреба в прошивці	Середня	Навчальне, експериментальне
Логічний аналізатор	Часова діагностика, точність	Дороговартісне обладнання, складність інтерпретації	Висока	Професійне тестування
Циклічне перемикавання входів	Повне охоплення логіки, легко симулюється	Збільшений час перевірки	Середня	Програмна перевірка, симуляція
Світлодіоди на виході	Візуалізація, зручність, швидкість	Лише ручний контроль, не придатний для складної логіки	Низька	Початкове тестування

Кінець таблиці 2.1

Метод	Переваги	Недоліки	Складність реалізації	Тип застосування
Симулятори (Tinkercad, Proteus)	Безпечність, без потреби в фізичному обладнанні	Не охоплює всі реальні фактори	Низька, середня	Освіта, моделювання, дизайн
Самотестування (BIST)	Автономність, висока швидкість	Доступно не в усіх мікросхемах	Висока	Вбудовані системи
Інверсний аналіз (Fault Injection)	Оцінка надійності, моделювання відмов	Ризик пошкодження мікросхеми, складне моделювання	Висока	Дослідження, тестування на стійкість

В роботі [21] автори провели огляд цифрових тестерів, включаючи ручні, напівавтоматичні та автоматичні моделі. Авторі детально зупиняються на тому, як цифрові тестери можуть подолати труднощі тестування ІС, такі як тривалість і витрати. В роботі підкреслено цінність цифрових тестерів для прискорення процесу тестування та розширення його масштабу.

Щоб зробити тестування цифрових інтегральних схем більш доступнішим і дешевшим, автор роботи [22] пропонує недорогий тестер на основі графічного інтерфейсу. Тестер має вбудований тактовий генератор і настроювані рівні напруги, що робить його придатним для тестування різних ІС, від найпростіших до найскладніших.

Цифровий тестер, створений шляхом змішування сигналів ІМС на основі ПЛІС описано в дослідниками в роботі [23]. Тестер має такі функції, як генератор сигналів і осцилограф, щоб переконатися, що цифрові та аналогові схеми

функціонують правильно. Тестуючи різноманітні мікросхеми зі змішаними сигналами, автори доводять, що пристрій працює так, як заявлено.

Для того, щоб скоротити процес тестування цифрових ІС, автори роботи [24] пропонують новий підхід. Метод використовує стиснуті тестові вектори, щоб зменшити час, витрачений на тестування. Автори доводять ефективність методу, використовуючи як симуляції, так і точне тестування за допомогою тестера ІС.

Автор роботи [25] детально описав розробку змінного цифрового тестера ІС на базі платформи швидкого прототипування ARMIntegrator для швидкого прототипування. Було протестовано тестер, провівши експерименти на декількох мікросхемах і порівняно результати з результатами інших доступних інструментів. Пристосованість тестера до різних типів мікросхем, завдяки своїй конфігурованості, робить його гнучким інструментом тестування.

В роботі [26] автори представили дизайн тестера цифрових ІС на основі мікроконтролера, здатного проводити як мульти-тестування, так і тестування контурів. Тестер може досліджувати різні електронні мікросхеми, в тому числі з програмованою логікою та пам'яттю. Ефективне тестування для великої кількості мікросхем одночасно, тепер можна проводити за допомогою функцій циклічного тестування і мультитестування. Тестується багато мікросхем, а результати порівнюються з іншими тестерами, щоб довести ефективність тестера.

Оскільки цифрові тестери мають вирішальне значення для тестування інтегральних схем, то вони стають все більш складними з розвитком технологій. Цифрові тестери стикаються з такими проблемами, як час і вартість тестування.

Для зниження вартості та прискорення процесу тестування сучасні цифрові тестери все частіше використовують паралельну перевірку кількох мікросхем одночасно. Також активно впроваджуються адаптивні алгоритми тестування, які скорочують кількість необхідних тестових векторів без втрати точності.

Зростає значення програмно-керованих платформ, як-от Arduino, які дозволяють швидко налаштувати тестові послідовності під конкретні задачі. Водночас важливим напрямом є розробка нових методів самотестування

мікросхем, що дозволяють виявляти помилки вже під час їхньої експлуатації.

На рисунку 2.1 представлено структурну схему тестера цифрових ІС.

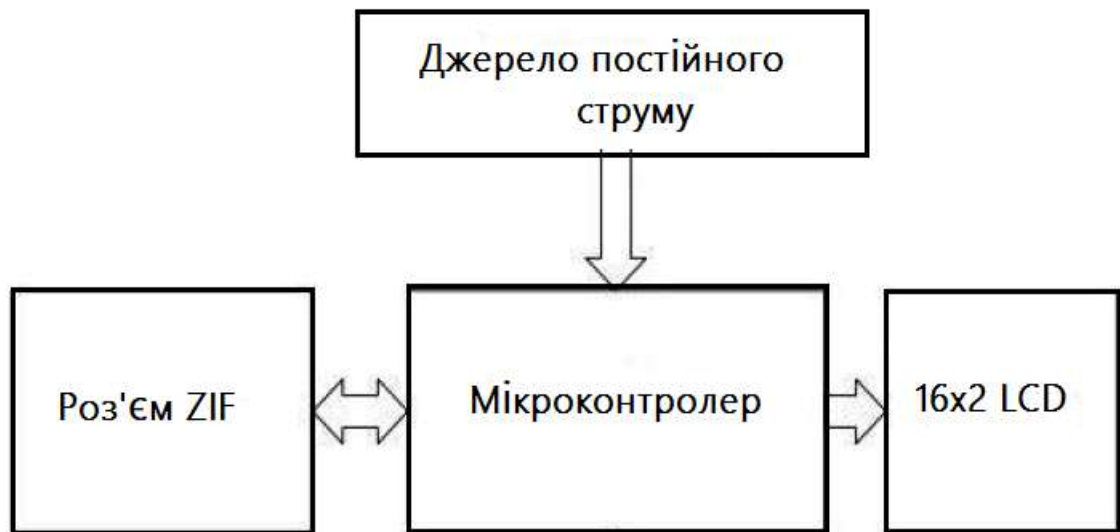


Рисунок 2.1 – Структурна схема тестера цифрових ІС

DC power supply (джерело постійного струму) – це електронний пристрій або схема, що забезпечує стабілізовану напругу постійного струму (DC, direct current) для живлення електронних компонентів, схем або пристроїв. На відміну від змінного струму (AC), який змінює напрямок і амплітуду з певною частотою (наприклад, 50 Гц у побутовій мережі), постійний струм тече лише в одному напрямку та має сталу напругу.

DC power supply може мати різні форми, від простого блока живлення з фіксованою напругою (наприклад, 5V або 12V) до лабораторного регульованого джерела, яке дозволяє точно налаштувати вихідну напругу й обмеження струму. Такі джерела часто включають стабілізатори напруги, захист від перевантаження, індикацію параметрів та можливість тонкого регулювання.

Для проєктів Arduino типове джерело – це адаптер на 9V або 12V, який підключається до роз'єму живлення або через роз'єм USB. Крім того, часто використовуються DC-джерела живлення для живлення макетів логічних схем через контактні плати (breadboards), забезпечуючи стабільні 3.3V або 5V.

ZIF-сокет (Zero Insertion Force) – це спеціальний тип роз'єму, призначений для безпечного та багаторазового підключення мікросхем без прикладання значного фізичного зусилля (рисунок 2.2). Його конструкція передбачає наявність рухомого механізму, зазвичай важеля, який відкриває або замикає контактні гнізда. У відкритому стані ніжки мікросхеми легко вставляються в роз'єм без опору, що виключає ризик їх пошкодження. Після вставлення мікросхеми, важіль опускається, і гнізда надійно затискають її контакти. Це особливо важливо для лабораторних умов, де потрібно часто змінювати мікросхеми для тестування або програмування. ZIF-сокети широко застосовуються у програматорах мікроконтролерів, логічних тестерах і розробницьких стендах. Вони сумісні з різними типами корпусів мікросхем, найчастіше DIP (dual in-line package). Завдяки мінімізації механічного навантаження, ZIF-сокети значно підвищують надійність і довговічність контактів. У проєкті генератора тестових сигналів такий сокет зручно використовувати для швидкої перевірки кількох логічних мікросхем без пайки.



Рисунок 2.2 – ZIF-сокет [27]

Порти 1 і 2 з'єднують мікроконтролер з роз'ємом ZIF. Ці роз'єми роблять передачу даних між ZIF-роз'ємом і мікроконтролером простою та

стандартизованою. Після отримання даних від мікроконтролера, мікроконтролер може маніпулювати ZIF-роз'ємом.

Результати та інші дані відображаються на рідкокристалічному дисплеї (РК-дисплеї). РК-дисплеї чітко і лаконічно показують результати тестів, дозволяючи оператору робити обґрунтовані висновки про життєздатність мікросхеми. Оскільки мікроконтролер може спілкуватися з РК-дисплеєм, то інформація яка відображається є одночасно динамічною і такою, що легко читається.

Цифровий тестер інтегральних схем складається з роз'єму ZIF, мікроконтролера і РК-дисплея. Разом вони забезпечують зручний метод тестування інтегральних схем.

Дискретне вейвлет-перетворення (DWT) може бути використовуватися в цифрових тестерах інтегральних схем як метод аналізу сигналів. У цьому застосуванні DWT може використовуватися для аналізу форми сигналів, що використовуються для тестування інтегральних схем, таких як тактові сигнали, сигнали живлення і сигнали даних.

DWT може витягувати характеристики з форми сигналу, такі як частота, амплітуда і фаза, які можуть допомогти виявити і діагностувати проблеми в інтегральних схемах [28]. Наприклад, DWT може виявити високочастотний шум або дзвін в тактових сигналах, що може вказувати на проблеми в мережі розподілу тактової частоти. Крім того, DWT можна використовувати для аналізу енергоспоживання інтегральних схем, що є важливим фактором у багатьох додатках. Аналізуючи сигнал енергоспоживання, DWT може допомогти ідентифікувати джерела споживання та визначити ефективність інтегральних схем.

РК-дисплей 16×2 складається з 16 стовпчиків і 2 рядків символів, що забезпечує 32 символи на екран. Він може відобразити алфавітно-цифрові символи, що робить його ідеальним для відображення тестових даних та іншої відповідної інформації. Щоб інтегрувати РК-дисплей 16×2 в цифровий тестер, можна підключити його до плати мікроконтролера використовуючи її цифрові контакти вводу/виводу. Плата може бути запрограмована на відображення певної

інформації на РК-дисплеї на основі результатів процесу тестування.

Крім того, РК-дисплей може відображати підказки користувача і меню, що дозволяє користувачеві взаємодіяти з цифровим тестером і виконувати різні функції тестування. Це може покращити загальний досвід користувача і зробити процес тестування більш ефективним.

16×2 LCD-модуль – це рідкокристалічний дисплей, який часто використовується у вбудованих проектах завдяки своїй низькій вартості, доступності та простоті використання для програмістів. Він складається з 16 стовпчиків і 2 рядків, загалом 32 символи, які можуть бути відображені. Кожен символ на РК-дисплеї 16×2 складається з 5×8 піксельних точок, що контролюються серією електродів та підсвічуванням. Напряга, що подається на електроди визначає стан пікселів і, таким чином вигляд символів на дисплеї. Для взаємодії з мікроконтролером або іншим електронним пристроєм, РК-дисплей 16×2 зазвичай має набір виводів даних і керування, які необхідно підключити. Виводи даних використовуються для передачі символічних даних на дисплей, в той час як керуючі виводи використовуються для керування операціями дисплея, такими як встановлення положення курсору або увімкнення та вимкнення дисплею.

Програмісти можуть використовувати бібліотеки або низькорівневе програмування для керування РК-дисплеєм 16×2 та відображення символів на ньому. Модуль можна використовувати для різних цілей, наприклад, для відображення тексту або стану інформації, створення користувацьких інтерфейсів або налагодження вбудованих систем.

Нам потрібен РК-дисплей 16×2 в цифровому тестері для інтегральних схем з кількох причин:

1. Відображення результатів тестування. РК-дисплей може відображати важливу інформацію про мікросхему, що тестується, наприклад, результати тестування та повідомлення про помилки. Ця інформація має вирішальне значення для користувача, щоб визначити, чи пройшла мікросхема процес

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

тестування.

2. Взаємодія з користувачем. РК-дисплей також може відображати підказки користувача і меню, що дозволяє користувачам взаємодіяти з цифровим тестером і виконувати різні функції тестування. Це покращує загальний користувацький досвід і робить процес тестування більш ефективним.

3. Налаштування. У разі виникнення помилок або несправностей в процесі тестування, РК-дисплей може відображати налагоджувальну інформацію та повідомлення про помилки. Це може допомогти виявити проблему і виправити її.

4. Моніторинг в реальному часі. РК-дисплей може відображати в реальному часі дані моніторингу, такі як рівні напруги, струму та інші параметри під час тестування. Це може допомогти проаналізувати поведінку мікросхеми та виявити несправності.

Загалом, РК-дисплей 16×2 є важливим компонентом цифрового тестера інтегральних схем, забезпечуючи простий і зручний спосіб відображення важливої інформації та взаємодії з тестовою системою.

Після введення даних про інтегральну мікросхему, мікроконтролер (MCU) отримує відповідну інформацію про мікросхему з своєї бази даних. Ця інформація включає очікувану поведінку вихідних сигналів мікросхеми та тестові умови, які необхідно застосувати до мікросхеми.

Потім MCU застосовує умови тестування, які генерує цифровий генератор сигналів, до мікросхеми. Вихідні сигнали мікросхеми потім фіксуються осцилографом і аналізуються за допомогою DWT. DWT використовується для вилучення характеристик з форми сигналу, таких як частота, амплітуда і фаза, які можна порівняти з очікуваною поведінкою ІС.

Якщо вихідні сигнали ІС відповідають очікуваній поведінці, то ІС вважається хорошою. І навпаки, якщо вихідні сигнали не збігаються з очікуваною поведінкою, то ІС вважається неправильною.

Результат тестування відображається на РК-дисплеї 16x2, а також з'являється повідомлення, що вказує на те, чи є мікросхема справною чи

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 30
Зм.	Арк.	№ докум.	Підпис	Дата		

несправною.

Таким чином, система може швидко і точно тестувати різні цифрові мікросхеми, в тому числі на основі DTL, TTL і CMOS технологіях, таких як 74xx серії. DWT відіграє важливу роль в цьому процесі, оскільки він надає як часову, так і частотну інформацію про сигнали, яка необхідна для точного виявлення та діагностики проблем в ІС, що тестуються.

2.2 Основні характеристики мікроконтролера АТМega328

Мікроконтролер Arduino АТМЕГА-328 складається з 14 вхідних та вихідних аналогових і цифрових контактів (з цих 6 контактів вважаються контактами ШІМ (широко-імпульсної модуляції)), 6 аналогові входи та решта цифрових входів (рисунок 2.3). Роз'єм живлення використовується для підключення плати Arduino до комп'ютера. Кабель з роз'ємом живлення (Power jack) використовується для підключення плати Arduino до комп'ютера. Зовнішнє джерело живлення (наприклад, батарея) підключається до мікроконтролера Arduino для забезпечення електроживлення.



Рисунок 2.3 – Мікроконтролер АТМega328 [29]

Arduino – це мікроконтролер з відкритим вихідним кодом, який не має

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

вбудованого зворотного зв'язку у своїй архітектурі. Ця плата Arduino має I2C-шину, що дозволяє здійснювати передавання даних від плати Arduino до зовнішніх виконавчих пристроїв.

Програмування плати Arduino здійснюється через послідовне з'єднання RS232 з мікроконтролерами серії ATmega. Робоча напруга плати становить 5 В, рекомендована вхідна напруга – від 7 В до максимуму 12 В. Постійний вхідний струм, що подається на плату Arduino, знаходиться в межах 40 мА.

Arduino містить різні типи пам'яті, зокрема Flash-пам'ять, EEPROM та SRAM. Довжина плати Arduino становить приблизно 68,64 мм, а ширина мікроконтролера близько 53,4 мм. Вага мікроконтролера Arduino становить приблизно 20 грамів.

У таких платах можуть використовуватись різні типи мікроконтролерів, зокрема 8-бітні мікроконтролери AVR від Atmel та 32-бітні мікропроцесори ARM від Atmel. Залежно від потреб, ці процесори застосовуються як у навчальних, так і в інженерних або промислових проєктах.

До прикладів промислового застосування Arduino належить керування виконавчими механізмами та сенсорами, що дозволяє реалізовувати автоматизовані системи контролю та моніторингу.

Проєктування цифрового тестера для інтегральних схем передбачає послідовне проходження кількох важливих етапів. Перш за все, визначається мета тестування, тобто які саме мікросхеми перевірятимуться, які логічні функції реалізують і які помилки потрібно виявляти.

Далі виконується аналіз технічних вимог до тестера: кількість входів/виходів, швидкодія, типи сигналів, інтерфейс користувача та необхідність відображення результатів. На основі цих даних створюється функціональна схема майбутнього пристрою, що включає контролер (наприклад, Arduino), генератор тестових сигналів, модуль зчитування відповідей і систему індикації або реєстрації результатів. Наступний етап – це вибір елементної бази: мікроконтролер, LCD-дисплей, кнопки, роз'єми, ZIF-сокет для мікросхем,

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 32
Зм.	Арк.	№ докум.	Підпис	Дата		

резистори, світлодіоди тощо. Після цього виконується розробка схеми підключення і створення прототипу, тобто фізичної макетної плати або моделі в симуляторі, наприклад, Tinkercad Circuits. Важливо також підготувати програмне забезпечення, яке буде керувати тестовим процесом: подавати послідовності логічних сигналів, аналізувати відповіді й виводити результати.

Після збирання тестера виконується початкове налагодження: перевірка правильності підключень, тестування кожного модуля окремо. Потім здійснюється повноцінна перевірка всіх функцій на реальних логічних мікросхемах. За необхідності вносяться корекції у схему або код.

На завершальному етапі оформлюється документація, включно з описом роботи тестера, схемою, прошивкою та інструкцією користувача

Розглянемо етапи проектування цифрового тестера для інтегральних схем. Розробка цифрового тестера для друкованих плат включає в себе різні компоненти та етапи, щоб гарантувати працездатність і надійність електронних пристроїв. Деякі з частин і процедур, які входять до створення цифрового тестера, наведені нижче:

1. До апаратного забезпечення входять апаратні елементи тестерів, такі як інтерфейс, з'єднання, генератори сигналів і вимірювальні пристрої.

2. Програмне забезпечення для тестування – це програма, яка керує випробувальним обладнанням і виконує випробування. Вона також збирає і аналізує дані випробувань.

3. Специфікація тесту охоплює необхідну функціональність гаджета, що тестується, тестові кейси, які потрібно провести, і параметри проходження/не проходження

4. Архітектура тесту відноситься до компонування тестових програмних і апаратних компонентів. Цей крок включає визначення необхідних елементів, створення конфігурації тестового обладнання, побудови процесу тестування та побудови алгоритмів, необхідних для виконання тестових кейсів.

5. Реалізація тесту – це етап, на якому тестове обладнання та програмне забезпечення створені та інтегруються.

6. На цьому етапі сконструйований цифровий тестер тестується, щоб переконатися, що він задовольняє критеріям, зазначеним у специфікації тесту. Тестові кейси виконуються на цільовому пристрої, а результати аналізуються як частина процесу валідації.

7. Обслуговування тесту є безперервним етапом, що включає в себе обслуговування цифрового тестера і внесення будь-яких необхідних змін або коригувань, щоб залишатися практичним і актуальним.

Розробка цифрового тестера для інтегральних схем складається з різних компонентів і етапів, які працюють разом, щоб забезпечити функціонування і надійність електронних пристроїв. Процес включає ретельне планування, проектування, реалізацію, валідацію та обслуговування для створення високоякісного цифрового тестера.

2.3 Моделювання схеми генератора логічних шаблонів

Метою моделювання було побудувати схему генератора логічних шаблонів, що виводяться на цифрові порти мікроконтролера Arduino Uno, та забезпечити індикацію поточного стану за допомогою LCD-дисплея. Сигнали призначені для тестування логічних мікросхем і електронних блоків.

Розглянемо детальніше структуру моделі (рисунок 2.4).

До складу віртуальної схеми увійшли наступні компоненти:

- Arduino Uno R3 як основний контролер;
- LCD-дисплей 16x2 з інтерфейсом I2C для виводу номера активного шаблону;
- світлодіоди трьох кольорів (синій, помаранчевий, зелений), підключені до цифрових виходів 2, 3 і 4;
- резистори по 220 Ом для обмеження струму світлодіодів;
- кнопка з підтягуючим резистором для ручного перемикання шаблонів у тестовому режимі.

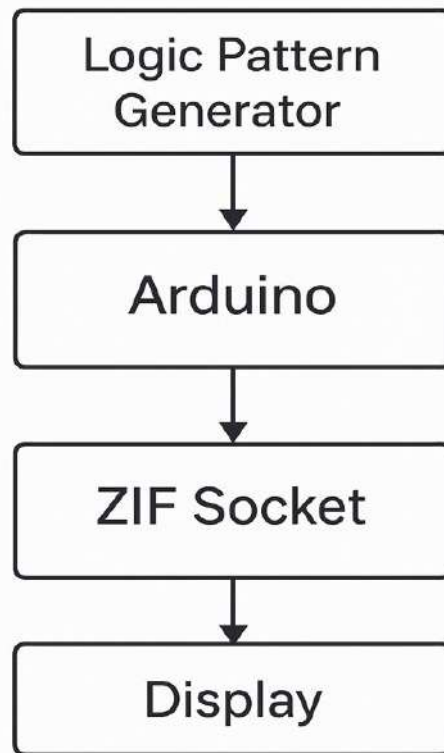


Рисунок 2.4 – Структурна схема моделі

Мікроконтролер ATMEGA328P, представлений на рисунку 2.5, є мікроконтролером AVR, який працює з 8-бітними даними. Він має вбудовану пам'ять 32 Кб і 1 Кб EEPROM.

Робоча напруга ATmega-328P коливається від 3,3 В до 5,5 В, причому 5 В є стандартом. У проектах, де доступна пам'ять визначає кількість символів, які можуть бути відображені, мікроконтролер ATMEGA328P є підходящим вибором завдяки своєму об'єму пам'яті, портам вводу/виводу та аналоговим/цифровим каналам. Тому для цього проекту було обрано мікроконтролер ATMEGA328P.

Шаблон – це попередньо визначена комбінація логічних рівнів, яка одночасно подається на кілька виходів Arduino.

Шаблони логічних сигналів у генераторі випробувальних сигналів потрібні для систематичного тестування цифрових мікросхем, зокрема логічних елементів (AND, OR, NOT, NAND тощо). Вони дозволяють подавати на входи мікросхеми заздалегідь визначені комбінації логічних рівнів (HIGH або LOW), щоб перевірити, чи правильно вона реагує на стандартні вхідні умови.

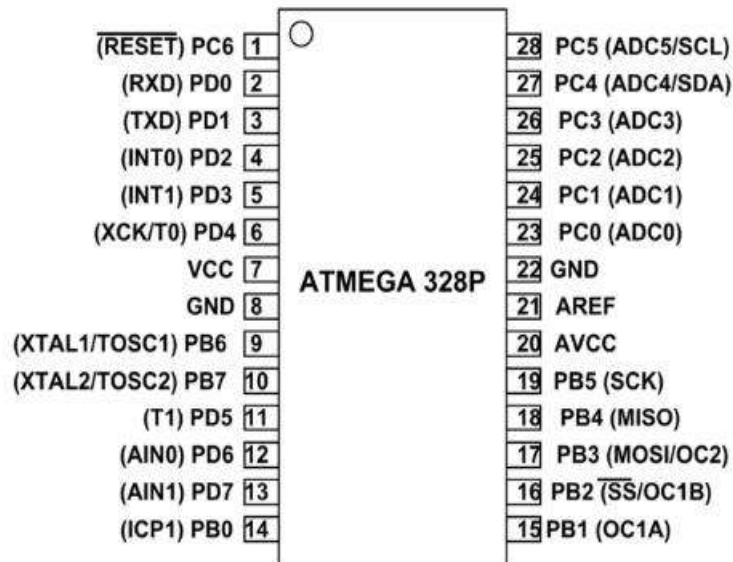


Рисунок 2.5 – Мікроконтролер ATMEGA328P [30]

Це особливо важливо під час налагодження схем або при контролі якості готових пристроїв. Завдяки шаблонам, можна імітувати типові вхідні ситуації, не формуючи вручну кожного окремого сигналу. Кожен шаблон є прикладом певного логічного стану, який відповідає одній комбінації вхідних змінних. Наприклад, якщо потрібно протестувати логічний вентиль AND, шаблон із усіма HIGH на входах дозволить перевірити, чи на виході буде очікуваний HIGH. У той же час шаблон із лише одним HIGH і рештою LOW допоможе переконатися, що мікросхема не видає помилкових результатів.

Такі шаблони виводяться на виходи Arduino послідовно, автоматично, що значно підвищує ефективність і точність перевірки. Отже, шаблони – це основа тестового впливу, яка забезпечує повторюваність, точність і автоматизацію процесу діагностики цифрових схем.

Блок-схема алгоритму генератора шаблонів – це графічне представлення послідовності дій, які виконує мікроконтролер Arduino для створення, виводу та перемикавання логічних сигналів на своїх вихідних пінів (рисунок 2.6). Така блок-схема дозволяє візуально уявити логіку функціонування пристрою, що особливо корисно при проектуванні, налагодженні або документуванні системи.

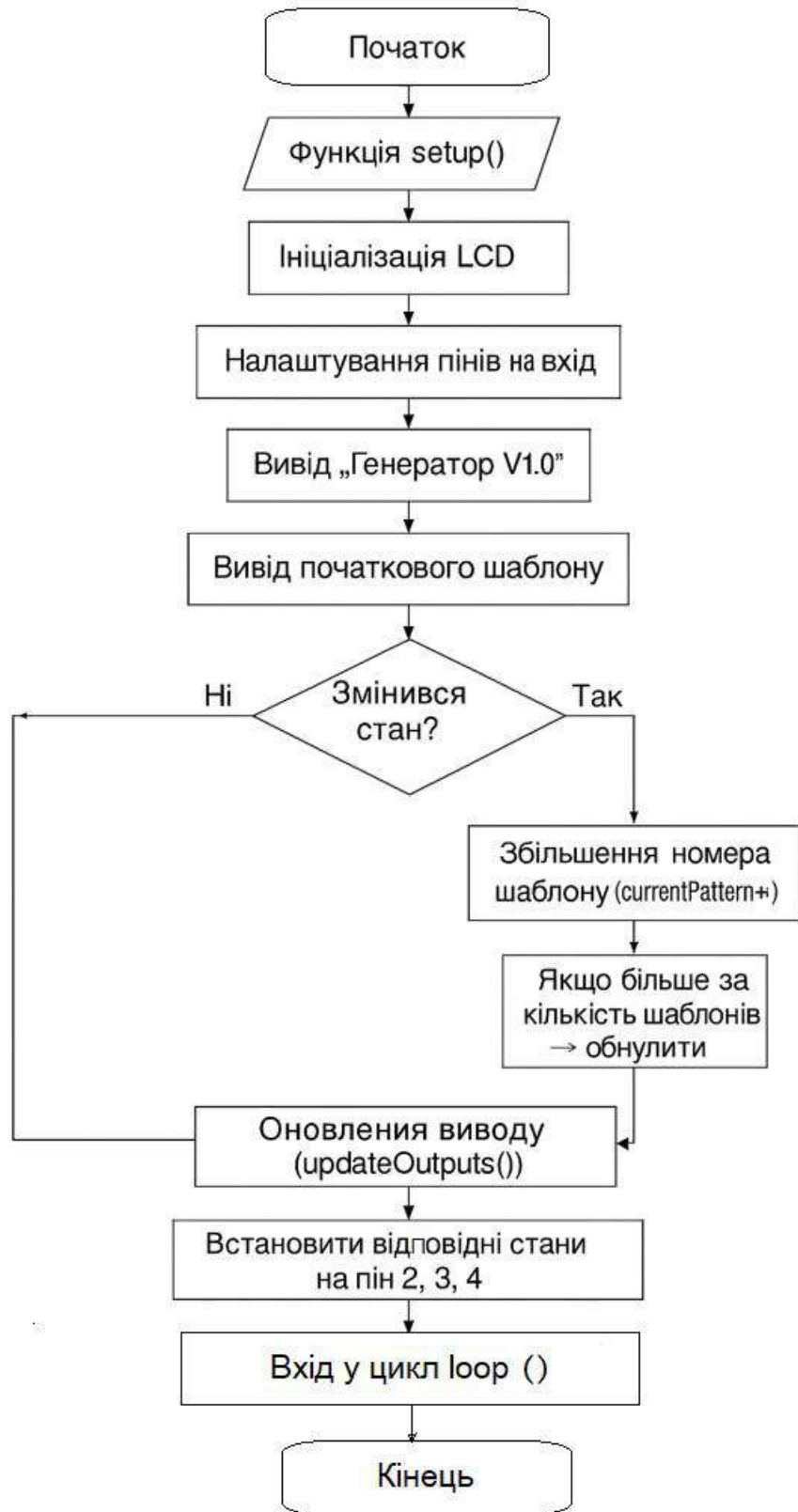


Рисунок 2.6 – Блок-схема алгоритму генератора шаблонів

У нашому випадку генератор шаблонів призначений для подачі послідовностей логічних рівнів (HIGH або LOW) на входи логічних мікросхем,

що дозволяє перевірити їхню працездатність. Це є ключовим елементом для тестування інтегральних схем типу AND, OR, NOT, NAND тощо.

Наприклад, шаблон {HIGH, LOW, LOW} означає, що перший пін (Pin 2) подає логічну «1», а інші (Pin 3 і 4) – логічний «0». Кожен шаблон дозволяє перевірити реакцію тестованої мікросхеми на певну вхідну комбінацію.

У проєкті реалізовані наступні шаблони:

1. {LOW, LOW, LOW} – всі піни низькі рівні.
2. {HIGH, LOW, LOW} – тільки перший пін високий.
3. {LOW, HIGH, LOW} – тільки другий пін високий.
4. {HIGH, HIGH, HIGH} – всі піни високі рівні.

У контексті розробки цифрових пристроїв, особливо таких, що керуються мікроконтролерами на зразок Arduino, діаграми станів є потужним інструментом для формалізації логіки переходів між різними режимами роботи.

Коли йдеться про генератор логічних шаблонів, кожен шаблон можна розглядати як окремий стан скінченного автомата. Такий підхід дозволяє структурувати керування шаблонами, візуалізувати послідовності перемикачів, та передбачити реакції системи на зовнішні події (наприклад, натискання кнопки).

Скінченний автомат (Finite State Machine, FSM) – це математична модель обчислювального пристрою, що перебуває в одному з наперед визначених станів і може переходити між ними у відповідь на зовнішні сигнали або події. Він описує систему, яка може перебувати лише в одному з обмеженого числа станів у будь-який момент часу та переходити з одного стану в інший у відповідь на певні події або сигнали. У програмуванні й електроніці FSM часто використовується як модель для керування поведінкою пристрою або процесу, який має дискретну послідовність подій.

У контексті генератора логічних шаблонів, реалізованого, наприклад, на мікроконтролері Arduino, скінченний автомат ідеально підходить для опису процесу перемикачів між заздалегідь заданими шаблонами логічних сигналів, що подаються на виводи. Кожен шаблон – це стан автомата, а перехід між шаблонами

відбувається у відповідь на зовнішній тригер (наприклад, натискання кнопки).

Таке представлення дозволяє формалізувати й автоматизувати поведінку системи.

Загальними перевагами використання FSM у проекті є наступні:

1. Простота в реалізації, тому що логіка переходів між станами легко програмується умовними конструкціями.
2. Зрозумілість та передбачуваність полягає в тому, що система завжди перебуває в одному визначеному стані, що полегшує діагностику.
3. Масштабованість тому що можна легко додати нові шаблони (тобто стани), не змінюючи основної архітектури.
4. Універсальність, оскільки FSM підходить для будь-яких послідовностей дій: від простих кнопок до складної логіки тестування.
5. Відповідність циклічному характеру, що означає що він ідеально підходить для задач, де після останнього стану потрібне повернення до початкового (циклічна модель).

У генераторі шаблонів станами є логічні комбінації на виводах (наприклад, LOW-LOW-LOW, HIGH-LOW-LOW тощо).

Вхідною подією є натискання кнопки, що викликає зміну стану.

Вихідною реакцією є зміна логічного рівня на пінах і оновлення інформації на дисплеї.

Початковий стан зазвичай — усі виводи LOW.

Фінальний стан або повернення в початковий реалізуються через модульне обчислення (наприклад, $(currentState + 1) \% numPatterns$).

FSM дозволяє створити надійну, реактивну й структуровану систему управління. Його особливо доцільно використовувати у проектах, де потрібно точно визначати, у якому режимі перебуває пристрій та як він повинен реагувати на події. У разі розширення функціональності генератора (наприклад, додавання таймерів, режимів тестування чи збереження шаблонів) FSM буде гнучким і ефективним каркасом для подальшої розробки.

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 39
Зм.	Арк.	№ докум.	Підпис	Дата		

Приклад діаграма станів шаблонів наведена на рисунку 2.7.

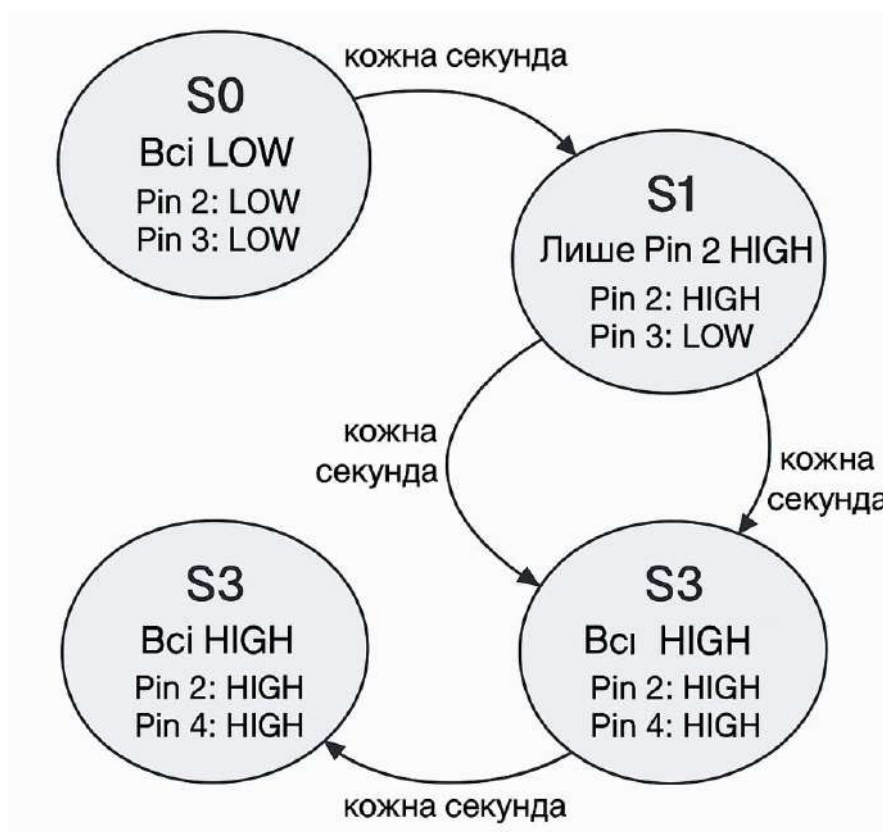


Рисунок 2.7 – Приклад діаграми станів шаблонів

У нашому випадку, автомат реалізований програмно всередині мікроконтролера Arduino, де кожен стан відповідає певному логічному шаблону на вихідних пінах.

Така модель потрібна для наступних дій:

1. Формалізація алгоритму роботи дозволяє чітко описати логіку перемикання шаблонів.
2. Модульність та масштабованість дає можливість легко додавати нові шаблони або розширювати логіку переходів.
3. Простота діагностики та налагодження є особливо важливим в електроніці, де видима візуалізація переходів полегшує відлагодження.
4. Можливість формальної перевірки FSM дозволяє аналізувати працездатність алгоритму ще до реалізації.

5. Ефективне використання ресурсів мікроконтролера означає, що переходи чітко контрольовані, немає зайвих умов.

У прикладі генератора шаблонів кожен шаблон є станом. Система починає з початкового стану, наприклад, усі пін-на виходи LOW (шаблон 0). Після натискання кнопки система переходить у наступний стан шаблон 1: лише один пін HIGH. І так далі. Після останнього шаблону автомат переходить знову у початковий стан (циклічна модель).

Стан 0: {LOW, LOW, LOW}.

Стан 1: {HIGH, LOW, LOW}.

Стан 2: {LOW, HIGH, LOW}.

Стан 3: {HIGH, HIGH, HIGH}, далі перехід знову до Стану 0.

Кожен перехід між станами ініціюється подією, тобто натисканням кнопки, що обробляється в коді Arduino з урахуванням антибрязкіткової логіки (фільтрації паразитних натискань). Така поведінка повністю відповідає класичній структурі детермінованого скінченного автомата.

2.4 Висновки до другого розділу

У другому розділі розглянуто метод подачі послідовних логічних шаблонів на вхід досліджуваних мікросхем. У цьому контексті описано підхід до реалізації таких шаблонів за допомогою мікроконтролера Arduino, що забезпечує простоту керування, програмованість та гнучкість. За допомогою блок-схем та діаграм станів, включаючи скінченний автомат (FSM), було формалізовано логіку роботи генератора шаблонів, що дозволяє легко масштабувати та змінювати поведінку системи.

Крім того, розглянуто методи перевірки цифрових IC, включно з використанням ZIF-сокетів, світлодіодної індикації, генераторів сигналів та подальшого візуального контролю. Пояснено доцільність використання Tinkercad Circuits як середовища моделювання для попереднього тестування логіки схеми, що дає змогу перевірити функціональність без фізичного збирання пристрою.

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата		

У результаті аналізу зроблено висновок, що побудова генератора логічних шаблонів на базі Arduino з можливістю послідовної подачі сигналів, зручного керування (через кнопку) та візуального відображення (через LCD-дисплей) є ефективним, доступним і наочним способом перевірки логічних мікросхем.

					КвРКІ. 22024.40.02.40 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ГЕНЕРАТОРА ВИПРОБУВАЛЬНИХ СИГНАЛІВ

3.1 Опис реалізації модулів апаратного та програмного забезпечення програмно-технічного засобу

Програмно-технічний засіб (ПТЗ), який було розроблено в межах цього проекту, виконує функцію генератора тестових логічних сигналів із можливістю ручного перемикавання між наперед заданими шаблонами. Основна задача такого пристрою – це надання зручного, простого у використанні способу для подачі на цифрові входи логічних мікросхем фіксованих комбінацій логічних рівнів (0 або 1) задля їх перевірки чи демонстрації роботи.

При цьому користувач має можливість натискати кнопку для циклічного перемикавання між шаблонами. Поточний шаблон виводиться на LCD-дисплей, а виходи (3 цифрові піни) змінюють свій логічний стан відповідно до обраної конфігурації. Таким чином, забезпечується просте і наочне управління логічними станами.

Для програмної реалізації була обрана платформа Tinkercad [40].

Tinkercad – це безкоштовна онлайн-платформа, розроблена компанією Autodesk, яка дозволяє створювати 3D-моделі, електронні схеми та симуляції програмованих мікроконтролерів, таких як Arduino.

Інтерфейс платформи інтуїтивно зрозумілий, що робить її особливо зручною для початківців, студентів, викладачів та всіх, хто тільки починає вивчати основи електроніки або 3D-моделювання. Однією з ключових функцій Tinkercad є редактор схем, який дозволяє створювати електронні проекти без необхідності фізично підключати компоненти. У середовищі можна зібрати схему з резисторів, діодів, кнопок, дисплеїв, Arduino та інших пристроїв, а також запрограмувати плату Arduino прямо у вікні браузера, використовуючи стандартну мову Arduino (C/C++).

Також Tinkercad підтримує симуляцію роботи мікроконтролера: після

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

запуску моделі користувач може спостерігати, як працює програма, наприклад, як блимає світлодіод, як зчитуються показники з сенсора, як працює сервомотор тощо. Це дає змогу тестувати логіку проекту без фізичної плати, що є ідеальним для навчання або попереднього моделювання.

Крім електроніки, Tinkercad має потужний 3D-редактор, який дозволяє створювати об'єкти для друку на 3D-принтерах. З допомогою простих форм (куб, сфера, призма тощо) користувач може створити будь-яку конструкцію, об'єднуючи їх, віднімаючи або перетворюючи. Платформа також має спільноту користувачів, яка ділиться проектами, схемами та інструкціями.

Завдяки своїй доступності, наочності й гнучкості Tinkercad є потужним інструментом для розробки, тестування та візуалізації проектів у сфері електроніки, автоматизації та дизайну. У поєднанні з фізичними Arduino-платами, він дозволяє значно пришвидшити цикл проектування та налагодження як для аматорів, так і для професіоналів.

Розглянемо тепер більш детально апаратне та програмне забезпечення проекту.

3.1.1 Апаратне забезпечення

Центральним обчислювальним модулем є ATmega328 мікроконтролер ATmega328P, який є складовою частиною широко використовуваної платформи Arduino Uno. ATmega328P – надійний і енергоефективний 8-бітний процесор від компанії Microchip (раніше Atmel). Він має 32 КБ Flash-пам'яті для зберігання програми, 2 КБ SRAM для оперативних обчислень та 1 КБ EEPROM для збереження даних між перезавантаженнями.

Мікроконтролер підтримує до 23 ліній вводу/виводу, 6 з яких можуть бути використані як аналогові входи, та має можливість керування широтно-імпульсною модуляцією (PWM). Його обчислювальні можливості та сумісність із великою кількістю бібліотек Arduino роблять ATmega328P ідеальним вибором для систем автоматизації, керування датчиками, двигунами та іншими пристроями.

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

Мікроконтролер, зокрема ATmega328P, є центральним елементом керування вбудованих систем і виконує низку важливих функцій, які забезпечують взаємодію з зовнішнім середовищем, обробку даних та прийняття рішень. Однією з ключових функцій є збір інформації з сенсорів, аналогових або цифрових, що дозволяє системі фіксувати параметри середовища (температура, вологість, освітленість тощо).

Далі мікроконтролер обробляє отримані дані відповідно до закладеної логіки програми. Це може включати математичні обчислення, порівняння, застосування правил «якщо–то», використання таблиць або алгоритмів машинного навчання (в обмеженому вигляді).

На основі результатів аналізу, мікроконтролер керує виконавчими пристроями, подаючи сигнали на реле, двигуни, світлодіоди, клапани чи інші механізми. Він також забезпечує зберігання і передачу даних, наприклад, у хмару через Wi-Fi-модуль або на SD-карту. Крім того, мікроконтролер може забезпечувати таймерні функції, наприклад, запуск певних дій за розкладом, і реагувати на зовнішні події (переривання), що дозволяє ефективно працювати з кнопками, сенсорами руху та іншими подіями в реальному часі.

Наступною складовою є вузол виводу логічних сигналів. Вузол виводу логічних сигналів відіграє важливу роль у візуалізації стану роботи мікроконтролера, особливо під час тестування, налагодження або демонстрації алгоритмів керування. У даній конфігурації до трьох цифрових пінів (2, 3, 4) мікроконтролера ATmega328P, який є частиною плати Arduino Uno, підключено світлодіоди (LED) через резистори обмеження струму номіналом 220 Ом. Ці резистори необхідні для захисту як світлодіодів, так і вихідних ліній мікроконтролера від надмірного струму, оскільки типова межа струму для одного піну становить не більше 40 мА. Резистор дозволяє зменшити струм через світлодіод до безпечного рівня (приблизно 10–20 мА), що цілком достатньо для яскравого світіння індикатора. У такій схемі кожен світлодіод виконує функцію індикатора, що відображає логічний стан (високий або низький рівень)

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 45
Зм.	Арк.	№ докум.	Підпис	Дата		

відповідного вихідного піну. Коли пін встановлюється в логічну «1» (високий рівень, ≈ 5 В), струм протікає через світлодіод, і він світиться. Якщо ж встановлено логічну «0» (≈ 0 В), світлодіод вимикається.

Це просте, але ефективне рішення дозволяє візуально контролювати процеси, які реалізуються в програмному коді, наприклад: чергування станів, спрацювання умов, активність алгоритмів тощо. Такий вузол особливо зручний для навчання, відлагодження систем керування, демонстрацій або діагностики несправностей у більш складних пристроях. В реальному застосуванні ці пінів можна підключати до входів тестованих мікросхем. Таким чином, ПТЗ виступає в ролі джерела цифрових сигналів.

Кнопка перемикання шаблонів у даній системі підключена до піну 5 мікроконтролера ATmega328P і слугує для циклічного перемикання між кількома режимами роботи або шаблонами керування. Вона реалізує функцію інтерфейсу користувача, дозволяючи змінювати логіку поведінки системи без необхідності перепрошивки або підключення до комп'ютера. Кнопка з'єднана одним контактом безпосередньо з GND (заземленням), а іншим – з цифровим входом мікроконтролера. Для спрощення схеми та уникнення необхідності в зовнішніх компонентах у коді використовується внутрішній підтягуючий резистор, який активується через параметр INPUT_PULLUP у функції pinMode(). Це означає, що в стані спокою на піні підтримується логічна «1» (≈ 5 В), а при натисканні кнопки, пін підтягується до нуля (GND), що реєструється як логічна «0».

Такий підхід дозволяє мінімізувати кількість зовнішніх елементів, що особливо важливо при обмеженому просторі на платі або для швидкого прототипування. Крім того, кнопка може використовуватися в поєднанні з дебаунсінгом у програмному коді, щоб уникнути помилкових спрацювань при механічних коливаннях контактів.

Циклічне перемикання, реалізоване за допомогою кнопки, зазвичай працює як лічильник режимів: кожне натискання змінює значення внутрішньої змінної, яка визначає, який із шаблонів активний у поточний момент. Це дозволяє

ефективно та інтуїтивно керувати багатофункціональною системою, використовуючи лише один фізичний елемент управління.

Рідкокристалічний LCD дисплей 16x2 з I2C модулем використовується для індикації назви пристрою та поточного активного шаблону. Завдяки інтерфейсу I2C вдається зекономити цифрові піни, оскільки використовується лише два: A4 (SDA) і A5 (SCL). Дисплей дозволяє наочно показати користувачеві номер активного шаблону (1, 2, 3, ...), а також у майбутньому – назви чи конфігурації сигналів. Дисплей відображає назву пристрою та номер поточного активного шаблону керування, що дозволяє користувачу візуально контролювати, в якому режимі працює система. Підключення дисплея здійснюється через інтерфейс I2C (інтер-інтегральний зв'язок), що дозволяє значно зменшити кількість необхідних цифрових пінів мікроконтролера. Замість стандартних чотирьох або більше проводів, що використовуються при паралельному з'єднанні, в I2C-версії використовується лише два контакти: A4 (SDA – лінія даних) і A5 (SCL – лінія синхронізації). Це особливо важливо в системах із великою кількістю підключених модулів.

LCD 16x2 дозволяє виводити до 16 символів у два рядки, що забезпечує достатньо місця для виводу інформації, такої як поточний шаблон, повідомлення про помилки, статуси системи чи інші параметри. Завдяки цьому елементу керування, користувач отримує наочний канал зворотного зв'язку, що підвищує ергономіку та зручність експлуатації системи. Крім того, такий дисплей дозволяє у майбутньому легко розширити функціонал, наприклад, додати відображення конфігурацій сигналів, значень параметрів середовища або інструкцій для обслуговування. Завдяки підтримці I2C дисплей легко інтегрується з Arduino та іншими мікроконтролерами через відповідні бібліотеки, що робить його доступним і ефективним рішенням для виводу даних у компактних керуючих системах.

Обрані компоненти апаратного забезпечення надають можливість надійного функціонування компонентів (таблиця 3.1).

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.1 – Компоненти апаратного забезпечення проєкту

Вид	Тип	Функції
Мікроконтролер	ATmega328P (у складі Arduino Uno)	1. Управління вихідними пінів (2, 3, 4). 2. Зчитування стану кнопки (пін 5). 3. Виведення інформації на LCD-дисплей (через інтерфейс I2C)
LCD-дисплей	16x2 LCD з модулем I2C Інтерфейс: I2C (SDA – A4, SCL – A5)	Відображення активного шаблону сигналу
Світлодіоди (LED)	3 (на виходи 2, 3, 4) Для захисту – резистори по 220–330 Ом для обмеження струму	Візуалізація логічних рівнів
Кнопка	Тактовна кнопка з підтягуванням до живлення (INPUT_PULLUP). З'єднання: один контакт до GND, інший до піну 5	Перемикання між шаблонами
Осцилограф	Опціонально в Tinkercad	Використовується для візуалізації змін логічного рівня на пінів 2, 3, 4

3.1.2 Програмне забезпечення

Програмне забезпечення даного проєкту розроблено на платформі Arduino, що базується на мові програмування C/C++.

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 48
Зм.	Арк.	№ докум.	Підпис	Дата		

Основна мета програми – забезпечити стабільну генерацію логічних сигналів на трьох вихідних лініях, з можливістю перемикання шаблонів за допомогою кнопки.

На початку програми здійснюється ініціалізація LCD-дисплея з I2C-інтерфейсом, що дозволяє економити цифрові порти контролера. Після ініціалізації на дисплеї виводиться стартове повідомлення, що містить назву пристрою та версію прошивки.

Далі у функції `setup()` налаштовуються три виводи мікроконтролера як виходи, на які подається логічна інформація. Кнопка налаштовується як вхід з внутрішнім підтягуючим резистором, що дозволяє обійтись без зовнішніх компонентів.

Основна логіка програми реалізується у функції `loop()`, яка постійно виконується. В ній відбувається опитування кнопки, застосовується програмна антидребезгова фільтрація, а також обробка натискання.

У разі виявлення дійсного натискання кнопки, програма переходить до наступного шаблону логічних сигналів. Шаблони зберігаються у двовимірному масиві `patterns`, де кожен рядок визначає комбінацію HIGH/LOW значень для трьох виходів. Після зміни шаблону викликається функція `updateOutputs()`, яка оновлює сигнали на виходах та змінює повідомлення на LCD-дисплеї.

Вивід шаблону на дисплей дозволяє користувачеві легко орієнтуватися, який із шаблонів активний. Усі шаблони перемикаються циклічно, тобто після останнього шаблону програма повертається до першого. Для покращення зручності використання всі шаблони пронумеровані, а нумерація на дисплеї починається з 1, хоча індексація масиву починається з нуля.

Програма є стабільною, простою у розширенні та зручною для модифікацій. Наприклад, додавання нових шаблонів сигналів зводиться до розширення масиву `patterns` та зміни значення `numPatterns`. Також можливе розширення функціоналу: реалізація таймера автоматичного перемикання, збереження останнього шаблону в EEPROM, або організація меню для вибору режимів.

У програмному кодi не використовується функція `delay()`, що дозволяє уникнути блокувань та забезпечити високу чутливість до натискань кнопки.

Завдяки використанню бібліотеки `LiquidCrystal_I2C` дисплей працює без затримок та flicker-ефектів, оскільки оновлюється лише при зміні шаблону.

Загалом програмне забезпечення організоване так, щоб забезпечити максимальну наочність, простоту керування і надійність роботи. Структура коду добре читається, що дозволяє легко адаптувати або вдосконалити програму під потреби користувача або лабораторних експериментів. Крім того, логічні рівні на виходах можна подавати як на світлодіоди для візуального контролю, так і безпосередньо на входи цифрових мікросхем. Це забезпечує зручність застосування генератора у процесі налагодження або тестування електронних схем.

3.2 Опис процесу створення програмованого генератора випробувальних сигналів

Для моделювання та попереднього тестування програмованого генератора випробувальних сигналів було використано онлайн-середовище `Tinkercad Circuits`, яке дозволяє в інтерактивному режимі створювати електронні схеми з мікроконтролерами `Arduino`, а також виконувати симуляцію роботи написаного програмного забезпечення. Середовище підтримує як візуальне складання схем, так і програмування мовою `C++` (`Arduino IDE`-стиль), що робить його зручним для попередньої перевірки логіки роботи пристрою.

У середовищі `Tinkercad` було створено новий електронний проєкт (схему), до якого додано наступні основні компоненти:

- мікроконтролер `Arduino Uno R3` як головний керуючий елемент системи (рисунок 3.1);
- три світлодіоди різного кольору (синій, помаранчевий, зелений), які візуалізують логічні стани на виходах (рисунок 3.2);
- резистори номіналом 220 Ом, підключені послідовно зі світлодіодами

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

для обмеження струму;

- LCD-дисплей 16x2 з I2C інтерфейсом для індикації активного шаблону (рисунок 3.3);
- кнопка для ручного перемикання шаблонів у тестовому варіанті (Опціонально);
- провідники, що забезпечують з'єднання між елементами згідно з логічною схемою.

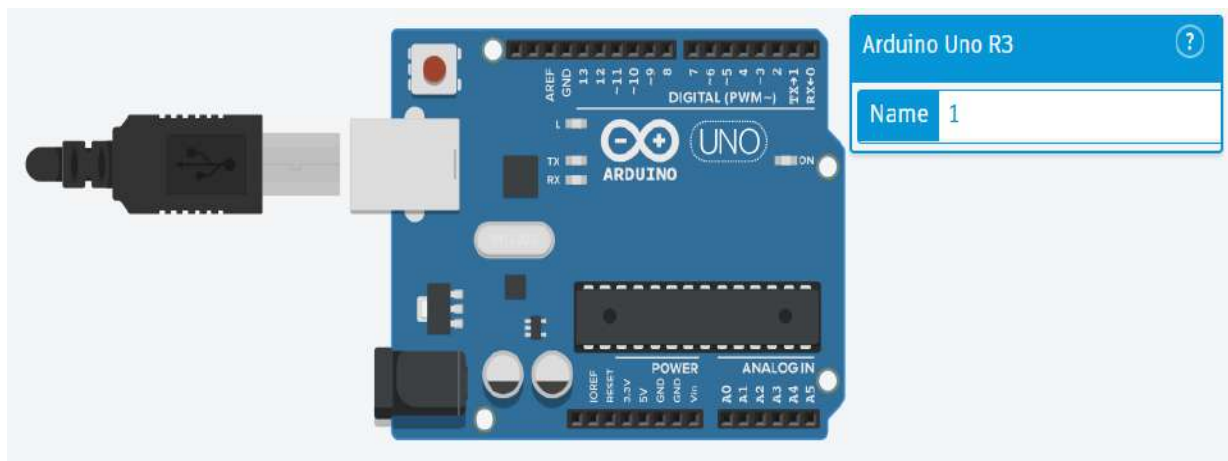


Рисунок 3.1 – Мікроконтролер Arduino Uno R3

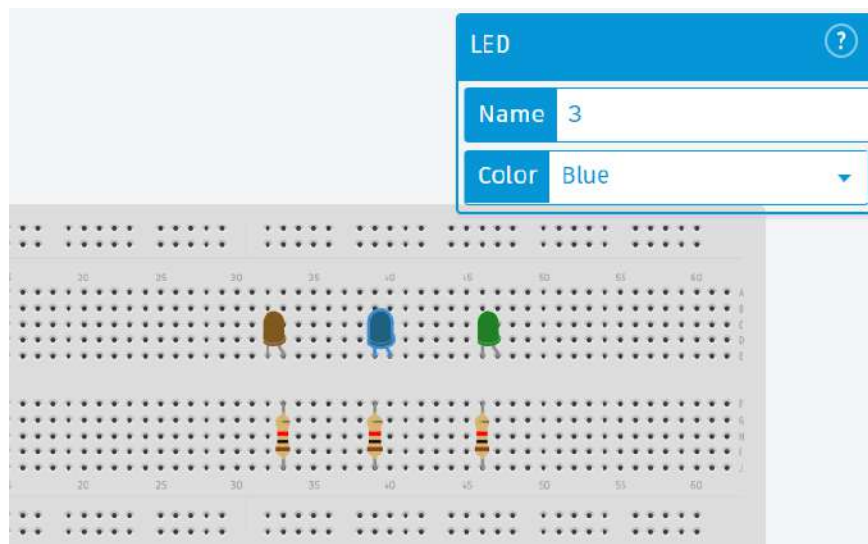


Рисунок 3.2 – Три світлодіоди різного кольору

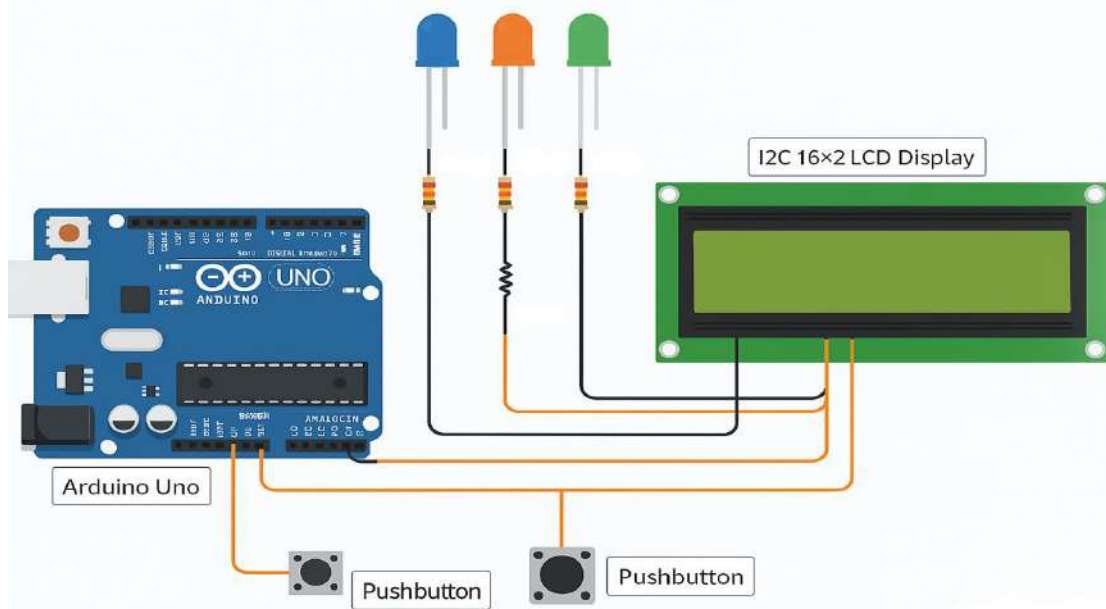


Рисунок 3.4 – Приклад підключення

Кнопка перемикання шаблонів, підключена до цифрового входу з внутрішнім підтягуючим резистором, дозволяє користувачу циклічно обирати один із попередньо запрограмованих шаблонів логічних сигналів, які автоматично відтворюються на вихідних пінів мікроконтролера. Це забезпечує гнучке керування режимами тестування без необхідності змінювати прошивку чи підключення.

На виході мікроконтролера підключені світлодіоди з резисторами, які відображають логічний стан відповідних ліній, дозволяючи користувачеві бачити, які сигнали подаються на мікросхему. Це спрощує діагностику, адже всі сигнали видно в реальному часі.

Програмне забезпечення написано в текстовому режимі редактора коду Tinkercad. Воно реалізує послідовне виведення наперед визначених логічних шаблонів на три цифрові виходи мікроконтролера.

Шаблони задаються як масив двійкових комбінацій, наприклад:

- {LOW, LOW, LOW};
- {HIGH, LOW, LOW};

- {LOW, HIGH, LOW};
- {HIGH, HIGH, HIGH}.

Кожна така комбінація відповідає одному з етапів циклу випробувальних сигналів. Логіка програми побудована так, що шаблони перемикаються автоматично з інтервалом 1 секунда. Поточний шаблон виводиться на LCD-дисплей з номером (наприклад, “Шаблон: 3”).

Після завершення схеми та завантаження коду, було запущено симуляцію (команда Start Simulation). У вікні моделювання в режимі реального часу можна спостерігати наступне:

- як по черзі вмикаються різні світлодіоди відповідно до шаблонів;
- як на LCD-дисплеї відображається відповідний номер шаблону;
- якщо використовувалась кнопка — перевірити її реакцію на натискання.

Таким чином, функціональність генератора була повністю змодельована в безпечному та зручному середовищі, без потреби у фізичному обладнанні.

Результат представлений на рисунку 3.5.

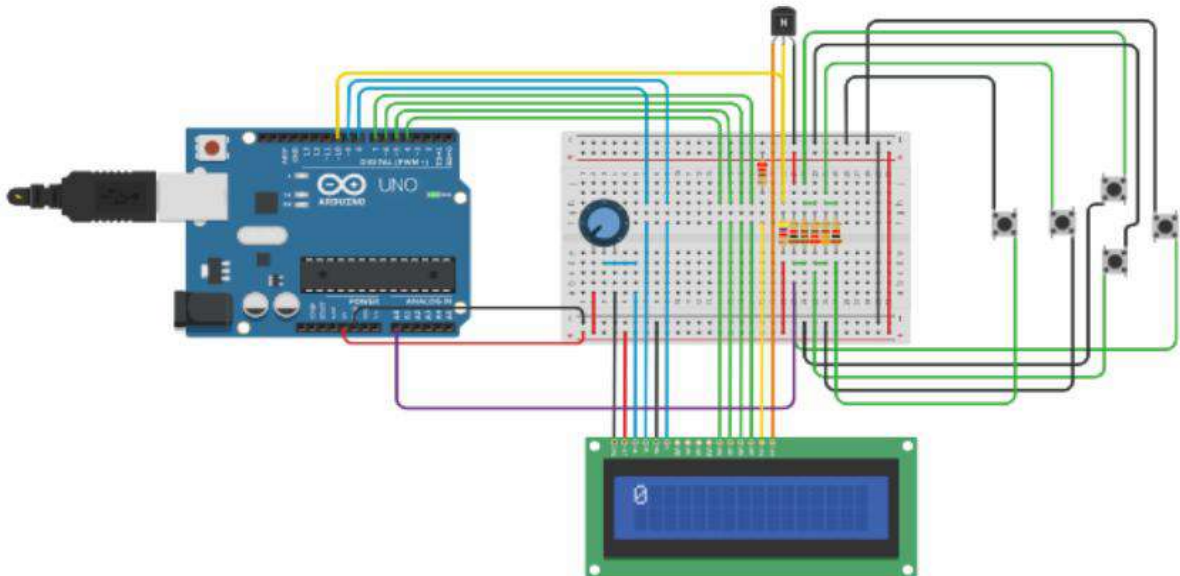


Рисунок 3.5 –Результат виконання команди Start Simulation

Прицьому схема електрична принципова має вигляд 3.6.

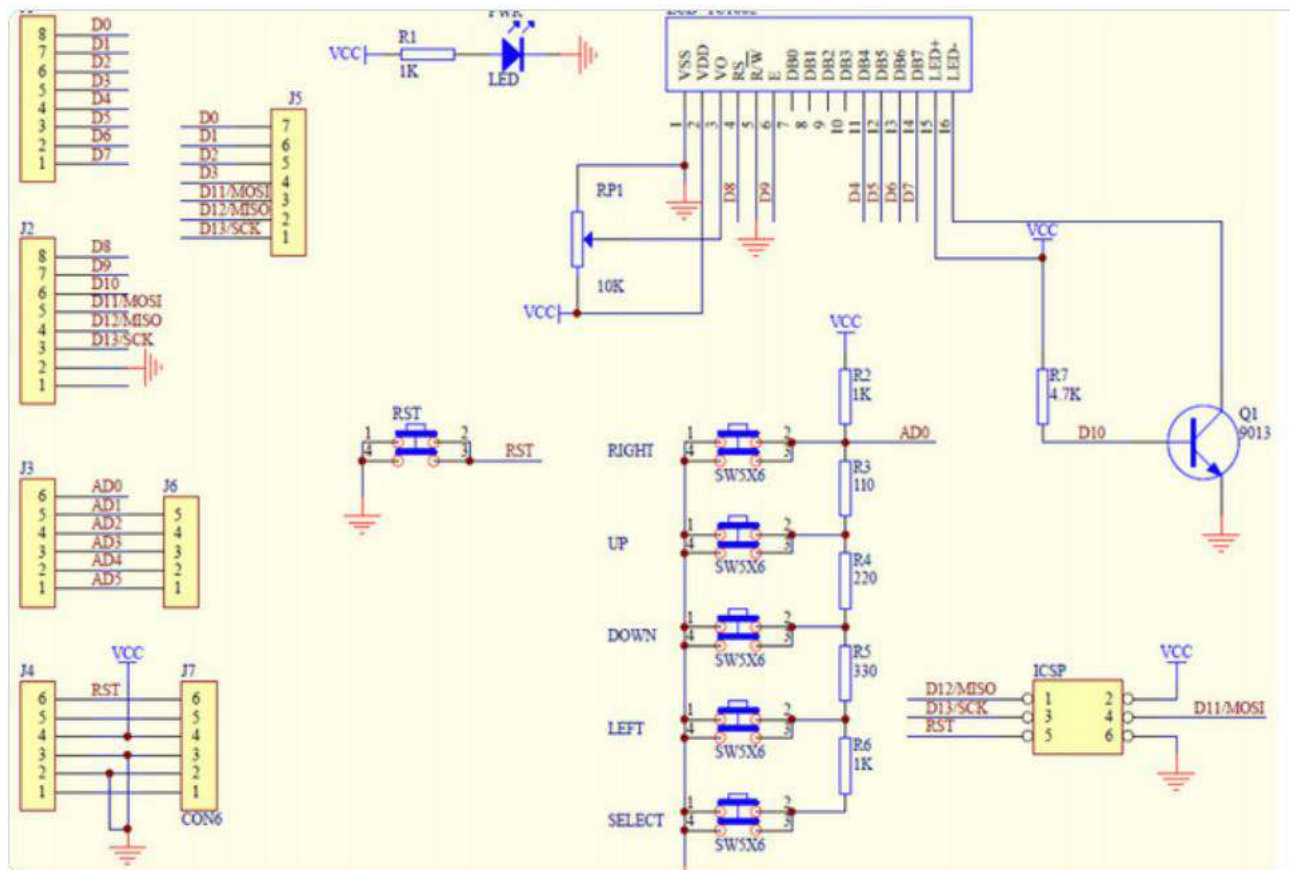


Рисунок 3.6 – Схема електрична принципова

Електрична принципова схема програмованого генератора випробувальних сигналів на базі мікроконтролера ATmega328 включає основні функціональні блоки живлення, управління та генерації сигналів. Живлення забезпечується стабілізованим джерелом 5 В, яке підключене до відповідних ніжок VCC і GND мікроконтролера.

Для забезпечення стабільної роботи мікроконтролера передбачено кварцовий резонатор на 16 МГц з двома конденсаторами по 22 пФ, підключеними до ніжок XTAL1 і XTAL2. Виводи мікроконтролера PORTD і PORTB використовуються для генерації цифрових логічних сигналів. До мікроконтролера підключено кнопки керування, з'єднані з входами PORTC через резистори підтягування. Для відображення поточного режиму роботи до портів мікроконтролера підключено LCD-дисплей з інтерфейсом I2C або паралельним з'єднанням. На виході генератора передбачено буферні елементи (наприклад,

74НС245) для захисту мікроконтролера та узгодження з логічними мікросхемами. Кожен вихід сигналу може мати світлодіод з обмежувальним резистором для індикації активності. Живлення та сигнали розведені на роз'єм для зручного підключення до мікросхем, що тестуються.

Графік перемикання шаблонів сигналів на виводах відображає часову залежність станів цифрових виходів мікроконтролера. У даному проєкті шаблони змінюються з фіксованим часовим інтервалом, наприклад, кожену секунду.

Графік перемикання шаблонів сигналів на виводах представлений на рисунку 3.6.



Рисунок 3.6 – Графік перемикання шаблонів сигналів на виводах

Кожен шаблон задає конкретну комбінацію логічних рівнів (HIGH або LOW) на вихідних пінових лініях, що з'єднані зі світлодіодами або входами логічної мікросхеми. Графік дозволяє наочно побачити, коли кожен пін (наприклад, пін 2, 3 або 4) змінює свій стан (рисунок 3.7 – 3.10).

Це є корисним для верифікації правильності генерації сигналів та синхронізації з тестованими схемами. Кожна наступна комбінація на графіку відповідає новому шаблону, який формується у певній послідовності.

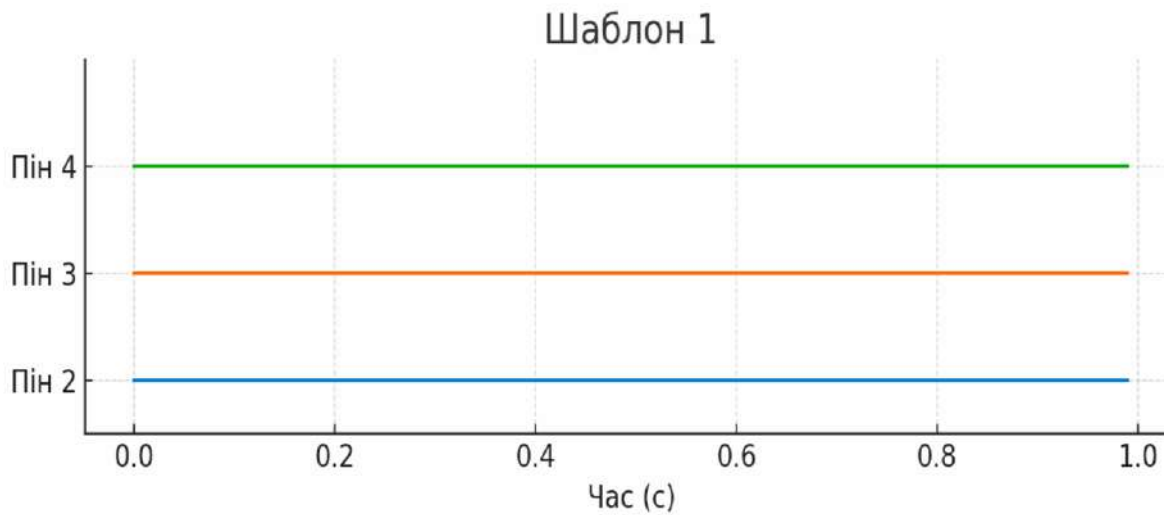


Рисунок 3.7 – Графік перемикання шаблону сигналів (шаблон 1)

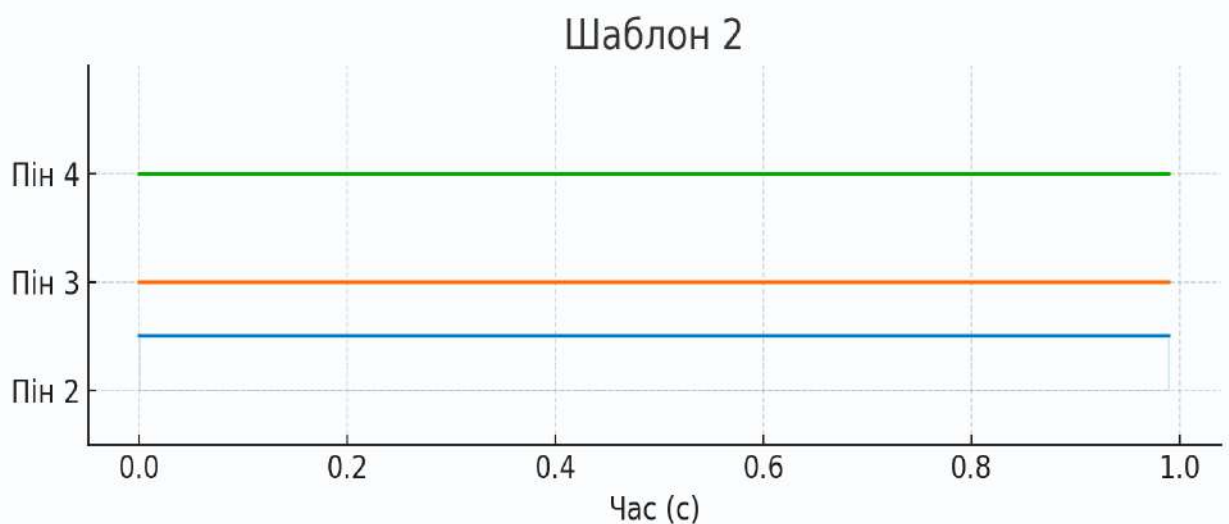


Рисунок 3.8 – Графік перемикання шаблону сигналів (шаблон 2)

Візуалізація такого графіка полегшує аналіз динаміки зміни станів на пінових виходах. У навчальних цілях він допомагає зрозуміти, як працює цифровий сигнал у часі. Для розробників – це інструмент відлагодження правильного таймінгу. У результаті, графік служить ключовим елементом документації цифрової системи контролю шаблонів. Такий графік є корисним під час моделювання або відлагодження схеми, оскільки дозволяє точно визначити момент переходу між станами. Він також допомагає перевірити відповідність

генератора логічних шаблонів очікуваним таблицям істинності для тестування цифрових мікросхем.

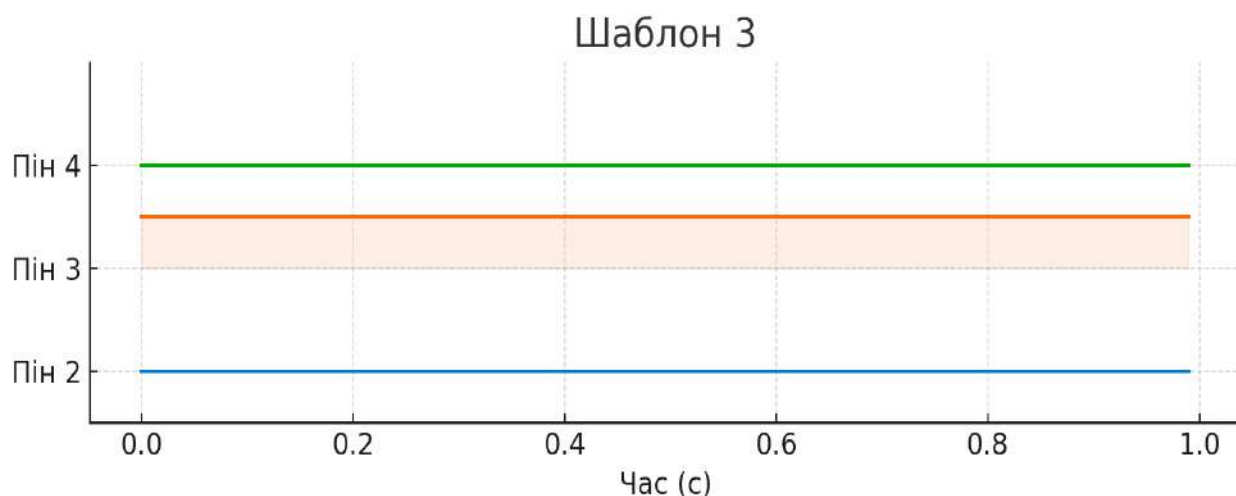


Рисунок 3.9 – Графік перемикання шаблону сигналів (шаблон 3)

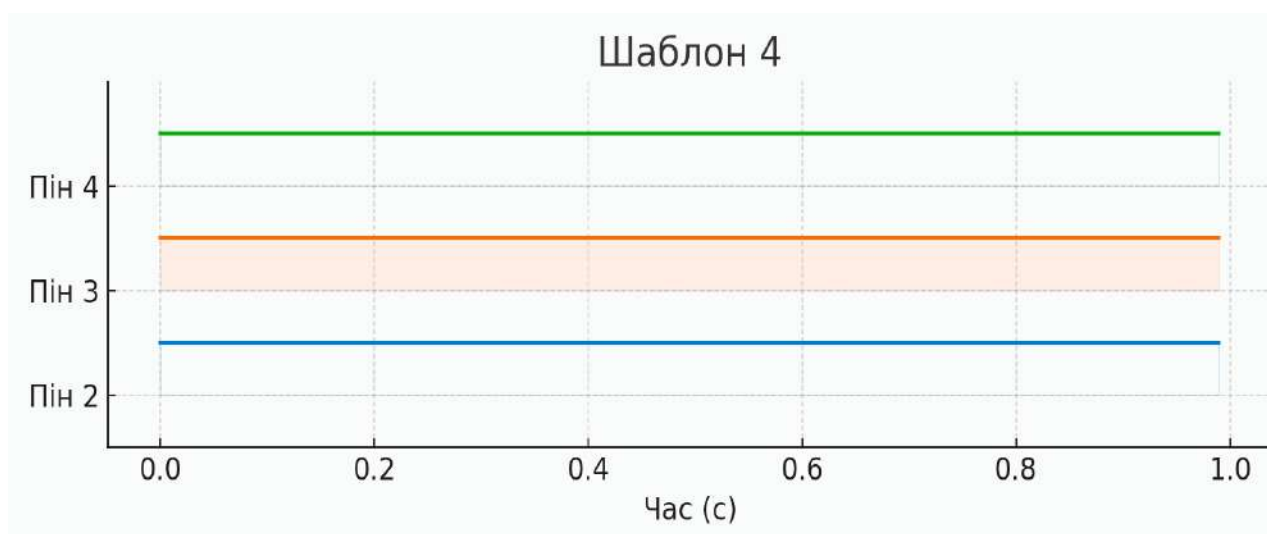


Рисунок 3.10 – Графік перемикання шаблону сигналів (шаблон 4)

Розглянемо, наприклад, графік роботи генератора шаблонів для перевірки логічного елемента AND.

Уявімо, що генератор змінює шаблон кожену секунду, і на пін 2 та пін 3 Arduino подаються сигнали, які з'єднані з входами A та B елемента AND. Вихід з

конденсатори, транзистори, логічні мікросхеми, дисплеї, мотори, світлодіоди, перемикачі, джерела живлення та багато іншого. Компоненти легко перетягувати на плату, з'єднувати між собою, задавати параметри й змінювати в процесі симуляції.

Ще одна перевага – це візуалізація процесів у схемі. Користувач може спостерігати за рівнями напруги, станами портів Arduino, струмами в колах тощо. Така візуалізація особливо важлива для навчання, адже вона дозволяє краще зрозуміти, як працює електроніка.

Tinkercad Circuits підтримує інтерактивне навчання. Багато онлайн-курсів і підручників використовують цю платформу для практики. Користувачі можуть створювати уроки, крок за кроком пояснювати принципи роботи схем і демонструвати результати прямо у вікні браузера.

Крім того, у платформі реалізована функція спільної роботи, можна поділитися проєктом через посилання або запросити інших користувачів до спільного редагування, що дуже зручно для командної роботи або консультацій з викладачами.

Оскільки всі дії виконуються у віртуальному середовищі, користувачі мають можливість експериментувати без ризику пошкодження обладнання. Це особливо важливо для новачків, які ще не впевнено працюють з реальними схемами.

Нарешті, варто зазначити, що Tinkercad Circuits є повністю безкоштовним.

Попри велику кількість переваг, Tinkercad Circuits має низку обмежень, які варто враховувати при виборі цього середовища для розробки електронних схем.

У порівнянні з професійними системами моделювання (наприклад, Proteus, Multisim, LTspice), Tinkercad має досить скромний набір електронних елементів. В ньому відсутні багато сучасних мікросхем, драйверів, сенсорів, комунікаційних модулів (наприклад, SPI/I2C розширення, модулі Bluetooth або Wi-Fi), що обмежує складність проєктів, які можна змоделювати.

У симуляторі не враховуються багато реальних ефектів, таких як паразитна

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

ємність, індуктивність, втрати на нагрівання, швидкість фронтів сигналів, шум тощо. Це може призвести до ситуацій, коли схема «працює» в Tinkercad, але в реальному житті ні.

У симуляції Tinkercad немає повноцінного осцилографа, мультиметра або аналізатора логічних рівнів, як це є в інших системах. Це обмежує можливості точного аналізу динамічних процесів у схемі.

Tinkercad Circuits не підходить для моделювання аналогових схем середньої та високої складності, зокрема підсилювачів, фільтрів, джерел живлення з ШІМ-керуванням, чи імпульсних перетворювачів.

Хоча середовище підтримує Arduino, воно дозволяє використовувати лише базову мову C++ і не підтримує повного спектру бібліотек Arduino IDE. Також недоступні розширені можливості, як-от використання таймерів у режимі CTC, переривань на низькому рівні або роботи з регістрами напряму.

При зростанні складності схеми, збільшенні кількості з'єднань та активних компонентів, система починає працювати повільніше, або взагалі не запускає симуляцію. Це суттєво обмежує розмір проектів.

Хоча можливе спільне редагування, платформа не має таких функцій як контроль версій, коментарі до змін, історія проекту, що є критично важливим у командній розробці складних пристроїв.

Tinkercad більше орієнтований на початківців, тому офіційна документація і підтримка не охоплює глибоких технічних питань. Це створює труднощі для досвідчених розробників, які хочуть реалізувати щось складніше, ніж базові приклади.

Окрім Arduino Uno (ATmega328P), інші типи мікроконтролерів (STM32, ESP32, PIC тощо) недоступні, що не дозволяє використовувати платформу для навчання чи моделювання сучасних embedded-систем.

Відсутність експорту у вигляді схем та друкованих плат. Tinkercad не дозволяє експортувати схеми у форматах, сумісних з програмами типу Eagle, KiCad чи Altium Designer. Це унеможливорює подальше перенесення проекту в

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

середовище для створення друкованих плат без повного перемальовування.

3.3 Висновки до третього розділу

У результаті проведеної симуляції програмованого генератора випробувальних сигналів на базі мікроконтролера ATmega328 було успішно змодельовано функціонування всіх основних режимів роботи пристрою. Аналіз отриманих графіків логічних рівнів підтвердив коректність генерації сигналів заданої частоти, тривалості імпульсів і шаблонів, відповідно до написаної прошивки. Графічні результати, отримані під час симуляції, наочно продемонстрували узгодженість між апаратною частиною (формування логічних сигналів на виходах) та програмною логікою перемикачів режимів.

Під час проектування було обґрунтовано вибір мікроконтролера ATmega328 як недорогого, енергоефективного та сумісного з Arduino-середовищем програмування. В якості апаратного забезпечення обрано базові компоненти, такі як кнопки, LCD-дисплей, буферні логічні елементи та стабілізоване джерело живлення. Це забезпечує простоту відтворення схеми як на макетній платі, так і на друкованій платі.

У ролі середовища розробки використано Arduino IDE, яке завдяки своїй простоті та наявності бібліотек дозволило швидко реалізувати функціонал перемикачів шаблонів, генерації сигналів та відображення інформації. Для симуляції проекту було обрано платформу Tinkercad Circuits, яка дозволила без фізичних компонентів протестувати поведінку системи, перевірити роботу програми та візуалізувати логіку генерації сигналів.

					КвРКІ. 22024.40.02.40 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень було розроблено програмований генератор випробувальних сигналів на базі мікроконтролера ATmega328, що є універсальним інструментом для перевірки справності логічних мікросхем, що дозволяє подавати на їх входи типові та користувацькі сигнали з різною частотою, послідовністю та логікою, що значно спрощує процес діагностики.

У межах розділу 1 проведено огляд мікроконтролерної платформи Arduino та основних логічних вентилів. В результаті проведеного огляду, доведено, що Arduino є доступним, гнучким і функціональним інструментом для швидкої розробки та тестування електронних систем, що дає змогу створювати як навчальні, так і промислові прототипи. Платформа має широку апаратну підтримку, велику кількість бібліотек і активну спільноту, що спрощує роботу навіть початківцям. Особливе значення має її здатність керувати виводами для подачі або зчитування логічних сигналів, що робить її ідеальною для роботи з логічними мікросхемами. Водночас логічні вентиля виконують базові функції обробки двійкової інформації, які лежать в основі роботи будь-якої цифрової схеми. Зокрема, вентиля AND, OR, NOT, NAND, NOR, XOR та XNOR дозволяють реалізовувати логічні операції, необхідні для побудови арифметичних пристроїв, контролерів, тригерів та логічних процесорів. Їх комбінації створюють складні схеми, що забезпечують обробку, зберігання та передачу інформації.

Проведений огляд показав, що Arduino дозволяє не лише керувати логічними рівнями, але й виводити інформацію на дисплеї, взаємодіяти з кнопками, сенсорами та іншими пристроями. Крім того, Arduino дозволяє моделювати та перевіряти різні шаблони подачі сигналів, що є корисним для аналізу поведінки логічних мікросхем у реальному часі.

У другому розділі розглянуто метод подачі послідовних логічних шаблонів на вхід досліджуваних мікросхем. У цьому контексті описано підхід до реалізації

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

таких шаблонів за допомогою мікроконтролера Arduino, що забезпечує простоту керування, програмованість та гнучкість. За допомогою блок-схем та діаграм станів, включаючи скінченний автомат (FSM), було формалізовано логіку роботи генератора шаблонів, що дозволяє легко масштабувати та змінювати поведінку системи.

Крім того, розглянуто методи перевірки цифрових ІС, включно з використанням ZIF-сокетів, світлодіодної індикації, генераторів сигналів та подальшого візуального контролю. Пояснено доцільність використання Tinkercad Circuits як середовища моделювання для попереднього тестування логіки схеми, що дає змогу перевірити функціональність без фізичного збирання пристрою.

У результаті аналізу зроблено висновок, що побудова генератора логічних шаблонів на базі Arduino з можливістю послідовної подачі сигналів, зручного керування (через кнопку) та візуального відображення (через LCD-дисплей) є ефективним, доступним і наочним способом перевірки логічних мікросхем.

У третьому розділі у результаті проведеної симуляції програмованого генератора випробувальних сигналів на базі мікроконтролера ATmega328 було успішно змодельовано функціонування всіх основних режимів роботи пристрою. Аналіз отриманих графіків логічних рівнів підтвердив коректність генерації сигналів заданої частоти, тривалості імпульсів і шаблонів, відповідно до написаної прошивки. Графічні результати, отримані під час симуляції, наочно продемонстрували узгодженість між апаратною частиною (формування логічних сигналів на виходах) та програмною логікою перемикання режимів.

Під час проектування було обґрунтовано вибір мікроконтролера ATmega328 як недорогого, енергоефективного та сумісного з Arduino-середовищем програмування. В якості апаратного забезпечення обрано базові компоненти, такі як кнопки, LCD-дисплей, буферні логічні елементи та стабілізоване джерело живлення. Це забезпечує простоту відтворення схеми як на макетній платі, так і на друкованій платі.

У ролі середовища розробки використано Arduino IDE, яке завдяки своїй

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 64
Зм.	Арк.	№ докум.	Підпис	Дата		

простоті та наявності бібліотек дозволило швидко реалізувати функціонал перемикачів шаблонів, генерації сигналів та відображення інформації. Для симуляції проєкту було обрано платформу Tinkercad Circuits, яка дозволила без фізичних компонентів протестувати поведінку системи, перевірити роботу програми та візуалізувати логіку генерації сигналів.

					КвРКІ. 22024.40.02.40 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Мікропроцесорна техніка : навч. посіб. / Цирульник С. М., Азаров О. Д., Крупельницький Л. В., Трояновська Т. І. Вінниця : ВНТУ, 2017. 123 с.
2. Мікроконтролер Ардуіно. URL: https://bitkit.com.ua/shho-take-arduino?srsltid=AfmBOoqSd0D_vbqХуА4уFv9DgUPNjQngMC42ooAeQYD4PeOQaz4C-IiZ (дата звернення: 12.04.2025).
3. Ismailov A. S., Jo'Rayev Z. B. Study of arduino microcontroller board. *Science and Education*. 2022. Vol. 3(3). Pp.172-179.
4. Martinez-Santos J. C., Acevedo-Patino O., Contreras-Ortiz S. H. Influence of arduino on the development of advanced microcontrollers courses. *IEEE revista iberoamericana de tecnologias del aprendizaje*. 2017. Vol. 12(4). Pp.208-217.
5. Yusro M., Guntor, N. A., Rikawarastuti. (2021, April). Utilization of microcontroller technology using Arduino board for internet of things (A systematic review). In *AIP conference proceedings*. 2021. Vol. 2331, No. 1, p. 060004.
6. Контролери Arduino. URL: <https://docs.arduino.cc/hardware/nano/> (дата звернення: 12.04.2025).
7. LilyPad Arduino ATmega328. URL: <https://www.mini-tech.com.ua/ua/lilypad-arduino> (дата звернення: 14.04.2025).
8. Arduino Ethernet. URL: <https://docs.arduino.cc/libraries/ethernet/> (дата звернення: 14.04.2025).
9. Nasir N. A. A. M., Mohtar M. N., Yunus N. A. M. (2018, November). Development of signal generator for lab on a chip application. In *2018 IEEE 5th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*. 2018. Pp. 1-4.
10. Kaswan K. S., Singh S. P., Sagar S. Role of Arduino in real world applications. *Int. J. Sci. Technol. Res*, 2020. Vol. 9(1). Pp. 1113-1116.
11. Jain P., Kumar P., Palwalia D. K. (2017, April). Irrigation management system with micro-controller application. In *2017 1st International Conference on*

					КВРКІ. 22024.40.02.40 ПЗ	Арк. 66
Зм.	Арк.	№ докум.	Підпис	Дата		

Electronics, Materials Engineering and Nano-Technology (IEMENTech). 2017. Pp. 1-6.

12. Gota D. I., Puscasiu A., Fanca A., Miclea L., Valean, H. (2020, May). Smart home automation system using Arduino microcontrollers. In *2020 IEEE International conference on automation, quality and testing, robotics (AQTR)*. 2020. Pp. 1-7.

13. Kim S. M., Choi Y., Suh J. Applications of the open-source hardware Arduino platform in the mining industry: A review. *Applied Sciences*. 2020. Vol. 10(14), 5018.

14. Hudedmani M. G., Swati I. K., Sneha S. G., Basamma B., Vivek M. N. Digital Logic Gate Simulation using Arduino Microcontroller. *Journal of Control System and Control Instrumentation*. 2017. Vol. 3(1). Pp.1-6.

15. Khaing S. W., Nopparatjamjomras S., Nopparatjamjomras T. R., Chitaree R. (2018, December). Development of Arduino-based logic gate training kit. In *Journal of Physics: Conference Series*. 2018. Vol. 1144. No. 1, p. 012134.

16. Lu Z., Jhoja J., Klein J., Wang X., Liu A., Flueckiger J., Chrostowski L. Performance prediction for silicon photonics integrated circuits with layout-dependent correlated manufacturing variability. *Optics express*. 2017. Vol. 25(9). Pp.9712-9733.

17. Da Silva M., Flottes M. L., Di Natale G., Rouzeyre B. Preventing scan attacks on secure circuits through scan chain encryption. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2018. Vol. 38(3). Pp. 538-550.

18. Aryan P., Sampath S., Sohn H. An overview of non-destructive testing methods for integrated circuit packaging inspection. *Sensors*. 2018. Vol. 18(7). P.1981.

19. Owen-Bellini M., Hacke P., Miller D. C., Kempe M. D., Spataru S., Tanahashi T., Topič, M. Advancing reliability assessments of photovoltaic modules and materials using combined-accelerated stress testing. *Progress in Photovoltaics: Research and Applications*. 2021. Vol. 29(1). Pp.64-82.

20. Elmelekawi A., Tammam A., Issa H. (2018, October). Unclonable key Generator Based on Chip signature and SRAM-PUF of ATmega328P chip. In *2018 28th International Conference on Computer Theory and Applications (ICCTA)*. 2018. Pp. 24-29).

					КВРКІ. 22024.40.02.40 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

21. Owolabi A. O., Oke A. O., Obalalu A. M., Khan U., Prasad B., Bajaj M. Development of an Energy-Efficient Digital Trainer for Logic Design Education Using ATmega328P Microcontroller. In *E3S Web of Conferences*. 2024. Vol. 591. P. 08001.

22. Nguimfack-Ndongmo J. D. D., Kentsa Zana K., Asoh D. A., Kengnou Telem N. A., Kuate-Fochie R., Kenné G. Development of a Simplified Programming Kit Based 16LF18856 for Embedded Systems Testing and Education in Developing Countries. *Journal of Electronic Testing*, 2022. Vol.38(6). Pp.589-602.

23. Kunikowski W., Czerwiński E., Olejnik P., Awrejcewicz J. An overview of ATmega AVR microcontrollers used in scientific research and industrial applications. *Pomiary Automatyka Robotyka*. 2015. Vol. 19(1). Pp. 15-19.

24. Salako E. A. A Time Switching System Using Atmega328 Microcontroller towards Solving Problem of Electrical Power Wastage. *International Journal of Image, Graphics and Signal Processing*. 2019. Vol. 10(6). P. 18.

25. Carminati M., Scandurra G. Impact and trends in embedding field programmable gate arrays and microcontrollers in scientific instrumentation. *Review of Scientific Instruments*. 2021. Vol. 92(9).

26. 28-pin ZIF socket. URL: <https://raspberrypi.com/ua/p/28-pin-zif-socket/> (дата звернення: 23.04.2025).

27. Чесановський І. І., Левчунець Д. О. Демодуляція сигналів в системах стандарту TETRA застосовуючи вейвлет перетворення. *Міжнародна науково-технічна конференція «Радіотехнічні поля, сигнали, апарати та системи» : матеріали конференції 14–20 березня 2016 р., м. Київ, Україна / КПІ ім. Ігоря Сікорського, РТФ*. 2016. С. 156–158.

28. Мікроконтролер ATmega328. URL: <https://diylab.com.ua/p64493856-mikrokontroler-atmega328.html> (дата звернення: 23.04.2025).

29. Мікроконтролер ATMEGA328P-PU (з завантажувачем Arduino Uno). URL: https://arduino.ua/prod15-mikrokontroller-atmega328p-pu?srsId=Afm_BOop2PgPGarUHb23612L_eLkdD3QAs1hfZ9qVFuvzALLUgbbjjLaQ (дата звернення: 26.04.2025).

					КвРКІ. 22024.40.02.40 ПЗ	Арк. 68
Зм.	Арк.	№ докум.	Підпис	Дата		

30. Sánchez L. G., Civera J. I., Pinatti C. O., Bellver, M. A. R. (2024, June). Modules for Practices with Microcontrollers: PIC16F88 and ATmega328 on Arduino. In *2024 XVI Congreso de Tecnología, Aprendizaje y Enseñanza de la Electrónica (TAEE)*. 2024. Pp. 1-6.

31. Govender K., Stubbs J., Wyngaard, A. (2019). Microcontroller-based time interval and correlation measurement for quantum optics experiments. *Measurement Science and Technology*. 2019. Vol. 30(7). P. 075008.

32. Indrawijaya R., Sariningrum R., Sukoco B. E., Muliawandana, D. Fractional-N PLL Synthesizer for FMCW Signal Generator with Dual-Mode Modulation Pattern. *Jurnal Elektronika dan Telekomunikasi*. 2018. Vol. 18(2). Pp. 46-52.

33. Grant N., Geiss B., Field S., Demann A., Chen T. W. Design of a hand-held and battery-operated digital microfluidic device using EWOD for lab-on-a-chip applications. *Micromachines*. 2021. Vol. 12(9). P. 1065.

34. Stoimenov E. C., Antchev H. M., Pandiev I. M. (2019, May). Design and Development of Class-D Audio Power Amplifiers Laboratory Kit for an Electronic Engineering Education. In *2019 X National Conference with International Participation (ELECTRONICA)*. 2019. (pp. 1-4).

35. Gbenga D. E., Shani A. I., Adekunle A. L. Smart walking stick for visually impaired people using ultrasonic sensors and Arduino. *International journal of engineering and technology*. 2017. Vol. 9(5). Pp. 3435-3447.

36. Barrett S. F. Arduino microcontroller processing for everyone! Springer Nature. 2022.

37. Nair V. S., Nair A. S. (2018, April). Portable Wireless Multipurpose Signal Viewer, Analyzer and Generator Using ATMEGA328P MCU and Android. In *2018 3rd International Conference for Convergence in Technology (I2CT)*. 2018. pp. 1-4.

38. Yener S. C., Mutlu R. A microcontroller-based ECG signal generator design utilizing microcontroller PWM output and experimental ECG data. In *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*. 2018. Pp. 1-4.

					КВРКІ. 22024.40.02.40 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

39. Autodesk Tinkercad URL: <https://www.tinkercad.com/> (дата звернення: 20.05.2025).

40. Elimestekawi A., Tammam A., Issa H. (2018, October). Unclonable key Generator Based on Chip signature and SRAM-PUF of ATmega328P chip. In *2018 28th International Conference on Computer Theory and Applications (ICCTA)*. 2018. Pp. 24-29).

41. Gawas H. M., Manikandan J. (2024, May). Design and development of an IoT based Signal Generator for Testing Equipments. In *2024 Zooming Innovation in Consumer Technologies Conference (ZINC)*. 2024. Pp. 151-156.

42. Kasri N. F., Piah M. A. M., Hamzah A., Adzis Z. Microcontroller-based pulse signal controller development for compact high voltage pulse generator: Practical development in food treatment technology. *Journal of Food Processing and Preservation*. 2021. Vol. 45(6). e15531.

43. Szabo R. Developing Different Test Conditions to Verify the Robustness and Versatility of Robotic Arms Controlled by Evolutionary Algorithms. *Electronics*. 2024. Vol. 13(11). P. 2130.

44. Ahmad B., Khan M. F., Goswami G. (2023, May). Implementation of Microcontroller based Variable Frequency Voltage Measurement Sensor for PLC based Standalone Induction Generator Control. In *2023 International Conference on Recent Advances in Electrical, Electronics & Digital Healthcare Technologies (REEDCON)*. 2023. Pp. 36-41.

45. Carminati M., Scandurra G. Impact and trends in embedding field programmable gate arrays and microcontrollers in scientific instrumentation. *Review of Scientific Instruments*. 2021. Vol. 92(9).

46. Amin M., Pamungkas W. W., Harsono D. (2023, December). Application System for Setting Values on High Voltage Power Supply Using MCP4725 Module Based on ATmega328P Microcontroller. In *IAIC International Conference Series*. 2023. Vol. 4. No. 1. Pp. 187-192.

47. Волков І. В., Войтех В. О. Перетворювач частоти для керування

					КВРКІ. 22024.40.02.40 ПЗ	Арк. 70
Зм.	Арк.	№ докум.	Підпис	Дата		

трифазними загальнопромисловими та енергоефективними асинхронними електродвигунами з використанням мікроконтролерів серії DSPIC33 MC та інтелектуальних силових модулів. *Вісник HTУ «ХІІІ»*. 2017. С. 393-396.

48. Amin M., Pamungkas W. W., Harsono, D. (2023, December). Application System for Setting Values on High Voltage Power Supply Using MCP4725 Module Based on ATmega328P Microcontroller. In *IAIC International Conference Series*. 2023. Vol. 4. No. 1. Pp. 187-192.

49. Çiçek S. Microcontroller-based random number generator implementation by using discrete chaotic maps. *Sakarya University Journal of Science*. 2020. Vol. 24(5). Pp. 832-844.

50. Ramya R., Jayachitra A., Priyadharshini S. Design and Implementation of Portable Signal Generator using AVR and Android App. *Turkish Journal of Computer and Mathematics Education*. 2021. Vol. 12(10). Pp. 3274-3280.

51. Maamar A. E. T., Helaimi M. H., Taleb R., Chabni F. (2019, November). Design and control of a single-phase series resonance inverter using an Arduino microcontroller. In *2019 International Conference on Advanced Electrical Engineering (ICAEE)*. 2019. Pp. 1-6).

52. Al-Shorman M. Y., Al-Kofahi M. M., Al-Kofahi O. M. A practical microwatt-meter for electrical energy measurement in programmable devices. *Measurement and Control* 2018. Vol. 51(9-10). Pp. 383-395.

53. ME P. R., Harish L., Krishna Kumar E. (2022, May). IoT based Industrial Automation for Various Load using ATmega328p Microcontroller. In *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*. 2022. Pp. 471-475.

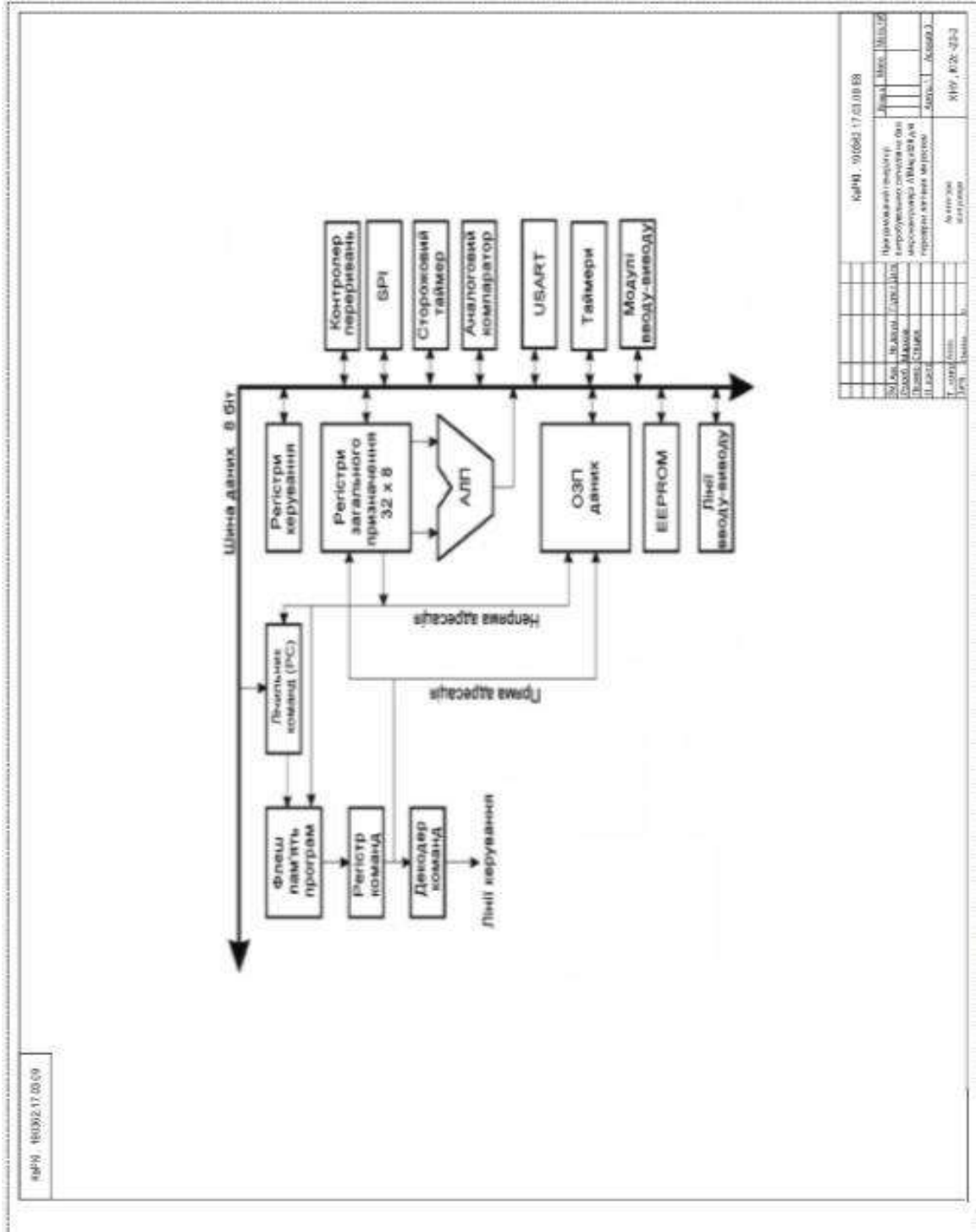
54. Silva W. R., Cunha R. O., Mendes J. B. Development of an integrated device based on the gain/phase detector and Arduino platform for measuring magnetoelastic resonance. *Measurement*. 2025. Vol. 242. P.115819.

55. Mysorekar C., Srivastava R., Sharma H., Kale S. (2023, November). Design and Implementation of a Portable Network Analyzer for In-line Sensor Measurements.

					КВРКІ. 22024.40.02.40 ПЗ	Арк. 71
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток А (обов'язковий)

КОПІЯ КРЕСЛЕННЯ «АРХІТЕКТУРА МІКРОКОНТРОЛЕРА»

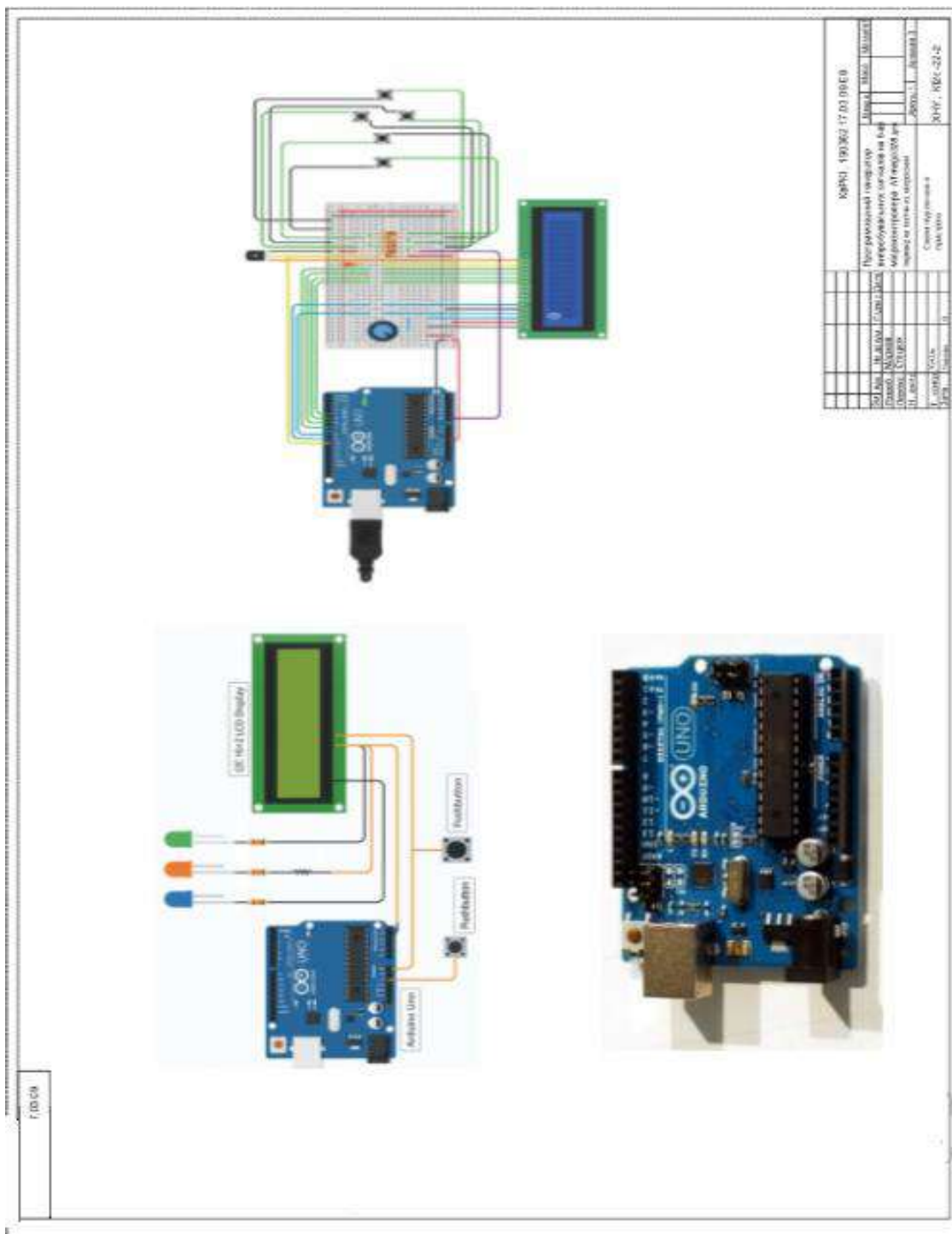


№ 17.00.00.00

№ 17.00.00.00		№ 17.00.00.00	
№ п/п	Вид	Місце	Масштаб
1	Лист	1	1:1
2	Лист	2	1:1
3	Лист	3	1:1
4	Лист	4	1:1
5	Лист	5	1:1
6	Лист	6	1:1
7	Лист	7	1:1
8	Лист	8	1:1
9	Лист	9	1:1
10	Лист	10	1:1
11	Лист	11	1:1
12	Лист	12	1:1
13	Лист	13	1:1
14	Лист	14	1:1
15	Лист	15	1:1
16	Лист	16	1:1
17	Лист	17	1:1
18	Лист	18	1:1
19	Лист	19	1:1
20	Лист	20	1:1
21	Лист	21	1:1
22	Лист	22	1:1
23	Лист	23	1:1
24	Лист	24	1:1
25	Лист	25	1:1
26	Лист	26	1:1
27	Лист	27	1:1
28	Лист	28	1:1
29	Лист	29	1:1
30	Лист	30	1:1
31	Лист	31	1:1
32	Лист	32	1:1
33	Лист	33	1:1
34	Лист	34	1:1
35	Лист	35	1:1
36	Лист	36	1:1
37	Лист	37	1:1
38	Лист	38	1:1
39	Лист	39	1:1
40	Лист	40	1:1
41	Лист	41	1:1
42	Лист	42	1:1
43	Лист	43	1:1
44	Лист	44	1:1
45	Лист	45	1:1
46	Лист	46	1:1
47	Лист	47	1:1
48	Лист	48	1:1
49	Лист	49	1:1
50	Лист	50	1:1
51	Лист	51	1:1
52	Лист	52	1:1
53	Лист	53	1:1
54	Лист	54	1:1
55	Лист	55	1:1
56	Лист	56	1:1
57	Лист	57	1:1
58	Лист	58	1:1
59	Лист	59	1:1
60	Лист	60	1:1
61	Лист	61	1:1
62	Лист	62	1:1
63	Лист	63	1:1
64	Лист	64	1:1
65	Лист	65	1:1
66	Лист	66	1:1
67	Лист	67	1:1
68	Лист	68	1:1
69	Лист	69	1:1
70	Лист	70	1:1
71	Лист	71	1:1
72	Лист	72	1:1
73	Лист	73	1:1
74	Лист	74	1:1
75	Лист	75	1:1
76	Лист	76	1:1
77	Лист	77	1:1
78	Лист	78	1:1
79	Лист	79	1:1
80	Лист	80	1:1
81	Лист	81	1:1
82	Лист	82	1:1
83	Лист	83	1:1
84	Лист	84	1:1
85	Лист	85	1:1
86	Лист	86	1:1
87	Лист	87	1:1
88	Лист	88	1:1
89	Лист	89	1:1
90	Лист	90	1:1
91	Лист	91	1:1
92	Лист	92	1:1
93	Лист	93	1:1
94	Лист	94	1:1
95	Лист	95	1:1
96	Лист	96	1:1
97	Лист	97	1:1
98	Лист	98	1:1
99	Лист	99	1:1
100	Лист	100	1:1

Додаток Б
(обов'язковий)

КОПІЯ КРЕСЛЕННЯ «СХЕМА ПІДКЛЮЧЕННЯ ПРИСТРОЮ»



Завідувачу кафедри КПС
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Володимира МАРКОВА

ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-22-2

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

Мар 2025 року

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Володимир МАРКОВ

Співавтор:

Назва: Марков_Програмований генератор випробувальних сигналів на базі мікроконтролера ATmega328 для перевірки логічних мікросхем

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 5.6%

Коефіцієнт подібності 2: 2.2%

Мікропробіли: 64

Заміна букв: 3

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-05-29 12:23:18.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-05-29

Дата



Доцент Андрій Нічепорук

експерт

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en_US, ru_RU, ua_UA. Errors in the documents: 12%

ID: 242417 Title: БКР Програмований генератор випробувальних сигналів на базі мікроконтролера АТМega328 для перевірки логічних мікросхем Added in a DB: 2025-05-29 Authors: Володимир МАРКОВ Heads: Василь СТЕЦЮК Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	96467	757	1084 (1%)	13 (2%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Марков Володимир Юрійович

Тема: Програмований генератор випробувальних сигналів на базі мікроконтролера ATmega328 для перевірки логічних мікросхем

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 60

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є перевірка логічних мікросхем різних типів за допомогою програмованого генератора випробувальних сигналів на базі мікроконтролера ATmega328.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У межах розділу 1 проведено огляд мікроконтролерної платформи Arduino та основних логічних вентилів. В результаті проведеного огляду, доведено, що Arduino є доступним, гнучким і функціональним інструментом для швидкої розробки та тестування електронних систем, що дає змогу створювати як навчальні, так і промислові прототипи. Проведений огляд показав, що Arduino дозволяє не лише керувати логічними рівнями, але й виводити інформацію на дисплеї, взаємодіяти з кнопками, сенсорами та іншими пристроями. Крім того, Arduino дозволяє моделювати та перевіряти різні шаблони подачі сигналів, що є корисним для аналізу поведінки логічних мікросхем у реальному часі. У другому розділі розглянуто метод подачі послідовних логічних шаблонів на вхід досліджуваних мікросхем. У цьому контексті описано підхід до реалізації таких шаблонів за допомогою мікроконтролера Arduino, що забезпечує простоту керування, програмованість та гнучкість. За допомогою блок-схем та діаграм станів, включаючи скінченний автомат (FSM), було формалізовано логіку роботи генератора шаблонів, що дозволяє легко масштабувати та змінювати

поведінку системи. У третьому розділі у результаті проведеної симуляції програмованого генератора випробувальних сигналів на базі мікроконтролера ATmega328 було успішно змодельовано функціонування всіх основних режимів роботи пристрою. Аналіз отриманих графіків логічних рівнів підтвердив коректність генерації сигналів заданої частоти, тривалості імпульсів і шаблонів, відповідно до написаної прошивки. Графічні результати, отримані під час симуляції, наочно продемонстрували узгодженість між апаратною частиною (формування логічних сигналів на виходах) та програмною логікою перемикання режимів.

4. Позитивні сторони роботи: є універсальним інструментом для перевірки справності логічних мікросхем, що дозволяє подавати на їх входи типові та користувацькі сигнали з різною частотою, послідовністю та логікою, що значно спрощує процес діагностики.

5. Негативні сторони роботи: недостатня увага огляду існуючих рішень при перевірці логічних мікросхем різних типів.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.


8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре С (3,75)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Омешко Оксана Тригорівна, рецензент
каф. 173 ХМУ

“ 9 ” 06 2025 р.

 (підпис)

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмований генератор випробувальних сигналів на базі мікроконтролера ATmega328 для перевірки логічних мікросхем

Автор: Володимир МАРКОВ

Спеціальність: 123- Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Василь СТЕЦЮК, старший викладач

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 3) в якості запозичень в окремих місцях системою зафіксовано послідовності нулів та одиниць при використанні логічних вентилів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 5.60% і адресується до 61 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС

Василь СТЕЦЮК

Андрій Нічепорук

Ольга ПАВЛОВА