

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

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Датчик напруги на базі мікроконтролера Arduino

Назва теми

КвРКІ 210490.21.02.69 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 IV курсу, група KI2-21-2


Підпис

Володимир СУРМА

Ініціали, прізвище

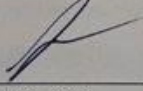
Керівник


Підпис, дата

Вадим ЗАДНІПРОВСЬКИЙ

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

«19» 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

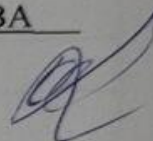
Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.



ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Володимиру СУРМІ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Датчик напруги на базі мікроконтролера Arduino

Керівник проекту (роботи) Вадим ЗАДНІПРОВСЬКИЙ, ст. викладач

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

Затверджена наказом ректора університету від 07.02.2025 р. № 23

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) _____

Постановка задачі щодо її удосконалення та аналіз структурних та функціональних

особливостей Arduino

Проектування системи обробки інформації датчика напруги на базі мікроконтролера

Arduino

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

Блок-схема

Електрично-принципова схема

Емуляція у Wokwi

6. Консультанти розділів дипломного проекту (роботи)

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|----------------|--------------|
| | | завдання видав | завдання при |
| Нормоконтроль | Тетяна КИСІЛЬ, доцент кафедри КПС | | |
| Антиплагіат | Андрій НІЧЕПОРУК, доцент кафедри КПС | | |

№
Р
я
д
к
а

1

2

7. Дата видачі завдання

« 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

| №з/п | Назва етапів (розділів) дипломного проекту (роботи) | Термін виконання етапів проекту (роботи) | Прим |
|------|---|--|-------|
| 1 | Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником | 10.01.2025 | ВИКОН |
| 2 | Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження | 01.02.2025 | ВИКОН |
| 3 | Робота над розділом 1 - постановка задачі щодо її удосконалення та аналіз структурних і функціональних особливостей arduino | 01.03.2025 | ВИКОН |
| 4 | Робота над розділом 2 - проектування системи обробки інформації датчика напруги на базі мікроконтролера arduino | 01.04.2025 | ВИКОН |
| 5 | Робота над розділом 3 - програмно-апаратна реалізація датчика напруги на базі мікроконтролера arduino | 29.04.2025 | ВИКОН |
| 6 | Оформлення пояснювальної записки згідно вимог | 25.05.2025 | ВИКОН |
| 7 | Попередній захист ВКР | 26.05.2025 | ВИКОН |
| 8 | Захист ВКР на засіданні ЕК | Червень 2025 року | |

Студент

Підпис

Володимир СУРМА
Ініціали, прізвище

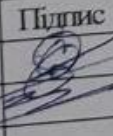
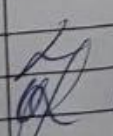
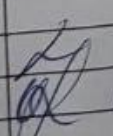
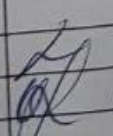
Керівник роботи

Підпис

Вадим ЗАДНІПРОВСЬКИЙ
Ініціали, прізвище

| № р я д к а | Ф о р м а т | Позначення | Найменування | К і л і с т і в | № с к з | П р и м і т к а |
|----------------------------|----------------------------|--------------------------|--------------------------------|--------------------------------------|------------------|--------------------------------------|
| | | | Текстові документи | | | |
| 1 | | КВРКІ 210490.21.02.69 ПЗ | Пояснювальна записка | 64 | | |
| | | | <u>Графічні матеріали</u> | | | |
| 2 | | КВРКІ 210490.21.02.69 Е1 | Блок-схема | 1 | | |
| 3 | | КВРКІ 210490.21.02.69 Е2 | Електрично-принципова схема | 1 | | |
| 4 | | КВРКІ 210490.21.02.69 Е3 | Монтажна плата | 1 | | |

КВРКІ 210490.21.02.69 ВП

| Зм | Арк | № докум | Підпис | Дата | Літера | Аркуш | Аркушів |
|-----------|-----|---------------|---|----------|---------------|-------|---------|
| Розробив | | Сурма |  | | У | 1 | 1 |
| Перевір. | | Задніпровська |  | | ХНУ, КІ2-21-2 | | |
| Н. контр. | | Кисіль |  | 19.06.14 | | | |
| Затв. | | Паалова |  | 19.06.14 | | | |

Відомість проекту

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Датчик напруги на базі мікроконтролера Arduino».

Автор роботи: Володимир СУРМА

Керівник роботи: Вадим ЗАДНІПРОВСЬКИЙ..

Пояснювальна записка: 64 с., 18 рис., 1 табл., 3 дод., 51 джерел.

Графічна частина: 3 креслення

ДАТЧИК НАПРУГИ, МІКРОКОНТРОЛЕР, МОДУЛЬ, ДІЛЬНИК НАПРУГИ, ARDUINO.

Метою цієї дипломної роботи є розробка та всебічне тестування датчика напруги, побудованого на базі мікроконтролерної платформи Arduino, яка є доступною та універсальною основою для створення електронних пристроїв. Це включає проектування архітектури системи з урахуванням оптимального розподілу функцій між апаратними та програмними компонентами, розробку програмного забезпечення для зчитування аналогових даних, їх детальної обробки, додавання часових міток для хронологічного відстеження, візуалізації результатів на дисплейному інтерфейсі та забезпечення бездротової передачі даних вимірювань через Bluetooth-модуль. Крім того, робота передбачає практичну реалізацію всіх компонентів, включаючи інтеграцію датчика, мікроконтролера та периферійних модулів, а також ретельну перевірку функціональності системи через серію тестів, спрямованих на оцінку точності, стабільності та надійності пристрою в різних умовах роботи.



Підпис студента

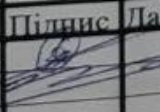


30.05.2025

Дата

ЗМІСТ

| | |
|--|----|
| ПЕРЕЛІК СКОРОЧЕНЬ | 3 |
| ВСТУП | 4 |
| 1 ПОСТАНОВКА ЗАДАЧІ ЩОДО ЇЇ УДОСКОНАЛЕННЯ ТА АНАЛІЗ СТРУКТУРНИХ І ФУНКЦІОНАЛЬНИХ ОСОБЛИВОСТЕЙ ARDUINO | 6 |
| 1.1 Аналіз предметної області і виявлення наявних проблем і завдань..... | 6 |
| 1.2 Порівняльний аналіз переваг та недоліків існуючих рішень..... | 12 |
| 1.3 Висновок до першого розділу..... | 20 |
| 2 ПРОЄКТУВАННЯ СИСТЕМИ ОБРОБКИ ІНФОРМАЦІЇ ДАТЧИКА НАПРУГИ НА БАЗІ МІКРОКОНТРОЛЕРА ARDUINO | 22 |
| 2.1 Віртуальна реалізація датчика напруги..... | 22 |
| 2.2 Призначення компонентів та їх застосування..... | 29 |
| 2.3 Підключення компонентів та їх функціонал..... | 36 |
| 2.4 Висновок до другого розділу..... | 41 |
| 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ДАТЧИКА НАПРУГИ НА БАЗІ МІКРОКОНТРОЛЕРА ARDUINO | 43 |
| 3.1 Огляд модулів та реалізація датчика напруги..... | 43 |
| 3.2 Розробка засобу датчика напруги в емуляторі Wokwi..... | 46 |
| 3.3 Структура коду та функціонал diagram.json..... | 50 |
| 3.4 Структура коду та функціонал sketch.ino..... | 55 |
| 3.5 Тестування проекту в емуляторі Wokwi..... | 59 |
| 3.6 Висновок до третього розділу..... | 64 |
| ВИСНОВКИ | 66 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ | 68 |
| ДОДАТОК А | 73 |
| ДОДАТОК Б | 74 |
| ДОДАТОК В | 75 |

КвРКІ.210490.21.02.69 ПЗ

| Зм. | Арк. | №локум. | Підпис | Дата | Літера | Аркуші | Аркущів |
|----------|------|----------------------|---|------|--------------|--------|---------|
| Виконав | | Володимир СУРМА |  | | у | 2 | 71 |
| Перевір. | | Вадим ЗАДНІПРОВСЬКИЙ |  | | ХНУ КІ2-21-2 | | |
| Н.контр. | | Тетяна КИСЛІТЬ |  | | | | |
| Затвер. | | Ольга ПАВЛОВА | | | | | |

Датчик напруги на базі мікроконтролера Arduino Пояснювальна записка

ПЕРЕЛІК СКОРОЧЕНЬ

AC – Змінний струм

ADC – Аналого-цифровий перетворювач

DC – Постійний струм

RMS – Середньоквадратичне значення

A – Ампер

V – Вольт

ВСТУП

У сучасному світі електроніка відіграє ключову роль у розвитку різноманітних галузей науки, техніки та промисловості, стаючи невід'ємною частиною інноваційних технологій, які трансформують повсякденне життя та промислові процеси. Однією з важливих складових електронних систем є контроль та моніторинг електричних параметрів, зокрема напруги, що забезпечує безпеку, ефективність і надійність роботи пристроїв у різних умовах. Вимірювання напруги є необхідним у багатьох сферах – від побутових пристроїв, таких як зарядні пристрої та побутова техніка, до складних промислових автоматизованих систем, де точність і оперативність даних критично впливають на функціонування обладнання. Розуміння принципів роботи таких систем дозволяє не лише оптимізувати їх використання, а й розробляти нові рішення для підвищення енергоефективності та захисту електронних компонентів.

Метою цієї роботи є розробка, складання та тестування датчика напруги на базі мікроконтролерної платформи Arduino, яка є широко доступною та гнучкою платформою для створення електронних пристроїв. У процесі реалізації передбачено ґрунтовне дослідження принципів вимірювання напруги, включаючи аналіз особливостей обробки аналогових сигналів, їх перетворення в цифровий формат та інтеграцію з апаратним забезпеченням. Окрім того, значна увага приділяється реалізації програмної частини проєкту, яка включає розробку алгоритмів для зчитування даних, їх обробки та відображення результатів вимірювання на різних інтерфейсах, таких як дисплей чи послідовний порт. Такий підхід дозволяє не лише створити функціональний пристрій, а й глибоко зрозуміти взаємодію між апаратними та програмними компонентами.

Цей проєкт має практичне значення, оскільки може бути використаний як основа для створення систем моніторингу електроживлення в реальних умовах, наприклад, для відстеження стабільності напруги в домашніх або промислових мережах. Крім того, він має потенціал для розробки пристроїв захисту від

перенапруги, які здатні своєчасно реагувати на аномалії та запобігати пошкодженню електронного обладнання. Не менш важливою є його цінність для навчальних цілей, де пристрій може слугувати демонстраційним зразком для вивчення принципів роботи аналогових і цифрових систем, а також для ознайомлення з основами програмування мікроконтролерів. Такий багатофункціональний підхід підкреслює актуальність проєкту в контексті сучасних потреб у розвитку електронних технологій та підготовки фахівців у цій галузі.

ПОСТАНОВКА ЗАДАЧІ ЩОДО ЇЇ УДОСКОНАЛЕННЯ ТА АНАЛІЗ СТРУКТУРНИХ І ФУНКЦІОНАЛЬНИХ ОСОБЛИВОСТЕЙ ARDUINO

1.1 Аналіз предметної області і виявлення наявних проблем і завдань

Контроль рівня напруги є важливою частиною електронних систем, особливо в тих випадках, коли підвищена або знижена напруга може призвести до виходу пристрою з ладу або втрати даних. У багатьох випадках виникає потреба в доступному, надійному та точному пристрої для вимірювання напруги з можливістю віддаленого моніторингу та обробки даних.

Одним з рішень для реалізації таких систем є використання платформи Arduino, яка дозволяє створювати функціональні прототипи з мінімальними вимогами до ресурсів. Arduino має вбудовані аналогові входи, до яких можна підключати датчики напруги та зчитувати значення в режимі реального часу. Однак, ефективне використання Arduino для вимірювання напруги вимагає врахування ряду технічних особливостей, серед яких:

- обмежений діапазон вхідної напруги (0-5 В для більшості моделей);
- необхідність створення зовнішнього дільника напруги для вимірювання вищих значень;
- вплив шумів на точність вимірювання;
- необхідність калібрування системи для забезпечення надійних результатів.

Сьогодні на ринку є багато стандартних рішень для вимірювання напруги, але вони часто дорогі, складні в налаштуванні або непридатні для певних застосувань. Тому розробка простого, доступного і точного датчика напруги на базі Arduino є актуальним завданням.

Найважливіші проблеми, які необхідно вирішити:

- обмеження діапазону вхідної напруги Arduino;
- необхідність забезпечення точності та стабільності вимірювань;

- захист Arduino від перенапруги;
- просте відображення або передача отриманих даних.

Тому перед розробником стоїть завдання розробити безпечну, точну, функціональну та просту у використанні систему вимірювання напруги. Проект допоможе поглибити знання з аналогової електроніки, схемотехніки та програмування мікроконтролерів, а також отримати практичний досвід розробки реального пристрою.

Датчик напруги на базі Arduino є простим, але ефективним способом вимірювання аналогової напруги з зовнішніх джерел, таких як батареї чи адаптери. Для його реалізації необхідно мати саму плату Arduino (наприклад, UNO або Nano), два резистори для створення напругового дільника, макетну плату, з'єднувальні дроти, джерело живлення, яке планується аналізувати, а також USB-кабель для завантаження програми. Бажано також мати мультиметр для попередньої перевірки точності вимірювань.

Оскільки Arduino працює з входною напругою до 5 В, необхідно попередньо знизити напругу через дільник на резисторах. Наприклад, щоб вимірювати до 25 В, можна використати резистори номіналом 30 кОм і 7.5 кОм. У такій конфігурації на аналоговий вхід A0 надходитиме напруга, знижена до безпечного рівня. Сигнал подається на A0, а обробка даних відбувається в скетчі Arduino IDE, де проводиться зчитування сигналу, його нормалізація до реального значення вольтажу та вивід у консоль.

В кодї реалізовані всі базові розрахунки: зчитування ADC, перетворення значення в реальні вольти з урахуванням коефіцієнта поділу, а також виведення результату в серійну консоль. Важливо правильно підібрати номінали резисторів, а також не перевищувати граничну напругу, щоб не пошкодити мікроконтролер. За потреби можна додатково захистити схему, використовуючи діоди або оптрони. У подальшому такий пристрій можна легко доопрацювати - наприклад, підключити дисплей, зберігати дані на SD-карту або передавати інформацію бездротово через Bluetooth або Wi-Fi.

Структурно Arduino побудована навколо мікроконтролера, найчастіше ATmega328P, який виконує обчислення та обробку сигналів. Він має вбудовану флеш-пам'ять для зберігання програми, оперативну пам'ять та енергонезалежну пам'ять для даних. До плати підключаються різноманітні цифрові та аналогові пристрої через порти введення/виведення, що дає змогу зчитувати дані з датчиків або керувати виконавчими механізмами. Arduino UNO, як один із найпопулярніших варіантів, має 14 цифрових пінів і 6 аналогових входів, деякі з них підтримують широтно-імпульсну модуляцію.

Плата підключається до комп'ютера через USB, який використовується як для живлення, так і для передачі даних. Окрім USB, Arduino має роз'єм для зовнішнього живлення, зазвичай на 9-12 В постійного струму, а також стабілізатор напруги, що дозволяє використовувати різні джерела живлення. Для забезпечення стабільної роботи використовується кварцовий резонатор із частотою, зазвичай 16 МГц. Крім того, передбачена система автоматичного скидання, що полегшує прошивку мікроконтролера без натискання кнопки Reset.

Функціональність Arduino значною мірою визначається її програмним забезпеченням. Arduino IDE дозволяє писати програми простою мовою, заснованою на C/C++, та завантажувати їх безпосередньо на плату. Для роботи з різноманітними пристроями доступна велика кількість бібліотек, які значно спрощують інтеграцію периферії. Arduino підтримує такі протоколи зв'язку, як UART, SPI та I2C, що дозволяє з'єднувати її з іншими мікроконтролерами, датчиками чи модулями зв'язку.

Завдяки широкій спільноті користувачів існує безліч прикладів, документації та готових рішень. Arduino IDE працює на різних операційних системах, включаючи Windows, macOS та Linux, що забезпечує гнучкість у виборі середовища розробки. Платформу можна розширювати, додаючи Ethernet- або Wi-Fi-модулі, дисплеї, карти пам'яті та інші компоненти, що дає змогу реалізовувати як прості, так і складні системи автоматизації та моніторингу.

Завдяки поєднанню простоти, універсальності та відкритої архітектури, Arduino залишається одним із найзручніших інструментів для створення інтелектуальних електронних систем.

Предметна область охоплює вбудовані системи, IoT, робототехніку та освіту. Сучасні тенденції вказують на зростання попиту на енергоефективні IoT-рішення, бездротові технології (Wi-Fi, Bluetooth, LoRa) та інтеграцію з хмарними сервісами й AI. Проблеми Arduino включають апаратні обмеження (низька потужність, брак підтримки нових протоколів), програмні (обмеження IDE, складність із дебагінгом), складність інтеграції з хмарними платформами та недостатню сумісність із новими датчиками. Початківцям бракує покрокових гайдів для складних проєктів, а досвідчені розробники обмежені можливостями платформи. Для вирішення цих проблем потрібні нові моделі з потужнішими мікроконтролерами, вдосконалення IDE (додавання дебагера, підтримка Python), модулі для хмарної інтеграції та освітні матеріали.

Порівняно з конкурентами, Arduino виграє завдяки простоті, великій спільноті та модульності, але поступається в продуктивності й енергоефективності. Raspberry Pi пропонує високу потужність, підтримку Linux і мультимедіа, але складніший і дорожчий, що робить його менш зручним для простих задач. ESP32 забезпечує вбудовані Wi-Fi/Bluetooth, низьке енергоспоживання й доступну ціну, але має меншу спільноту та обмежену модульність. STM32 вирізняється продуктивністю та підтримкою RTOS, але складний для освоєння й дорожчий. Для вдосконалення Arduino варто інтегрувати Wi-Fi/Bluetooth у базові моделі, розробити гібридну платформу, що поєднує простоту Arduino з потужністю Raspberry Pi, підвищити енергоефективність для IoT і розширити підтримку хмарних сервісів та нових протоколів.

Першою моделлю плати була Arduino Serial (рис. 1.1), в якій використовувався мікроконтролер ATmega8. Її основною перевагою була простота використання, доступна ціна та можливість програмування за допомогою стандартного комп'ютерного середовища без потреби в дорогому обладнанні.



Рисунок 1.1 – Перша модель плати “Arduino Serial” [40]

Історія платформи Arduino почалася у 2005 році в Італії, в маленькому містечку Івреа. У той час група викладачів і студентів Інституту дизайну взаємодії в Івреа шукала доступний спосіб розробляти інтерактивні проекти, які поєднували б апаратне та програмне забезпечення. Традиційні мікроконтролери були складними для вивчення початківцями і вимагали дорогих програмістів та ліцензійного програмного забезпечення.

Щоб вирішити ці проблеми, Массімо Банзі, один із співзасновників платформи, разом з Девідом Куартіеллесом, Томом Ігое, Джанлукою Мартіно та іншими розробив першу версію плати Arduino. Назва «Arduino» походить від бару Bar di Re Arduino в місті Івреа, де творчі люди часто зустрічалися для обговорення ідей. Цей бар, у свою чергу, був названий на честь короля Ардуіно (рис. 1.2) Італії (Re Arduino d'Italia), який правив у 10-11 століттях.

Завдяки відкритому апаратному та програмному забезпеченню Arduino швидко завоювала популярність у всьому світі. Це породило величезну спільноту розробників, які створили тисячі бібліотек, прикладів, модулів і проектів для широкого спектру потреб – від простої домашньої автоматизації до складних робототехнічних систем.



Рисунок 1.2 – Король Ардуїн

Arduino вирізняється не лише своєю архітектурною простотою, а й здатністю до масштабування, що дозволяє будувати як однофункціональні системи, так і багаторівневі розробки з розподіленою логікою. Однією з ключових особливостей є гнучкість у взаємодії з різноманітними типами модулів - сенсорами, реле, контролерами двигунів, бездротовими передавачами тощо. Завдяки цьому Arduino здатна ефективно працювати в умовах реального часу, реагуючи на зміни середовища або керуючи фізичними процесами.

Окрему роль відіграє підтримка великої кількості сумісних плат - від компактних версій, таких як Arduino Nano, до більш потужних, як Arduino Mega або Due, які мають збільшену кількість пінів і більші обсяги пам'яті. Це відкриває

можливості для масштабних проєктів, де потрібна велика кількість периферійних підключень або складна логіка керування.

Arduino також підтримує інтеграцію з хмарними сервісами через додаткові плати розширення (shields), зокрема Wi-Fi та GSM-модулі, що дозволяє збирати дані віддалено, здійснювати моніторинг або реалізовувати віддалене керування пристроями. У таких застосуваннях часто використовуються MQTT або HTTP-протоколи, які легко реалізуються засобами платформи.

Розширення можливостей Arduino також забезпечується використанням графічних середовищ розробки, таких як Ardublock або Tinkercad Circuits, які особливо корисні на етапі початкового проєктування, оскільки дозволяють швидко візуалізувати логіку роботи системи без глибоких знань програмування. Це значно пришвидшує навчання та експериментування з різними схемами.

Ще однією перевагою є сумісність з іншими мовами програмування та платформами, наприклад, можливість зв'язку Arduino з Python через послідовний порт для збору даних або керування зовнішніми програмами. Також доступні середовища на базі Node.js, що дозволяє створювати вебінтерфейси для взаємодії з апаратурою в реальному часі.

Завдяки такій адаптивності Arduino використовується не лише в освітніх цілях, але й у промислових прототипах, домашній автоматизації, системах контролю навколишнього середовища, робототехніці та носимих пристроях. У багатьох випадках ця платформа виконує роль містка між цифровими технологіями та фізичним світом, забезпечуючи просту реалізацію інтерактивної поведінки на основі вхідних даних.

1.2 Порівняльний аналіз переваг та недоліків існуючих рішень

Для створення надійної та функціональної системи вимірювання напруги використовується комбінований підхід. Резистивний дільник напруги розміщується на вході джерела сигналу для регулювання високих значень (наприклад, до 25 В) до безпечного рівня для аналого-цифрового перетворювача.

Резистори повинні мати високу точність (1 % або краще), щоб мінімізувати похибки вимірювання. Після дільника встановлюється захисний ланцюг: послідовний резистор і стабілітрон (рис. 1.3) (наприклад, стабілітрон на 5,1 В) або швидкодіючий TVS-діод(рис. 1.4). Цей захист обмежує напругу на вході АЦП і запобігає її перевищенню, що може призвести до пошкодження Arduino.



Рисунок 1.3 – Стабілітрон [41]



Рисунок 1.4 – Швидкодіючий TVS-діод [42]

Порівняльний аналіз переваг та недоліків рішень для створення датчика напруги на базі Arduino передбачає розгляд типових підходів до реалізації таких

пристроїв, включаючи використання різних апаратних модулів, схем та програмних засобів. Основні рішення включають використання аналогових входів Arduino із дільником напруги, спеціалізованих модулів для вимірювання напруги (наприклад, INA219), а також альтернативних платформ, таких як ESP32. Нижче представлено аналіз цих рішень.

Рішення з використанням аналогових входів Arduino і дільника напруги є найпростішим і найдешевшим. Воно передбачає підключення резисторного дільника напруги до аналогового входу плати (наприклад, Arduino Uno) для зниження вимірювальної напруги до безпечного рівня (0–5 В). Переваги цього підходу включають низьку вартість, оскільки потрібні лише резистори, простоту реалізації, що робить його доступним для початківців, і можливість використання стандартних бібліотек Arduino для зчитування аналогових даних. Однак є значні недоліки: низька точність через шум аналогового входу та залежність від якості резисторів, обмежений діапазон вимірювань (залежить від опору дільника), необхідність калібрування для підвищення точності та відсутність гальванічної розв'язки, що може бути небезпечним при вимірюванні високих напруг.

Використання спеціалізованих модулів, таких як INA219, забезпечує більш точне вимірювання напруги та струму завдяки вбудованому АЦП і підтримці I2C-інтерфейсу. Переваги цього рішення включають високу точність (роздільна здатність до 12 біт), широкий діапазон вимірювань (до 26 В для INA219), просту інтеграцію з Arduino через готові бібліотеки, а також можливість вимірювання струму, що розширює функціонал. Недоліки: вища вартість порівняно з дільником напруги, необхідність додаткового підключення та налаштування I2C, а також залежність від якості модуля, оскільки дешеві моделі можуть мати похибки.

Використання альтернативних платформ, таких як ESP32, дозволяє поєднати вимірювання напруги з бездротовими можливостями (Wi-Fi, Bluetooth). ESP32 має вбудований АЦП і може працювати з дільником напруги або модулями типу INA219. Переваги включають вищу продуктивність завдяки двоядерному процесору, вбудовані бездротові модулі для передачі даних у реальному часі,

низьке енергоспоживання в режимі глибокого сну, що корисно для IoT-додатків, і доступну ціну. Недоліки: складніша програмна логіка порівняно з Arduino, менш розвинена спільнота для специфічних задач вимірювання напруги, а також нижча роздільна здатність АЦП (10 біт у базовій конфігурації), що може впливати на точність.

Порівняно з іншими підходами, Arduino з дільником напруги виграє за простотою та вартістю, але програє за точністю та безпекою. Модуль INA219 є оптимальним для проєктів, де потрібна висока точність і додатковий функціонал, але вимагає більших витрат. ESP32 підходить для IoT-застосунків із бездротовою передачею даних, але його складність може бути надмірною для простих задач. Рекомендації для вдосконалення включають комбінацію Arduino з модулем INA219 для балансу між вартістю та точністю, додавання гальванічної ізоляції для безпеки та використання ESP32 для проєктів, що потребують зв'язку з хмарними сервісами.

Для підвищення точності і стабільності бажано використовувати зовнішній АЦП, наприклад, ADS1115. Ця мікросхема має 16-бітну роздільну здатність, що дозволяє набагато точніше виявляти зміни напруги, ніж внутрішній 10-бітовий АЦП Arduino. Він також має внутрішній підсилювач і дозволяє підключати декілька каналів одночасно. Дані зчитуються ADS1115 через інтерфейс I2C, що робить інтеграцію з Arduino або ESP надзвичайно простою.

Для зручної передачі даних вимірювань використовуються мікроконтролери з інтегрованим модулем зв'язку. Залежно від вимог, це можуть бути ESP32 або ESP8266, які підтримують Wi-Fi і дозволяють передавати дані на сервер, хмару, веб-інтерфейс або через MQTT. Якщо потрібна передача на великі відстані, краще використовувати модуль LoRa. Це дозволяє зручно реалізувати бездротову систему моніторингу, де вимірювання доступні в режимі реального часу з будь-якого місця.

Таким чином, схема дозволяє безпечно вимірювати високі напруги, точно записувати значення, стабільно передавати дані та ефективно захищати мікроконтролер. Це універсальне рішення для багатьох завдань в галузі електроніки, енергомоніторингу та Інтернету речей.

За допомогою вбудованих аналогових входів Arduino дозволяють вимірювати напругу, при цьому аналогові виводи (A0-A5) Arduino зчитують напругу через 10-бітний аналого-цифровий перетворювач (АЦП).

Вбудовані аналогові входи Arduino дозволяють легко вимірювати напругу до 5 В (або 3,3 В) без додаткових компонентів.

Обмеженнями системи є обмежений діапазон вимірювань, який становить від 0 до 5 В (або від 0 до 3,3 В залежно від моделі), що вимагає використання ділника напруги для обробки вищих значень, однак це може призводити до зниження точності через втрату роздільності сигналу. АЦП із 10-бітною роздільною здатністю, що еквівалентно приблизно 4,88 мВ на крок, виявляється недостатньо чутливим для задач, які потребують високої точності, особливо при аналізі невеликих коливань напруги. Крім того, система виявляє підвищену чутливість до електромагнітних перешкод, що може впливати на стабільність результатів, а також не здатна безпосередньо вимірювати змінний струм без інтеграції додаткових схем для його перетворення.

Сфери застосування охоплюють переважно моніторинг низьких постійних напруг, таких як батареї чи сигнали від датчиків, що робить систему придатною для реалізації хобійних проєктів, де пріоритет віддається простоті та доступності, а не максимальній точності.

Модулі, оснащені операційними підсилювачами типу LM358, дозволяють проводити вимірювання напруги до 25 В і вище завдяки застосуванню масштабування сигналу, що адаптує вхідні значення до прийняттого діапазону для подальшої обробки. Підвищена точність досягається за рахунок можливості регулювання рівня підсилення, що забезпечує гнучкість у роботі з різними рівнями напруги. Підключення таких модулів до платформи Arduino здійснюється без особливих зусиль через стандартні роз'єми, а їхня вартість коливається в межах від 1 до 5 доларів США, що робить їх економічно доступними. Водночас варто врахувати певні обмеження: калібрування резисторів вимагає значних зусиль через складність процесу, а для напруг, що перевищують 25–30 В, необхідно передбачити

додатковий захист для запобігання пошкодженню компонентів. Модулі демонструють підвищену чутливість до електромагнітних перешкод виключно в умовах постійного струму (DC), що може впливати на стабільність вимірювань. Потенційні сфери застосування включають контроль стану батарей та сонячних панелей у діапазоні 5–25 В, де важливими є точність і надійність даних.

Модулі, оснащені трансформатором струму типу ZMPT101B, призначені для роботи зі змінною напругою та забезпечують безпечне вимірювання завдяки ефективній електричній ізоляції, дозволяючи точно визначати значення в діапазоні від 0 до 250 В змінного струму (AC). Аналоговий вихід цих модулів, який варіюється в межах 0–5 В, сумісний із платформами на базі Arduino, що полегшує їх інтеграцію в системи обробки даних. Такі модулі знаходять застосування в аналізі електромереж та розробці розумних лічильників енергоспоживання, де важливими є точність і надійність.

Однак їхнє використання має певні обмеження: модулі не призначені для вимірювання постійного струму (DC), що звужує сферу їхньої функціональності. Обробка синусоїдального сигналу для обчислення ефективного значення (RMS) вимагає складних алгоритмів, що може ускладнити програмну реалізацію. Точність вимірювань значною мірою залежить від стабільності частоти мережі, зокрема 50 або 60 Гц, а їхня вартість коливається в межах від 3 до 10 доларів США, що відображає їхню доступність.

Спеціалізовані модулі, такі як INA219, вирізняються високою точністю вимірювань ($\pm 0,5\%$), що забезпечується завдяки застосуванню 12-бітного аналого-цифрового перетворювача, який дозволяє деталізувати дані з підвищеною роздільною здатністю. Інтерфейс I2C, інтегрований у ці модулі, сприяє економії цифрових пінів мікроконтролера, полегшуючи підключення до платформ, таких як Arduino, і забезпечуючи зручність у багатокomпонентних системах. Вони здатні вимірювати напругу до 26 В і струм до 3,2 А, при цьому характеризуються низьким енергоспоживанням, що робить їх придатними для енергоефективних рішень.

Водночас використання таких модулів пов'язане з певними обмеженнями: їхня ціна коливається в межах від 5 до 15 доларів США, що може бути вищим порівняно з іншими альтернативами, а для повноцінної роботи необхідні відповідні бібліотеки програмного забезпечення та базові знання про протоколи I2C. Модулі призначені виключно для роботи з постійним струмом (DC), що звужує їхній діапазон застосування, а для вимірювання великих струмів потребують додаткового шунта для коректного розподілу навантаження. Потенційні сфери використання включають точне моніторинг напруги та струму в системах керування енергією, де критичними є надійність і деталізація даних.

Саморобні схеми характеризуються значною гнучкістю, дозволяючи адаптувати конструкцію до специфічних потреб користувача залежно від поставлених завдань. Їх створення передбачає використання недорогих компонентів, таких як резистори та конденсатори, що робить їх економічно вигідними, а також сприяє практичному засвоєнню основ електроніки через самостійне конструювання. Цей підхід сприяє розвитку навичок аналізу та експериментування в процесі роботи з апаратним забезпеченням.

Разом із тим, такі схеми мають певні обмеження: їх проектування вимагає значних зусиль і глибокого розуміння електричних принципів, а ймовірність помилок у підключенні чи розрахунках може призводити до пошкодження мікроконтролера, зокрема Arduino. Вони також виявляють підвищену чутливість до електромагнітних шумів, що може впливати на стабільність вимірювань, а тривалість розробки часто перевищує час, необхідний для готових модулів. Сфери застосування охоплюють індивідуальні кастомні проєкти, а також навчальні цілі, де пріоритетом є освоєння основ електроніки та набуття практичного досвіду.

Для модулів на базі операційних підсилювачів LM358, які ускладнені калібруванням резисторів, обмежені напругами >25-30 В, чутливі до перешкод і працюють лише з DC, пропонується використовувати прецизійні резистори та потенціометри для налаштування підсилення, замінюючи потенціометр фіксованим резистором після калібрування, а також програму в Arduino IDE для

автоматичного визначення коефіцієнтів корекції. Діапазон розширюється дільником напруги перед LM358 для напруг до 50 В із захистом через діоди Зенера або TVS-діоди. Перешкоди зменшуються конфігурацією диференціального підсилювача та конденсаторами 0.1-1 мкФ на вході й виході. Для низькочастотного АС додається схема випрямлення з діодним мостом і конденсатором. Покращений модуль підходить для моніторингу напруги в системах середнього діапазону (5-50 В), таких як сонячні панелі чи автомобільні батареї.

Модулі для змінної напруги ZMPT101B, які не вимірюють DC, ускладнені обробкою синусоїди для RMS, залежать від частоти 50/60 Гц і коштують \$3-10, можна покращити комбінацією з дільником напруги або INA219 для вимірювання DC. Обробка RMS спрощується бібліотекою EmonLib і збільшенням частоти вибірки до 1000 зразків за період. Точність підвищується калібруванням для конкретної частоти мережі або адаптивними алгоритмами для змінної частоти. Вартість знижується створенням власної плати на базі трансформатора. Покращений ZMPT101B ідеально підходить для «розумних будинків» і моніторингу електромереж (0-250 В АС) з точним вимірюванням RMS.

Спеціалізовані модулі INA219, вартістю від 5 до 15 доларів США, вимагають від користувача базового розуміння протоколу I2C і мають обмеження щодо вимірювань: максимальна напруга становить 26 В, а струм - 3,2 А, при цьому вони призначені виключно для роботи з постійним струмом (DC). Для підвищення економічної ефективності чи адаптації до специфічних потреб можливе використання дешевших альтернатив, таких як INA226, або застосування INA219 лише в ситуаціях, де критична важливість точності вимірювань виправдовує їхню вартість. Робота з інтерфейсом I2C значно спрощується завдяки наявності бібліотеки Adafruit_INA219 та детальної документації, яка включає підтримку адресації для кількох модулів одночасно.

Діапазон вимірювань можна розширити за допомогою зовнішнього дільника напруги для обробки напруг, що перевищують 26 В, або шунтувальних резисторів для великих струмів, хоча це може дещо знизити точність результатів через

додаткові втрати чи спотворення сигналу. Для вимірювання змінного струму (AC) необхідно інтегрувати випрямляч із подальшою обробкою сигналу, що додає складності до конструкції. Оптимізована версія модуля INA219, завдяки своїй високій точності, виявляється придатною для детального моніторингу напруги та струму в системах керування енергією, зокрема в зарядних станціях чи пристроях Інтернету речей (IoT), де важливими є надійність і прецизійність даних.

Запропоновані підходи дозволяють подолати основні недоліки існуючих рішень: вбудовані аналогові входи Arduino набувають ширшої функціональності завдяки інтеграції дільників напруги, технології oversampling (технікою обробки сигналів) та фільтрів, що покращують обробку сигналів у діапазоні низьких напруг. Модулі на базі операційних підсилювачів LM358 адаптуються для ефективного вимірювання середніх напруг до 50 В за умови додавання захисних елементів і систем фільтрації. Модуль ZMPT101B демонструє оптимальну працездатність для вимірювань змінного струму (AC) у електромережах після відповідного калібрування, тоді як INA219 забезпечує найвищу точність для постійного струму (DC), частково долаючи свої обмеження завдяки застосуванню зовнішніх компонентів, таких як дільники чи шунти.

Для реалізації складних проєктів комбіноване використання методів, зокрема ZMPT101B для обробки AC і INA219 для DC, сприяє досягненню максимальної універсальності та надійності системи. Такий підхід дозволяє врахувати специфіку різних типів сигналів і забезпечити стабільність роботи в умовах, де потрібні одночасні вимірювання різноманітних електричних параметрів.

1.3 Висновок до першого розділу

У процесі аналізу структурних і функціональних особливостей Arduino було виявлено, що ця платформа є універсальним і доступним інструментом для розробки різноманітних електронних систем - від навчальних проєктів до прототипів реальних пристроїв. Аналіз предметної області дозволив встановити, що попри широкий спектр застосування, Arduino має обмеження, які можуть

стримувати її використання у складних або промислових умовах. Серед ключових проблем – обмежені апаратні ресурси базових моделей, відсутність вбудованих засобів захисту, а також потреба у додаткових модулях для реалізації сучасних функцій, як-от бездротовий зв'язок чи хмарна інтеграція.

Порівняльний аналіз показав, що головними перевагами Arduino є простота освоєння, велика спільнота користувачів, відкритий код і широкий вибір бібліотек. Недоліками залишаються обмежена продуктивність мікроконтролерів, обмеження по енергоспоживанню та нестача вбудованих інтерфейсів на платах базового рівня. У порівнянні з іншими платформами, такими як Raspberry Pi або ESP32, Arduino поступається в обчислювальній потужності, але виграє в стабільності роботи з фізичними сигналами та керуванні пристроями в реальному часі.

Висновком цього дослідження є розуміння необхідності вдосконалення існуючих рішень, що базуються на Arduino, шляхом інтеграції з більш потужними модулями, оптимізації енергоспоживання, а також розробки адаптивного програмного забезпечення для розширення функціональності. Такий підхід дозволить подолати обмеження платформи та забезпечити її ефективне використання у більш складних і вимогливих технічних проєктах.

ПРОЄКТУВАННЯ СИСТЕМИ ОБРОБКИ ІНФОРМАЦІЇ ДАТЧИКА НАПРУГИ НА БАЗІ МІКРОКОНТРОЛЕРА ARDUINO

2.1 Віртуальна реалізація датчика напруги

Система для вимірювання напруги на базі мікроконтролера Arduino Uno призначена для обробки сигналів від датчика напруги, реалізованого через резистивний дільник, з подальшим виведенням результатів на комп'ютер через послідовний порт із використанням Serial Monitor у середовищі Arduino IDE.

Основою системи є плата Arduino Uno, яка базується на мікроконтролері ATmega328P із 10-бітним аналогово-цифровим перетворювачем (АЦП), здатним вимірювати напругу в діапазоні 0–5 В із роздільною здатністю приблизно 4.88 мВ на рівень (5 В / 1023). Датчик напруги формується за допомогою резистивного дільника, що складається з двох резисторів, наприклад, $R1 = 30 \text{ кОм}$ і $R2 = 20 \text{ кОм}$, які знижують вхідну напругу до безпечного рівня для аналогового входу Arduino.

Принцип роботи дільника ґрунтується на формулі $V_{\text{out}} = V_{\text{in}} * (R2 / (R1 + R2))$, що для вказаних резисторів дає коефіцієнт 0.4, дозволяючи вимірювати напругу до 12.5 В (5 В / 0.4). Для вимірювання вищих напруг, наприклад, до 25 В, можна використати резистори з більшим співвідношенням, такі як $R1 = 80 \text{ кОм}$ і $R2 = 20 \text{ кОм}$, що забезпечує коефіцієнт 0.2 і дозволяє безпечно обробляти напругу до 25 В. Схема підключення включає з'єднання точки між резисторами дільника з аналоговим входом A0 на Arduino, підключення нижнього кінця R2 до GND плати, а також забезпечення спільної землі між Arduino і джерелом напруги, яке може бути, наприклад, батареєю або лабораторним блоком живлення. Живлення Arduino здійснюється через USB-порт від комп'ютера або зовнішнє джерело напруги 7–12 В, що підключається до роз'єму V_{in} . Програмне забезпечення, написане в Arduino IDE, ініціалізує послідовний порт зі швидкістю 9600 бод для передачі даних, зчитує аналогове значення з входу A0, яке лежить у діапазоні 0–1023, конвертує його в

напругу за формулою $\text{voltage} = \text{sensorValue} * (5.0 / 1023.0)$, а потім обчислює вхідну напругу, ділячи отримане значення на коефіцієнт дільника (наприклад, 0.4).

Розроблена система для вимірювання напруги на базі Arduino забезпечує виведення даних у Serial Monitor із заданою періодичністю, яка становить раз на секунду, що дозволяє користувачеві зручно спостерігати за динамікою змін напруги в реальному часі, забезпечуючи наочність і своєчасність аналізу електричних параметрів. Такий інтервал оновлення даних є оптимальним для більшості застосувань, оскільки він дає змогу відстежувати коливання напруги без надмірного навантаження на інтерфейс чи програму. Для досягнення високої точності вимірювань необхідно ретельно враховувати можливі джерела похибок, які можуть впливати на результати. Наприклад, неточність номіналів резисторів, яка зазвичай становить $\pm 5\%$ для стандартних компонентів, може призводити до відхилень у коефіцієнті резистивного дільника. Крім того, шуми в електричній схемі, спричинені зовнішніми електромагнітними перешкодами чи нестабільністю джерела, також погіршують якість даних. Нестабільність опорної напруги Arduino, яка залежить від якості живлення 5 В на платі, є ще одним фактором, що потребує уваги. До цього додається нелінійність аналогово-цифрового перетворювача (АЦП), яка може спотворювати результати, особливо при вимірюванні малих чи граничних значень напруги.

Для підвищення точності та надійності системи рекомендується проводити калібрування, використовуючи еталонне джерело напруги з точно відомим значенням, наприклад, прецизійний блок живлення чи референсну напругу. Цей процес передбачає порівняння вимірних даних із еталонними показниками, після чого в програмне забезпечення вводиться коригувальний коефіцієнт, який компенсує виявлені відхилення. Такий підхід дозволяє значно покращити точність обчислень і забезпечити відповідність результатів реальним значенням. Складання схеми вимагає особливої уваги до деталей, зокрема ретельної перевірки всіх підключень, щоб уникнути помилок, таких як коротке замикання, яке може пошкодити компоненти. Також важливо переконатися, що напруга на аналоговому

вході Arduino не перевищує допустимого рівня 5 В, оскільки це може призвести до виходу з ладу мікроконтролера. Після завершення складання схеми програмний код завантажується на плату через USB-порт, що є стандартним і зручним способом передачі програмного забезпечення. Результати вимірювань перевіряються через Serial Monitor, де значення вхідної напруги відображаються з точністю до кількох мілівольт, що дозволяє оцінити коректність роботи системи та її відповідність поставленим вимогам.

Система має певні обмеження, які необхідно враховувати під час її експлуатації та розробки. Точність вимірювань значною мірою залежить від якості використовуваних компонентів, таких як резистори, конденсатори та джерела живлення. Максимальна напруга, яку можна виміряти, обмежується параметрами резистивного діляника, що вимагає ретельного підбору номіналів резисторів для конкретного діапазону. Відсутність гальванічної розв'язки робить систему непридатною для роботи з високовольтними чи потенційно небезпечними джерелами без застосування додаткових захисних засобів, таких як ізолятори чи оптрони. Це обмеження потребує уваги, особливо при використанні системи в реальних умовах, де безпека є пріоритетом.

Для вдосконалення системи та розширення її функціональності можна реалізувати низку модифікацій, які значно підвищують її практичну цінність. Наприклад, інтеграція LCD-дисплея дозволяє організувати локальне виведення даних, що усуває потребу в постійному підключенні до комп'ютера та робить систему більш автономною. Використання зовнішнього аналогово-цифрового перетворювача з вищою роздільною здатністю, наприклад, 12-бітного чи 16-бітного, забезпечує детальніші вимірювання, що є критичним для застосувань, де потрібна висока точність. Для захисту від перенапруги рекомендується додати стабілітрон або оптрон, як показано на схематичному зображенні (Рис. 2.1), що підвищує надійність і безпеку системи. Крім того, впровадження бездротової передачі даних через модулі Bluetooth або Wi-Fi відкриває можливості для віддаленого моніторингу, дозволяючи користувачеві отримувати дані на смартфон,

комп'ютер чи хмарну платформу в реальному часі. Такі вдосконалення роблять систему більш гнучкою, універсальною та придатною для сучасних інженерних і навчальних завдань, що потребують розширеної функціональності та адаптивності.

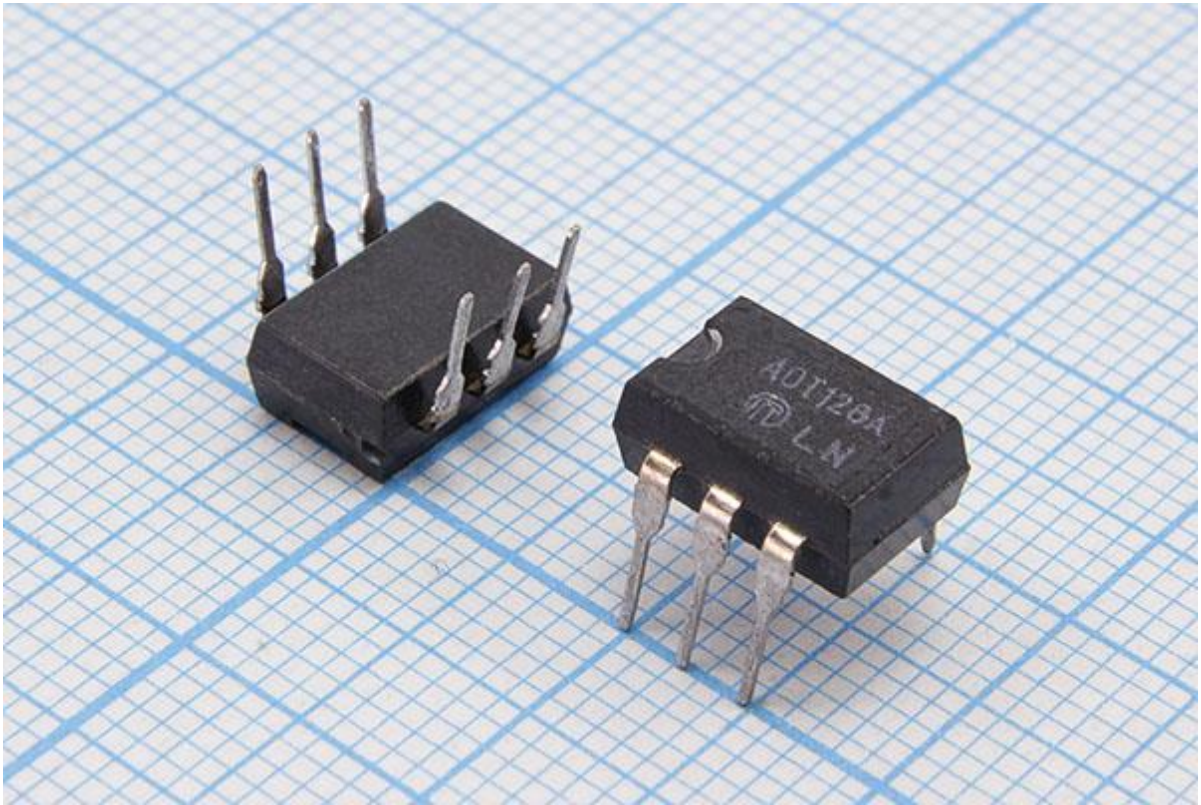


Рисунок 2.1 – Оптрон [51]

Основою системи є плата Arduino Uno, оснащена мікроконтролером ATmega328P, який забезпечує надійну обробку сигналів завдяки вбудованому 10-бітному аналогово-цифровому перетворювачу. Цей перетворювач здатен зчитувати аналогові сигнали в діапазоні від 0 до 5 В, забезпечуючи роздільну здатність приблизно 4.88 мВ на рівень, що є достатнім для більшості базових вимірювань. Для формування датчика напруги використовується резистивний дільник, який складається з двох резисторів, наприклад, із номіналами 60 кОм і 40 кОм. Таке поєднання створює коефіцієнт поділу 0.4, дозволяючи системі безпечно вимірювати напругу до 12.5 В. Для розширення діапазону вимірювань, наприклад,

до 25 В, можна замінити резистори на 80 кОм і 20 кОм, що забезпечить коефіцієнт 0.2, адаптуючи систему до вищих напруг. Середня точка між резисторами, де формується вихідний сигнал, з'єднується з аналоговим входом плати, наприклад, із піною А3. Один кінець дільника підключається до джерела напруги, такого як батарея чи блок живлення, а інший - до загальної маси системи, забезпечуючи коректне вимірювання. Щоб мінімізувати вплив електромагнітних перешкод, паралельно резистору, що йде до землі, підключається керамічний конденсатор на 22 нФ, який ефективно згладжує коливання сигналу. Живлення плати здійснюється через USB-порт під час програмування чи тестування, або через зовнішнє джерело з напругою від 7 до 12 В для автономної роботи. Для забезпечення стабільності живлення рекомендується використовувати стабілізатор напруги, наприклад, LM317, який гарантує постійний рівень напруги навіть при коливаннях джерела.

Програмне забезпечення для системи розробляється в середовищі Arduino IDE, яке є зручним і широко використовуваним інструментом для створення програм для мікроконтролерів. Воно налаштовує послідовний порт на швидкість передачі даних 38400 бод, що забезпечує стабільний зв'язок із комп'ютером для виведення даних через Serial Monitor. Аналоговий сигнал із входу А3 зчитується за допомогою функції `analogRead()`, яка повертає цифрове значення в діапазоні від 0 до 1023, що відповідає напрузі від 0 до 5 В. Отримані дані перераховуються у вхідну напругу з урахуванням коефіцієнта дільника, наприклад, 0.4, за формулою, де вхідна напруга обчислюється як зчитане значення, помножене на 5.0, поділене на 1023.0 і додатково поділене на 0.4. Для підвищення якості та надійності вимірювань застосовується алгоритм ковзного середнього, який обробляє 10 послідовних зразків, що дозволяє ефективно зменшити вплив шумів і коливань сигналу. Оброблені результати виводяться в Serial Monitor із періодичністю 600 мс, що забезпечує зручне спостереження за змінами напруги в реальному часі. Програмне забезпечення також може бути доповнене функцією збереження максимального та мінімального значень напруги за сеанс, що полегшує аналіз коливань і виявлення аномалій у роботі джерела напруги.

Точність вимірювань системи залежить від кількох факторів, зокрема від допусків резисторів, які рекомендується обирати з похибкою $\pm 1\%$, щоб мінімізувати відхилення в коефіцієнті дільника. Коливання напруги живлення можуть впливати на роботу аналогово-цифрового перетворювача, що знижує точність. Для усунення цього рекомендується використовувати зовнішнє джерело опорної напруги, наприклад, REF5025, яке забезпечує високу стабільність і надійність вимірювань. Калібрування системи проводиться шляхом підключення джерела з точно відомою напругою, наприклад, 4 В, що дозволяє порівняти отримані результати з еталонними значеннями. На основі цього в програму вводиться коригувальний множник, який компенсує систематичні похибки, підвищуючи точність обчислень.

Система має певні обмеження, які необхідно враховувати під час її використання. Наприклад, 10-бітна роздільність аналогово-цифрового перетворювача обмежує точність при вимірюванні дуже малих змін напруги, що може бути критичним для високоточних застосувань. Схема також чутлива до зовнішніх електромагнітних перешкод, якщо не застосоване належне екранування, що може спотворювати результати. Залежність від комп'ютера для виведення даних через Serial Monitor обмежує автономність системи, що потребує додаткових рішень для роботи в польових умовах. Максимальна напруга, яку можна вимірювати, залежить від параметрів резистивного дільника, і без додаткових захисних елементів система не придатна для роботи з небезпечними чи високовольтними джерелами.

Для вдосконалення функціональності системи можна реалізувати кілька модифікацій, які значно розширюють її можливості. Наприклад, підключення OLED-дисплея з роздільною здатністю 128x32 пікселів дозволяє виводити дані локально, усуваючи потребу в комп'ютері. Використання високоточного аналогово-цифрового перетворювача, такого як ADS1256 із 24-бітною роздільною здатністю, забезпечує значно детальніше вимірювання. Для захисту схеми від перенапруги рекомендується встановити діод Шоттки, який обмежить небезпечні

стрибки напруги. Бездротова передача даних може бути реалізована через модуль SX1278 (LoRa), що забезпечує зв'язок на великі відстані, або через MQTT-протокол, який дозволяє віддалений доступ до даних через інтернет. Для логування даних із мітками часу використовується RTC-модуль DS3231, який забезпечує точну фіксацію часу для кожного вимірювання, що полегшує аналіз тривалих процесів.

Практична цінність системи полягає в її здатності застосовуватися для контролю напруги в автономних енергосистемах, таких як сонячні чи вітрові установки, де важливо відстежувати стан джерел живлення. Вона також ефективна для тестування електронних компонентів, дозволяючи перевіряти їх поведінку під різними напругами. У навчальних експериментах із цифровою обробкою сигналів система допомагає студентам зрозуміти принципи роботи аналогово-цифрових перетворювачів і програмування. Крім того, вона може бути інтегрована в проєкти автоматизації, наприклад, для керування навантаженням залежно від рівня напруги, що робить її універсальним інструментом для інженерних рішень.

Процес складання системи передбачає ретельне прототипування на макетній платі, де всі компоненти з'єднуються за допомогою проводів. Перевірка з'єднань мультиметром дозволяє виявити та усунути потенційні помилки, такі як короткі замикання чи неправильні контакти. Тестування проводиться з використанням регульованого джерела напруги, що дає змогу оцінити точність вимірювань. Програмний код, розроблений в Arduino IDE, завантажується на плату через USB-порт, після чого результати вимірювань порівнюються з показаннями еталонних приладів, таких як мультиметр, для підтвердження коректності роботи. Система вирізняється високою доступністю компонентів, що робить її економічно вигідною, а також гнучкістю, яка дозволяє адаптувати її до різних завдань. Можливість інтеграції з сучасними технологіями, такими як IoT чи бездротові протоколи, робить цю розробку перспективною для навчальних цілей і практичного використання, забезпечуючи комплексний підхід до вивчення та застосування принципів електроніки й програмування.

2.2 Призначення компонентів та їх застосування

Основні компоненти, що використовуються в проєкті:

- плата Arduino Uno R3;
- резистори 100 кОм і 150 кОм;
- конденсатор керамічний 100 нФ;
- стабілітрон 5.1 В;
- макетна плата (400 контактів);
- з'єднувальні дроти (Male-to-Male, 20 шт.);
- джерело живлення (батарея 9 В + конектор);
- OLED-дисплей 0.96" (128x64, I2C);
- модуль DS3231 RTC;
- модуль HC-06 (Bluetooth).

Це основа системи, яка забезпечує обробку сигналів від датчика напруги та виконання програмного коду. Мікроконтролер ATmega328P із 10-бітним аналогово-цифровим перетворювачем (АЦП) зчитує аналогові сигнали з роздільною здатністю 4.88 мВ у діапазоні 0-5 В. У проєкті плата зчитує сигнал із аналогового входу (наприклад, A0), обробляє його за допомогою запрограмованої логіки та передає результати через послідовний порт або на підключений дисплей. Живлення плати здійснюється через USB для програмування та тестування або через зовнішнє джерело для автономної роботи.

Ці резистори (Рис. 2.2) формують резистивний дільник, який знижує вхідну напругу до безпечного рівня для аналогового входу Arduino. Для резисторів 100 кОм (R1) і 150 кОм (R2) коефіцієнт дільника становить $150 : (100 + 150) = 0.6$, що дозволяє вимірювати напругу до 8.33 В. Високий номінал резисторів зменшує струм через дільник, що знижує енергоспоживання та нагрівання. Точність $\pm 1\%$ забезпечує мінімальну похибку в обчисленнях. Резистори підключаються послідовно: R1 до джерела напруги, R2 до GND, а їхня середня точка - до A0.



Рисунок 2.2 – Резистор 100 кОм. [43]

Конденсатор керамічний на 100 нФ (Рис. 2.3) застосовується для згладжування високочастотних перешкод у сигналі, що надходить до аналогового входу Arduino. Підключений паралельно резистору, з'єднаному з GND, він фільтрує шуми від джерела напруги чи зовнішніх електромагнітних впливів, забезпечуючи стабільніші вимірювання.



Рисунок 2.3 – Конденсатор керамічний 100 нФ [44]

Стабілітрон на 5.1 В (1N4733A) захищає мікроконтролер від перенапруги. У разі перевищення напруги на вході A0 (наприклад, через некоректне підключення) він обмежує її до безпечного рівня, запобігаючи пошкодженню плати. Стабілітрон розміщується між аналоговим входом і масою.

Макетна плата на 400 контактів (Рис. 2.4) слугує основою для швидкого складання схеми без паяння. На ній розміщуються Arduino, резистори, конденсатор, стабілітрон та інші елементи, що дозволяє легко модифікувати систему під час тестування. Її модульна структура забезпечує гнучкість у підключенні компонентів, що сприяє швидкому прототипуванню. Плата оснащена двома основними блоками по 300 і 100 контактів, які розділені центральною канавкою для розміщення мікроконтролерів, таких як Arduino. Завдяки контактам із пружинним механізмом забезпечується надійне з'єднання без ризику пошкодження елементів. Її компактний розмір і зносостійкість роблять її ідеальним інструментом для багаторазового використання в процесі розробки та налагодження системи.

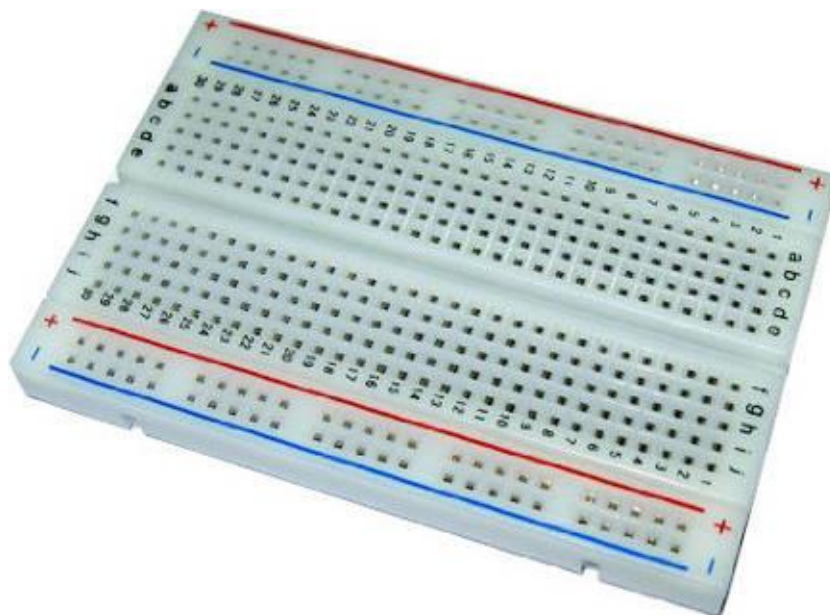


Рисунок 2.4 – Макетна плата на 400 контактів [45]

З'єднувальні дроти типу Male-to-Male (Рис. 2.5) забезпечують електричні зв'язки між компонентами на макетній платі. Вони використовуються для

підключення датчика напруги до Arduino, а також для інтеграції дисплея чи Bluetooth-модуля, гарантуючи надійний контакт і гнучкість у зміні конфігурації. Такі дроти дозволяють легко адаптувати схему до різних експериментальних потреб, що є важливим для швидкого прототипування. Ці дроти виготовляються з високоякісних матеріалів, що забезпечує стійкість до зносу під час багаторазового використання. Завдяки зручним з'єднувачам типу "тато-тато" вони забезпечують швидке підключення без необхідності пайки, що підвищує безпеку та зручність роботи. Їхня довжина та кольорове маркування полегшують ідентифікацію та організацію проводів у складних схемах.



Рисунок 2.5 – З'єднувальні дроти типу Male-to-Male [46]

Батарея 9 В із конектором (Рис. 2.6) забезпечує автономне живлення системи, коли USB-підключення недоступне. Конектор під'єднується до роз'єму Vin на Arduino, дозволяючи проводити вимірювання в автономному режимі, наприклад, для тестування напруги акумуляторів у польових умовах. Така конфігурація забезпечує портативність пристрою, що є особливо зручним для використання в умовах, де доступ до стаціонарного живлення обмежений. Ця батарея гарантує стабільне живлення протягом тривалого часу, залежно від споживання струму

всіма компонентами системи. Для оптимальної роботи рекомендується контролювати рівень заряду батареї, щоб уникнути переривання вимірювань через її розрядку. Конектор також дозволяє легко замінити батарею без необхідності розбирання схеми, що підвищує практичність пристрою в експлуатації.



Рисунок 2.6 – Батарея 9 В із конектором [47]

OLED-дисплей 0.96" (128x64, I2C) (Рис. 2.7) відображає виміряну напругу локально, усуваючи потребу в комп'ютері. Підключений через інтерфейс I2C до пінів A4 (SDA) і A5 (SCL), він показує значення напруги в реальному часі, що зручно для демонстрації роботи системи чи використання в автономних умовах.

Цей дисплей підтримує налаштування шрифтів і розміру тексту, що дозволяє оптимізувати відображення інформації залежно від потреб користувача. Для живлення він підключається до шини 5V і GND, що забезпечує сумісність із Arduino без додаткових адаптерів. Його низьке енергоспоживання робить його ідеальним для автономних систем із батарейним живленням, таких як портативні вимірювальні пристрої. Завдяки компактним розмірам і високому контрасту

зображення дисплей забезпечує чітке відображення даних навіть у слабкому освітленні.



Рисунок 2.7 – OLED-дисплей 0.96" (128x64, I2C) [48]

Модуль DS3231 RTC (Рис. 2.8) додає функцію фіксації часу, дозволяючи прив'язувати вимірювання до міток часу. Підключений через I2C, він забезпечує точний час для логування даних, що корисно для аналізу тривалих вимірювань, наприклад, при моніторингу заряду батареї.

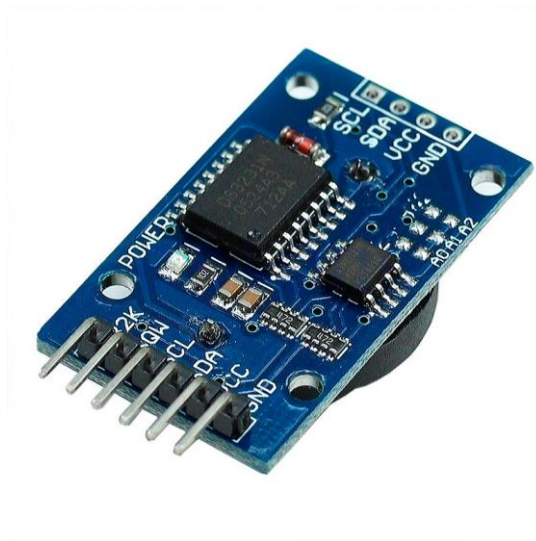


Рисунок 2.8 – Модуль DS3231 RTC [49]

Модуль HC-06 (Bluetooth) (Рис. 2.9) забезпечує бездротову передачу даних про напругу на смартфон чи комп'ютер. Підключений до пінів TX/RX через SoftwareSerial (наприклад, піни 10 і 11), він дозволяє віддалено контролювати показники через мобільний додаток, додаючи сучасний аспект до проєкту.



Рисунок 2.9 – Модуль HC-06 (Bluetooth) [50]

У системі резистивний дільник знижує напругу джерела (наприклад, 6 В) до рівня, безпечного для А0. Конденсатор і стабілітрон забезпечують чистоту сигналу та захист. Arduino обробляє сигнал, конвертуючи цифрові значення в напругу з урахуванням коефіцієнта дільника (0.4). Дані відображаються на OLED-дисплеї, передаються через Bluetooth або виводяться в Serial Monitor. Модуль DS3231 додає часові мітки для логування. Усі компоненти зібрано на макетній платі, а батарея 9 В забезпечує автономність. Ця конфігурація ідеально підходить для дипломної роботи, демонструючи принципи обробки аналогових сигналів, бездротового зв'язку та локального виведення даних, з можливістю розширення, наприклад, додаванням SD-карти чи IoT-функцій.

2.3 Підключення компонентів та їх функціонал

Плата Arduino Uno R3 виступає центральним елементом системи, забезпечуючи обробку аналогових сигналів, виконання програмного коду та координацію роботи всіх підключених модулів. Оснащена мікроконтролером ATmega328P, вона має вбудований 10-бітний аналогово-цифровий перетворювач, який дозволяє зчитувати напругу в діапазоні від 0 до 5 В із роздільною здатністю приблизно 4.88 мВ на рівень. Аналоговий вхід, зокрема пін A0, використовується для отримання сигналу від резистивного дільника, який формує датчик напруги. Крім того, плата забезпечує живлення для периферійних пристроїв через піни 5V і 3.3V, а також підтримує зв'язок через послідовний порт для передачі даних на комп'ютер або через піни для бездротового зв'язку. Живлення плати може здійснюватися через USB-кабель, що зручно під час програмування та тестування, або через зовнішнє джерело, підключене до піна Vin, що забезпечує автономність системи. У даній системі плата обробляє вхідні дані, виконує обчислення для конвертації цифрових значень у напругу, координує виведення результатів на дисплей і через Bluetooth, а також інтегрує часові мітки від модуля RTC.

Резистивний дільник, сформований із резисторів на 100 кОм (R1) і 150 кОм (R2), є ключовим елементом датчика напруги, знижуючи вхідну напругу до рівня, безпечного для аналогового входу Arduino. Завдяки співвідношенню резисторів, коефіцієнт дільника становить 0.4, що дозволяє вимірювати напругу до 8.33 В, оскільки максимальна напруга на вході A0 не повинна перевищувати 5 В. Позитивний вивід джерела напруги, наприклад, акумулятора чи лабораторного блоку живлення, підключається до одного кінця резистора R1. Другий кінець R1 з'єднується з одним кінцем R2 і одночасно з піною A0, формуючи вихідну напругу дільника. Кінець резистора R2 підключається до загальної шини маси (GND), яка об'єднує всі компоненти системи. Така конфігурація забезпечує пропорційне зниження напруги, що дозволяє Arduino точно зчитувати значення, які потім програмно перераховуються у вхідну напругу з урахуванням коефіцієнта дільника.

Вибір резисторів із високими номіналами зменшує струм через дільник, що сприяє енергоефективності та зниженню тепловиділення.

Керамічний конденсатор на 100 нФ відіграє роль фільтра, який згладжує високочастотні перешкоди в сигналі, що надходить до аналогового входу A0. Він підключається паралельно резистору R2, тобто один його вивід з'єднується з піною A0 (середньою точкою дільника), а другий - із шиною GND. Така конфігурація дозволяє усунути коливання напруги, спричинені електромагнітними шумами або нестабільністю джерела, що особливо важливо при роботі з реальними джерелами напруги, такими як батареї чи блоки живлення. Фільтрація сигналу забезпечує стабільніші дані, що підвищує точність вимірювань і знижує ймовірність помилок у цифровій обробці.

Стабілітрон на 5.1 В (1N4733A) використовується як захисний елемент, що запобігає пошкодженню мікроконтролера в разі перевищення напруги на вході A0. Його катод підключається до середньої точки резистивного дільника (A0), а анод - до шини GND, створюючи паралельне з'єднання з конденсатором. Якщо напруга на A0 перевищує 5.1 В через помилку в підключенні чи стрибок напруги, стабілітрон обмежує її до безпечного рівня, захищаючи аналоговий вхід Arduino. Це забезпечує надійність системи в реальних умовах, де можливі непередбачені відхилення параметрів джерела напруги.

Макетна плата на 400 контактів слугує основою для фізичного складання системи, дозволяючи швидко та без паяння з'єднати всі компоненти. Вона містить шини живлення та землі, які розподіляють напругу 5 В, 3.3 В і GND від Arduino до периферійних модулів. На макетній платі розміщуються резистори, конденсатор, стабілітрон, а також підключення для дисплея, RTC і Bluetooth-модуля. Завдяки модульній конструкції макетної плати, система легко модифікується, що дозволяє тестувати різні конфігурації під час розробки дипломного проєкту. З'єднувальні дроти типу Male-to-Male забезпечують електричні контакти між усіма компонентами, дозволяючи гнучко з'єднувати шини Arduino з іншими модулями та датчиком. Для чіткості рекомендується використовувати дроти різного кольору:

червоні для ліній живлення, чорні для GND, інші кольори для сигнальних з'єднань, що полегшує діагностику та складання.

Джерело живлення, представлене батареєю 9 В із конектором, забезпечує автономне функціонування системи, що є важливим для демонстрації роботи поза лабораторними умовами. Позитивний вивід батареї підключається до піна Vin на Arduino, а негативний – до шини GND, що об'єднує всі компоненти. Конектор спрощує заміну батареї, а напруга 9 В є оптимальною для стабільної роботи плати Arduino через вбудований регулятор напруги. У разі роботи з комп'ютером живлення подається через USB-кабель, що дозволяє одночасно програмувати плату та передавати дані через послідовний порт. Така гнучкість у виборі джерела живлення робить систему універсальною для різних сценаріїв використання, від тестування в лабораторії до польових вимірювань.

OLED-дисплей розміром 0.96 дюйма з роздільною здатністю 128x64 пікселів і інтерфейсом I2C забезпечує локальне відображення вимірюваної напруги, що робить систему автономною від комп'ютера. Його пін VCC підключається до шини 5V на Arduino, пін GND – до загальної шини маси, а піни SDA і SCL – до пінів A4 і A5 Arduino відповідно. Інтерфейс I2C дозволяє передавати дані на дисплей, де вони відображаються у вигляді числових значень напруги, наприклад, “V = 6.25 В”, із частотою оновлення, визначеною програмно. Це забезпечує зручність моніторингу в реальному часі та підвищує презентабельність системи для дипломної роботи, демонструючи її здатність працювати без зовнішнього ПК.

Модуль реального часу DS3231 RTC додає функцію фіксації часу, дозволяючи прив'язувати кожне вимірювання напруги до конкретної мітки часу. Це особливо корисно для довгострокового моніторингу, наприклад, при аналізі зміни напруги акумулятора протягом дня. Модуль підключається через той же інтерфейс I2C: пін VCC до шини 5V, GND до шини маси, SDA до A4, SCL до A5, паралельно з дисплеєм. Завдяки високій точності вбудованого годинника, DS3231 забезпечує надійне логування даних, які можуть бути збережені в пам'яті модуля або передані для подальшого аналізу. У програмі Arduino модуль використовується

для додавання часових міток до виведених даних, що підвищує інформативність системи.

Модуль Bluetooth HC-06 забезпечує бездротову передачу даних про напругу на зовнішні пристрої, такі як смартфон чи комп'ютер. Його пін VCC підключається до шини 3.3V (або 5V, якщо модуль підтримує), GND – до шини маси, TXD – до піна 10 Arduino (RX для SoftwareSerial), а RXD – до піна 11 через дільник напруги з резисторів, щоб знизити 5 В до 3.3 В і захистити модуль. Це дозволяє передавати виміряні значення напруги в реальному часі через Bluetooth-з'єднання, наприклад, до мобільного додатка, що забезпечує віддалений моніторинг. Така функціональність додає сучасний аспект до системи, роблячи її привабливою для демонстрації в дипломній роботі.

Робота системи починається з ініціалізації всіх модулів: послідовний порт налаштовується на швидкість, наприклад, 38400 бод, I2C-інтерфейс активується для дисплея та RTC, а SoftwareSerial – для Bluetooth. Аналогове значення з A0 зчитується з частотою, наприклад, раз на 600 мс, обробляється та відображається у вигляді напруги. Дисплей показує поточне значення, RTC додає часові мітки, а Bluetooth передає дані на смартфон. У разі використання USB дані також надсилаються в Serial Monitor для аналізу. Така інтеграція компонентів забезпечує гнучкість і багатофункціональність системи, дозволяючи використовувати її для моніторингу напруги в акумуляторах, сонячних панелях чи лабораторних експериментах. (Рис. 2.10). Система адаптується до різних умов роботи завдяки можливості регулювання частоти зчитування та налаштування портів, що робить її придатною для тривалих спостережень. Для підвищення точності обробки даних можна збільшити кількість зразків для усереднення, що особливо корисно при аналізі нестабільних джерел живлення. Використання Bluetooth і Serial Monitor одночасно дозволяє паралельно вести моніторинг у реальному часі та зберігати лог для подальшого аналізу, що розширює можливості дослідження. Додатково, система може бути інтегрована з іншими модулями, такими як SD-карта, для локального зберігання даних, що підвищує її автономність у польових умовах.

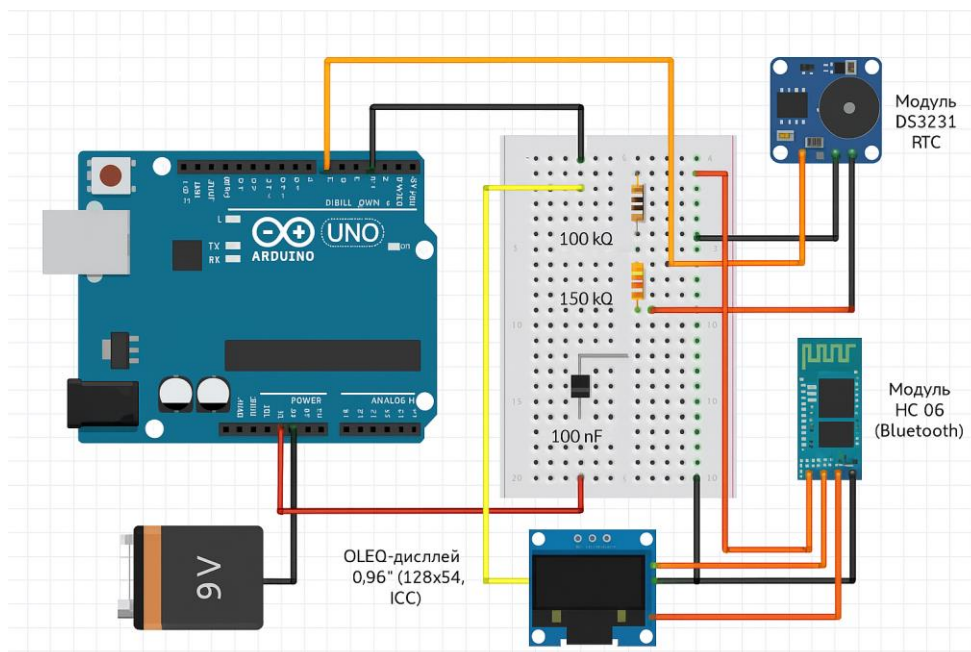


Рисунок 2.10 – Схема підключення.

Структура коду та функціонал файлу `sketch.ino` забезпечили реалізацію основного алгоритму обробки даних, який охоплює зчитування 10 зразків із аналогового входу A0, обчислення їхнього середнього значення для зменшення впливу шумів, перетворення отриманих даних у значення напруги з урахуванням коефіцієнта резистивного дільника, форматування міток часу, отриманих від модуля RTC, а також виведення результатів на OLED-дисплей, у Serial Monitor і через Bluetooth-модуль. Цей код виявив ознаки модульності та гнучкості, надаючи можливість проводити калібрування через введення коригувальних множників і адаптувати систему до додаткових функцій, таких як збереження даних у логах чи інтеграція з технологіями Інтернету речей (IoT), що робить його придатним для розширення функціоналу в майбутніх розробках. Водночас обмеження, зумовлені 10-бітною роздільною здатністю аналого-цифрового перетворювача (АЦП), який забезпечує лише 1024 рівні та крок приблизно 4,88 мВ при стандартному діапазоні 0-5 В, і витратами пам'яті на форматування текстових рядків, підкреслюють потребу в подальшій оптимізації для забезпечення ефективності при виконанні більш складних завдань, таких як обробка великих обсягів даних чи робота з високоточними вимірюваннями. Така оптимізація може включати використання

зовнішніх АЦП із вищою роздільною здатністю або вдосконалення алгоритмів для зменшення навантаження на пам'ять, що є важливим для підвищення загальної продуктивності системи.

2.4 Висновок до другого розділу

Розроблена система вимірювання напруги на базі мікроконтролера Arduino Uno R3 демонструє ефективне поєднання апаратних і програмних компонентів для точного збору, обробки та відображення даних про електричну напругу в реальному часі. Використання плати Arduino Uno R3 як центрального елемента забезпечує надійну обробку аналогових сигналів завдяки вбудованому 10-бітному аналогово-цифровому перетворювачу, дозволяючи з високою точністю зчитувати значення напруги. Резистивний дільник, сформований резисторами на 100 кОм і 150 кОм, ефективно адаптує вхідну напругу до безпечного рівня, тоді як керамічний конденсатор на 100 нФ і стабілітрон на 5.1 В гарантують стабільність сигналу та захист плати від потенційних перенапруг. Макетна плата та з'єднувальні дроти забезпечують гнучкість і простоту складання, дозволяючи швидко модифікувати систему під час тестування, що є важливим для дослідницьких і навчальних цілей.

Інтеграція OLED-дисплея з інтерфейсом I2C додає можливість локального відображення вимірених значень, що робить систему автономною та зручною для демонстрації. Модуль DS3231 RTC розширює функціональність, додаючи точні часові мітки до даних, що необхідно для довгострокового моніторингу. Модуль Bluetooth HC-06 забезпечує бездротову передачу даних, що дозволяє віддалено контролювати показники через смартфон або комп'ютер, підвищуючи практичну цінність системи. Використання батареї 9 В із конектором забезпечує автономність, що робить систему придатною для роботи в польових умовах.

Система вирізняється простотою реалізації, доступністю компонентів і високою адаптивністю, що дозволяє використовувати її в широкому спектрі

застосувань - від навчальних експериментів із електроніки та програмування до практичного моніторингу напруги в акумуляторах, сонячних панелях чи інших енергосистемах. Можливість розширення, наприклад, додаванням модулів для збереження даних чи інтеграцією з IoT-платформами, відкриває перспективи для подальшого вдосконалення.

ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ДАТЧИКА НАПРУГИ НА БАЗІ МІКРОКОНТРОЛЕРА ARDUINO

3.1 Огляд модулів та реалізація датчика напруги

Система вимірювання напруги на базі мікроконтролера Arduino представляє собою комплексне рішення, яке забезпечує точний збір, обробку та відображення даних про електричну напругу в реальному часі, поєднуючи апаратні компоненти з програмним забезпеченням для виконання складних завдань. Основою системи є плата Arduino Uno R3, оснащена мікроконтролером ATmega328P, який має вбудований 10-бітний аналогово-цифровий перетворювач, здатний зчитувати аналогові сигнали в діапазоні від 0 до 5 В із роздільною здатністю приблизно 4.88 мВ на рівень, що дозволяє проводити вимірювання з достатньою точністю для більшості навчальних і практичних застосувань. Ця плата обробляє сигнали від датчика напруги, виконує програмну логіку, координує роботу периферійних модулів і забезпечує виведення даних через різні інтерфейси, що робить її універсальним інструментом для реалізації проєктів у галузі електроніки. Для створення датчика напруги використовується резистивний дільник, який складається з резисторів на 100 кОм і 150 кОм, що дозволяє знизити вхідну напругу до безпечного рівня для аналогового входу плати. Позитивний вивід джерела напруги, наприклад, акумулятора чи лабораторного блоку живлення, підключається до одного кінця першого резистора, тоді як другий його кінець з'єднується з першим виводом другого резистора і аналоговим входом А0 на платі Arduino. Кінець другого резистора підключається до загальної шини маси, що забезпечує коректне вимірювання напруги. Високі номінали резисторів сприяють зменшенню струму через дільник, що підвищує енергоефективність системи та знижує тепловиділення, що є важливим для тривалої роботи.

Керамічний конденсатор на 100 нФ відіграє роль фільтра, усуваючи високочастотні перешкоди в сигналі, що надходить до аналогового входу. Він

підключається паралельно другому резистору, з'єднуючи середню точку дільника з шиною маси, що дозволяє згладжувати коливання, спричинені електромагнітними шумами чи нестабільністю джерела напруги, забезпечуючи більш стабільні та надійні вимірювання. Стабілітрон на 5.1 В слугує захисним елементом, запобігаючи пошкодженню плати в разі перевищення напруги на вході A0. Його катод підключений до середньої точки дільника, а анод - до шини маси, що обмежує напругу до безпечного рівня 5.1 В, захищаючи мікроконтролер від можливих перенапруг, які можуть виникнути через помилки в підключенні чи стрибки напруги в джерелі.

Макетна плата на 400 контактів виступає основою для складання системи, дозволяючи швидко та без паяння з'єднувати всі компоненти, що робить її ідеальною для прототипування та тестування в рамках дипломного проєкту. Вона забезпечує зручне розміщення шин живлення 5 В і 3.3 В, а також маси, які розподіляють енергію до всіх модулів. З'єднувальні дроти типу Male-to-Male забезпечують електричні контакти між компонентами, дозволяючи гнучко конфігурувати схему. Використання проводів різного кольору, наприклад, червоного для ліній живлення та чорного для маси, полегшує складання та діагностику, знижуючи ризик помилок підключення. Джерело живлення, представлене батареєю 9 В із конектором, забезпечує автономність системи, що є важливим для роботи поза лабораторними умовами, наприклад, під час польових вимірювань. Позитивний вивід батареї підключається до піна Vin на платі, а негативний - до шини маси, що забезпечує стабільне живлення через вбудований регулятор напруги. USB-підключення до комп'ютера використовується для програмування та тестування, дозволяючи одночасно передавати дані через послідовний порт.

OLED-дисплей розміром 0.96 дюйма з роздільною здатністю 128x64 пікселів і інтерфейсом I2C забезпечує локальне відображення вимірюваної напруги, що робить систему незалежною від комп'ютера. Його пін живлення підключається до шини 5 В, маса - до шини GND, а піни даних і тактового сигналу - до пінів A4 і A5 на

Arduino, що дозволяє передавати дані через протокол I2C. Дисплей відображає значення напруги з точністю до двох знаків після коми, що зручно для моніторингу в реальному часі та демонстрації роботи системи. Модуль реального часу DS3231 додає до вимірювань мітки часу, що необхідно для логування даних у довгострокових експериментах, наприклад, при аналізі зміни напруги акумулятора протягом тривалого періоду. Він підключається паралельно дисплею до тих самих пінів A4 і A5, а також до шин 5 V і GND, забезпечуючи точну фіксацію часу завдяки вбудованому годиннику. Модуль Bluetooth HC-06 дозволяє передавати дані бездротово на зовнішні пристрої, такі як смартфон чи комп'ютер, що забезпечує віддалений моніторинг. Його пін живлення підключається до шини 3.3 V (або 5 V, якщо модуль підтримує), маса – до GND, вихід даних – до піна 10, а вхід – до піна 11 через дільник напруги з резисторів 1 кОм і 2 кОм для захисту модуля від надмірної напруги.

Програмне забезпечення, розроблене в середовищі Arduino IDE, відповідає за ініціалізацію всіх модулів, включаючи послідовний порт для зв'язку з комп'ютером, інтерфейс I2C для дисплея та RTC, а також SoftwareSerial для Bluetooth. Аналогове значення з входу A0 зчитується регулярно, обробляється шляхом усереднення кількох зразків для зменшення шумів і конвертується в напругу з урахуванням коефіцієнта дільника. Дані виводяться на дисплей, у Serial Monitor і через Bluetooth із частотою оновлення 600 мс, що забезпечує плавне спостереження за змінами напруги. Часові мітки від RTC додаються до кожного вимірювання, що полегшує аналіз даних. Точність вимірювань залежить від якості резисторів, які бажано обирати з допуском $\pm 1\%$, щоб мінімізувати похибки дільника. Нестабільність живлення може впливати на АЦП, тому використання зовнішнього джерела опорної напруги, наприклад, REF5025, значно підвищує надійність. Калібрування проводиться шляхом порівняння даних із еталонним джерелом напруги, наприклад, 4 V, із подальшим введенням коригувального множника в програму для компенсації систематичних відхилень.

Система має обмеження, зокрема 10-бітну роздільність АЦП, яка знижує точність при вимірюванні малих змін напруги, чутливість до зовнішніх перешкод без належного екранування та обмеження максимальної напруги параметрами дільника. Відсутність гальванічної ізоляції робить її непридатною для високовольтних джерел без додаткового захисту. Для вдосконалення можна додати дисплей із роздільною здатністю 128x32 пікселів для локального виведення, високоточний АЦП, наприклад, ADS1256 із 24-бітною роздільною здатністю, або захисний діод Шотткі для безпеки. Бездротова передача через LoRa-модуль SX1278 чи Wi-Fi з MQTT-протоколом забезпечить віддалений доступ до даних, а SD-карта дозволить зберігати вимірювання для подальшого аналізу. Система застосовується для моніторингу напруги в автономних енергосистемах, тестування електронних компонентів, навчальних експериментів із програмування та обробки сигналів, а також у проєктах автоматизації для керування навантаженням залежно від напруги. Складання виконується на макетній платі з ретельною перевіркою з'єднань мультиметром, а тестування проводиться з регульованим джерелом для порівняння результатів із еталонними приладами. Завдяки доступності компонентів, гнучкості та можливості інтеграції з сучасними технологіями, система є ідеальним рішенням для навчальних і практичних завдань, демонструючи принципи роботи з аналоговими сигналами та програмування в інженерних проєктах.

3.2 Розробка засобу датчика напруги в емуляторі Wokwi

Емулятор Wokwi є популярним веб-інструментом, який використовується для створення, симуляції та тестування електронних проєктів, зокрема тих, що базуються на мікроконтролерах Arduino. Його широкі функціональні можливості роблять платформу корисною як для початківців, так і для досвідчених інженерів і розробників, які працюють у сфері вбудованих систем, електроніки та автоматизації. З огляду на зростаючу потребу в ефективному моделюванні апаратних рішень без безпосереднього використання фізичних компонентів,

Wokwi стає релевантним інструментом у контексті академічних і прикладних розробок, зокрема таких, як датчик напруги на основі Arduino, що розглядається в межах даного проєкту.

Однією з ключових переваг Wokwi є повна доступність - інструмент працює безпосередньо в браузері, не потребує встановлення спеціального програмного забезпечення чи драйверів. Це значно спрощує запуск нових проєктів, знижує бар'єр входу для користувачів з базовими знаннями та дозволяє розпочати роботу практично з будь-якого пристрою, що має доступ до Інтернету. Інтерфейс платформи вирізняється інтуїтивною логікою побудови, що полегшує створення схем, підключення компонентів та написання й компіляцію програмного коду. Завдяки цьому розробник отримує можливість зосередитися на реалізації логіки проєкту, а не на технічних труднощах, пов'язаних із конфігурацією середовища.

Wokwi підтримує широкий перелік апаратних компонентів, серед яких - плати Arduino Uno, Nano, ESP32, а також численні периферійні модулі: датчики (температури, вологості, освітленості тощо), дисплеї (зокрема SSD1306 OLED), RTC-модулі (DS3231), Bluetooth-модулі (HC-06), а також базові елементи на кшталт резисторів, стабілітронів, конденсаторів і кнопок. Такий набір дозволяє реалізовувати як прості, так і складні апаратні конфігурації. У контексті симуляції датчика напруги ці можливості є особливо корисними, оскільки дозволяють відтворити роботу системи на всіх її рівнях - від зчитування аналогових значень до їхньої обробки, виводу на дисплей і передачі по бездротових каналах.

Окрему увагу варто звернути на підтримку бібліотек, таких як Adafruit_GFX, RTCLib, Wire, SoftwareSerial та інших. Це дозволяє писати програмний код з використанням звичних інструментів, не змінюючи його структури чи логіки. Таким чином, реалізований у Wokwi код з високим ступенем ймовірності залишатиметься сумісним із реальним фізичним пристроєм, що спрощує перехід від етапу симуляції до реального прототипу.

Платформа також підтримує редагування коду в реальному часі з можливістю миттєвого запуску симуляції. Це сприяє оперативному тестуванню

навіть незначних змін, виявленню помилок у логіці програми чи схемі, а також покращує ефективність процесу налагодження. Вбудований Serial Monitor дозволяє контролювати вихідні повідомлення та стан системи під час виконання коду, імітуючи реальні умови розробки. Крім того, наявність емуляції Bluetooth-з'єднання забезпечує базове тестування бездротової передачі даних, що актуально для багатьох сучасних проєктів.

Функція збереження проєктів за допомогою унікальних посилань значно полегшує співпрацю між учасниками команди, дозволяючи ділитися результатами роботи або створювати публічні демонстраційні приклади. Можливість експорту схем у форматі JSON-файлів слугує додатковим засобом документування та архівації проєктів, що важливо в контексті довготривалих досліджень або освітнього процесу.

Таким чином, Wokwi - це потужний і гнучкий інструмент для моделювання систем на базі мікроконтролерів, який дозволяє ефективно поєднувати апаратне проектування, програмування та тестування в одному середовищі. Його застосування у розробці та симуляції датчика напруги на базі Arduino підтверджує здатність інструменту забезпечити високий рівень деталізації, функціональності й інтеграції, що робить його доцільним вибором для як навчальних, так і інженерно-прикладних задач.

Однією з ключових переваг Wokwi є можливість імітувати поведінку реальних компонентів у віртуальному середовищі, що дозволяє тестувати схеми без фізичного обладнання. Наприклад, у нашому проєкті датчика напруги потенціометр у Wokwi замінює джерело напруги, а резистивний діляк із резисторів 100 кОм і 150 кОм коректно моделює зниження напруги для аналогового входу A0. OLED-дисплей, модуль DS3231 і HC-06 дозволяють перевірити виведення даних і бездротовий зв'язок, наближено до реальної системи. Це дає змогу протестувати логіку програми та з'єднання компонентів перед фізичною реалізацією, мінімізуючи ризик помилок. Інтерактивність Wokwi, зокрема можливість змінювати параметри, як-от повзунок потенціометра, і

спостерігати їхній вплив на виведення даних, сприяє глибшому розумінню роботи схеми. Експорт схем у графічні формати полегшує створення документації, а підтримка спільної роботи дозволяє кільком користувачам одночасно працювати над проектом, що корисно для командної розробки.

Проте Wokwi має обмеження, які можуть впливати на його використання. Основним недоліком є спрощена симуляція аналогових компонентів, таких як резистори, конденсатори чи стабілітрони. У нашій схемі конденсатор на 100 нФ і стабілітрон на 5.1 В додано для фільтрації шумів і захисту, але Wokwi не повною мірою відтворює їхню фізичну поведінку, зокрема вплив перешкод чи перенапруги, через віртуальну природу. Це ускладнює тестування реальних умов, де шуми чи нестабільність джерела відіграють роль. Wokwi не включає всі можливі компоненти, наприклад, високоточні АЦП типу ADS1256 чи модулі LoRa SX1278, що вимагає використання заміників або ручного налаштування через JSON. Зміна параметрів компонентів, як-от опір резисторів чи ємність конденсаторів, часто потребує редагування файлу `diagram.json`, що може бути незручним для користувачів без досвіду роботи з JSON.

Симуляція складних фізичних явищ, таких як теплові ефекти чи нелінійність компонентів, у Wokwi обмежена, що знижує точність моделювання аналогових систем. Наприклад, стабілітрон може некоректно імітувати обмеження напруги, що вводить у оману при тестуванні захисту. Залежність від інтернет-з'єднання є ще одним недоліком, оскільки Wokwi працює виключно онлайн, що може створювати проблеми в умовах обмеженого доступу до мережі. Безкоштовна версія має обмеження на кількість компонентів і складність проектів, а платна версія, хоча й розширює можливості, може бути недоступною для деяких користувачів. Точність таймінгів для модулів, таких як DS3231 RTC, не завжди відповідає реальній поведінці, що може викликати розбіжності. У нашому проекті це менш критично, оскільки основна логіка зчитування напруги, виведення на дисплей і через Bluetooth працює коректно, але для систем із високоточними вимогами до часу чи

аналогових сигналів можуть знадобитися альтернативні інструменти, наприклад Proteus.

Емуляція бездротових модулів, як HC-06, у Wokwi є спрощеною, і для перевірки реального Bluetooth-з'єднання потрібне фізичне тестування. Обмежена документація для деяких компонентів і необхідність ручного налаштування через JSON можуть ускладнити роботу для новачків. Незважаючи на це, Wokwi залишається потужним інструментом для швидкого прототипування Arduino-проектів, пропонуючи зручність, широку бібліотеку компонентів і підтримку симуляції в реальному часі. Для нашого датчика напруги він ефективно моделює основну функціональність, але для врахування реальних фізичних обмежень, таких як шуми чи перенапруги, рекомендується доповнити симуляцію фізичним тестуванням.

3.3 Структура коду та функціонал diagram.json

Файл diagram.json у симуляторі Wokwi є ключовим елементом для створення віртуальної схеми електронного проєкту, дозволяючи детально описати компоненти, їх розташування та електричні з'єднання для подальшої візуалізації та симуляції роботи системи. У контексті проєкту датчика напруги на базі Arduino Uno, цей файл визначає апаратну конфігурацію, включаючи плату Arduino Uno, потенціометр, який виконує роль резистивного дільника з резисторів 100 кОм і 150 кОм, OLED-дисплей SSD1306 із роздільною здатністю 128x64 пікселів і інтерфейсом I2C, модуль DS3231 RTC для роботи з реальним часом і модуль HC-06 Bluetooth для бездротової передачі даних. Він є структурованим JSON-об'єктом, що забезпечує точне відтворення схеми у браузері, дозволяючи Wokwi створювати графічне представлення компонентів, з'єднувати їх віртуальними дротами та моделювати їх взаємодію з кодом програми, наприклад, для зчитування аналогового сигналу з потенціометра через пін A0, виведення результатів на дисплей і передачі даних через Bluetooth. Оскільки Wokwi не підтримує стабілітрон

5.1 В і конденсатор 100 нФ як окремі компоненти, потенціометр замінює резистивний дільник, а захист і фільтрація враховуються лише в описі реальної схеми.

Файл `diagram.json` має кілька основних функцій. По-перше, він описує апаратні компоненти, включаючи їх типи, унікальні ідентифікатори та фізичне розташування на віртуальній робочій області Wokwi, що задається координатами у пікселях. По-друге, він визначає електричні з'єднання між пінами компонентів, включаючи колір проводів для візуальної ідентифікації та маршрути, які допомагають уникнути перехрещень і створюють акуратний вигляд схеми. По-третє, файл містить метадані, такі як версія формату, ім'я автора та інформація про редактор, що забезпечує сумісність із платформою Wokwi. Завдяки цій структурі Wokwi може точно відтворити схему, дозволяючи користувачу тестувати код, взаємодіяти з компонентами, наприклад, регулювати повзунок потенціометра для імітації зміни напруги, і спостерігати результати на дисплеї чи у віртуальному терміналі Bluetooth.

Структура файлу `diagram.json` поділена на кілька ключових полів. Перше поле – це версія формату файлу, зазвичай позначена як одиниця, що вказує на сумісність із поточною версією Wokwi. Друге поле містить ім'я автора, наприклад, Grok 3, яке ідентифікує творця схеми. Третє поле позначає редактор, який завжди є Wokwi, підтверджуючи, що схема створена для цієї платформи. Четверте поле – це масив компонентів, де кожен компонент описаний як об'єкт із кількома атрибутами. Кожен компонент має тип, що визначає, який елемент рендерити, наприклад, `wokwi-arduino-uno` для плати Arduino чи `wokwi-potentiometer` для потенціометра, унікальний ідентифікатор для посилань у з'єднаннях, координати розташування у вигляді значень `top` і `left` для розміщення на схемі та додаткові атрибути, які зазвичай залишаються порожніми. П'яте поле – це масив з'єднань, де кожне з'єднання описує зв'язок між двома пінами, колір дроту та маршрут, який задає шлях дроту через вертикальні та горизонтальні зсуви у пікселях.

У проєкті датчика напруги `diagram.json` описує конкретну конфігурацію. Arduino Uno розташована у центрі віртуальної схеми з координатами `top: 0, left: 0`, що робить її центральним елементом. Потенціометр розміщений ліворуч зверху з координатами `top: -100, left: -150`, що забезпечує зручний доступ до його повзунка для симуляції зміни напруги. Його сигнальний пін з'єднаний із аналоговим входом A0 Arduino, один крайній пін – із шиною 5V, а інший – із GND, імітуючи резистивний дільник із коефіцієнтом 0.4. OLED-дисплей SSD1306 розташований праворуч із координатами `top: -50, left: 200`, підключений до пінів A4 і A5 через I2C для виведення напруги та часу, а також до шин живлення 5V і GND. Модуль DS3231 RTC розміщений праворуч нижче дисплея з координатами `top: 40, left: 200`, і підключений паралельно до тих же I2C-пінів A4 і A5, а також до шин 5V і GND, забезпечуючи мітки часу. Модуль HC-06 Bluetooth розташований знизу Arduino з координатами `top: 200, left: 0`, з'єднаний із цифровими пінами 10 і 11 для передачі даних через SoftwareSerial, а також із шиною 3.3V і GND для живлення.

З'єднання у файлі `diagram.json` детально описують електричні зв'язки. Наприклад, сигнальний пін потенціометра `pot1:SIG` з'єднується з `uno:A0` зеленим дротом, що відображає передачу аналогового сигналу. Живлення потенціометра `pot1:VCC` підключене до `uno:5V` червоним дротом, а `pot1:GND` – до `uno:GND.1` чорним. Для OLED-дисплея `oled1:SDA` і `oled1:SCL` з'єднані з `uno:A4` і `uno:A5` синіми та жовтими дротами, відповідно, що відповідає I2C-протоколу. Аналогічно, DS3231 `rtc1:SDA` і `rtc1:SCL` підключені до тих же пінів A4 і A5 паралельно. HC-06 `bt1:TXD` і `bt1:RXD` з'єднані з `uno:10` і `uno:11` зеленим і жовтим дротами, а живлення `bt1:VCC` - із `uno:3.3V` червоним дротом. Маршрути дротів, наприклад, `["v-30", "h150"]`, задають шлях, щоб дроти виглядали акуратно і не перетиналися, де `v` означає вертикальний зсув, а `h` – горизонтальний.

Wokwi використовує `diagram.json` для кількох етапів роботи. На першому етапі файл завантажується під час відкриття проєкту, дозволяючи платформі зрозуміти, які компоненти потрібно відобразити. На другому етапі Wokwi рендерить графічну схему, розміщуючи компоненти за заданими координатами,

наприклад, Arduino у центрі, потенціометр ліворуч. На третьому етапі платформа малює дроти між пінами з урахуванням кольорів і маршрутів, створюючи візуально зрозумілу схему. Нарешті, Wokwi забезпечує електричний зв'язок між компонентами, дозволяючи коду sketch.ino взаємодіяти з ними, наприклад, зчитувати напругу з потенціометра, обробляти її та виводити на дисплей чи через Bluetooth. У нашому проєкті потенціометр імітує вихід резистивного дільника (0 – 5 В), що відповідає вхідній напрузі до 8.33 В, Arduino обробляє сигнал, DS3231 додає мітки часу, OLED відображає результати, а HC-06 передає дані у віртуальний термінал.

Файл diagram.json виконує роль вихідної конфігурації для створення віртуальної електронної схеми датчика напруги, що дає змогу здійснити попереднє тестування, верифікацію логіки роботи та оцінку функціональності проєкту в цифровому середовищі без потреби у використанні фізичних апаратних компонентів. Такий підхід дозволяє проводити моделювання на ранніх етапах розробки, мінімізуючи ризики, пов'язані з апаратними обмеженнями або помилками під час збирання схеми. Крім того, він значно знижує часові та матеріальні витрати, пов'язані з багаторазовим налагодженням прототипів.

На основі цієї віртуальної схеми створюється середовище, у якому можливо зчитувати аналогові вхідні сигнали, здійснювати їх обробку з урахуванням параметрів дільника напруги, виконувати прив'язку отриманих даних до поточного часу за допомогою RTC-модуля, а також виводити результати різними способами. Зокрема, дані можуть відображатися локально на OLED-дисплеї або передаватися на зовнішні пристрої через Bluetooth-з'єднання. Такий підхід реалізує багаторівневу обробку та передачу даних, що дозволяє відстежувати стан системи в режимі реального часу, оцінювати її стабільність і виявляти можливі відхилення в роботі.

Реалізація зазначеної віртуальної моделі дає змогу всебічно дослідити взаємодію програмного забезпечення з апаратною частиною системи. Це створює умови для глибшого розуміння функціональних принципів аналогової електроніки,

логіки побудови схем, а також особливостей програмування мікроконтролерів у контексті обробки реальних фізичних сигналів. Така форма симуляції є особливо корисною на етапі прототипування: вона дозволяє перевірити різні сценарії роботи, виявити помилки в логіці взаємодії компонентів і за потреби оперативно адаптувати або вдосконалити конфігурацію проєкту відповідно до нових технічних вимог чи умов експлуатації.

Таблиця 3.1 – Функціонал компонентів diagram.json

| Назва компонента | Атрибут компонента | Опис компонента |
|----------------------|--|--|
| Arduino Uno | type: wokwi-arduino-uno, id: uno | Основна мікроконтролерна плата для обробки сигналів, виконання коду та керування модулями. |
| Потенціометр | type: wokwi-potentiometer, id: pot1 | Імітує резистивний дільник (100 кОм і 150 кОм), видає аналоговий сигнал 0–5 В на пін A0. |
| OLED-дисплей SSD1306 | type: wokwi-ssd1306, id: oled1 | Дисплей 0.96" (128x64, I2C) для локального виведення напруги та міток часу. |
| DS3231 RTC | type: wokwi-ds3231, id: rtc1 | Модуль Bluetooth для бездротової передачі даних напруги та часу на зовнішні пристрої. |
| HC-06 Bluetooth | type: wokwi-hc06, id: bt1 | Модуль Bluetooth для бездротової передачі даних напруги та часу на зовнішні пристрої. |

3.4 Структура коду та функціонал sketch.ino

Код sketch.ino забезпечує функціонування датчика напруги на базі плати Arduino Uno в симуляторі Wokwi, виконуючи зчитування аналогових даних із потенціометра, який моделює резистивний дільник із резисторів 100 кОм і 150 кОм із коефіцієнтом ділення 0.4, обробляючи отримані значення, додаючи мітки часу від модуля DS3231 RTC і виводячи результати кількома способами: на OLED-дисплей SSD1306 із роздільною здатністю 128x64 пікселів через інтерфейс I2C, у Serial Monitor для відлагодження та через модуль HC-06 Bluetooth для бездротової передачі на зовнішні пристрої, такі як смартфон чи комп'ютер. Цей код розроблено відповідно до завдання кваліфікаційної роботи, що передбачає створення датчика напруги, здатного вимірювати напругу в діапазоні до 8.33 В, забезпечуючи достатню точність завдяки усередненню зчитувань, захист у реальній схемі через стабілітрон і конденсатор, а також можливість масштабування для інтеграції в складніші системи, наприклад, у підсистему керування мікрокліматом для «Розумного будинку».

Робота коду починається з підключення необхідних бібліотек. Бібліотека Wire забезпечує зв'язок через I2C-протокол для комунікації з дисплеєм і RTC-модулем. Бібліотеки Adafruit_GFX і Adafruit_SSD1306 дозволяють керувати OLED-дисплеєм, створюючи графічний інтерфейс для відображення тексту, такого як значення напруги та мітки часу. RTCLib відповідає за взаємодію з модулем DS3231, забезпечуючи доступ до даних реального часу. SoftwareSerial створює віртуальний послідовний порт на цифрових пінах 10 і 11 для зв'язку з модулем HC-06 Bluetooth, дозволяючи передавати дані без використання апаратного UART Arduino. Далі визначаються константи: SCREEN_WIDTH і SCREEN_HEIGHT задають роздільну здатність дисплея 128x64 пікселів, а OLED_ADDRESS вказує I2C-адресу дисплея, зазвичай 0x3C, хоча в деяких модулях вона може бути 0x3D. Створюються об'єкти для роботи з апаратними модулями: display для дисплея, rtc

для DS3231 і bluetooth для HC-06, із зазначенням пінів 10 (RX) і 11 (TX) для SoftwareSerial.

Функція `setup` виконує ініціалізацію системи під час запуску Arduino. Спочатку відкривається послідовний порт зі швидкістю 38400 бод для виведення даних у Serial Monitor, що використовується для відлагодження та перевірки роботи програми. Аналогічно ініціалізується Bluetooth-зв'язок із тією ж швидкістю, що дозволяє передавати дані у віртуальний термінал Wokwi або на реальний пристрій. Шина I2C активується через `Wire.begin` для забезпечення зв'язку з дисплеєм і RTC. Далі перевіряється ініціалізація OLED-дисплея за допомогою методу `display.begin`, який активує внутрішнє джерело живлення дисплея та встановлює I2C-адресу. Якщо ініціалізація не вдала, наприклад, через неправильну адресу чи відсутність зв'язку, у Serial Monitor виводиться повідомлення `OLED initialization failed`, і програма зупиняється у нескінченному циклі. Подібна перевірка виконується для DS3231: якщо модуль не відповідає, виводиться повідомлення `RTC initialization failed` із зупинкою програми. Якщо RTC втратив живлення, наприклад, через відключення батареї, час автоматично синхронізується з моментом компіляції коду за допомогою макросів `DATE` і `TIME`. На завершення функція `setup` очищає дисплей, встановлює розмір шрифту 1, колір тексту білий, позицію курсора у верхньому лівому куті, виводить повідомлення `Voltage Monitor` і відображає його на екрані протягом двох секунд, сигналізуючи про готовність системи.

Функція `loop` є основним циклом програми, який виконується постійно після ініціалізації. Вона відповідає за зчитування, обробку та виведення даних. Спочатку визначається константа `samples` зі значенням 10, яка вказує кількість зразків для усереднення аналогових зчитувань із піна A0. У циклі `for` зчитується 10 значень із аналогового входу A0 за допомогою `analogRead`, із затримкою 10 мс між кожним зчитуванням, що дозволяє стабілізувати сигнал і зменшити вплив випадкових шумів. Отримані значення сумуються в змінну `sum`, після чого обчислюється середнє значення `sensorValue` шляхом ділення суми на кількість зразків. Це середнє

значення конвертується у вхідну напругу за формулою $\text{voltage} = \text{sensorValue} * (5.0 / 1023.0) / 0.4$, де $5.0 / 1023.0$ враховує 10-бітну роздільність АЦП Arduino, що відповідає діапазону 0–5 В, а ділення на 0.4 компенсує коефіцієнт резистивного дільника, дозволяючи обчислити вхідну напругу до 8.33 В. Поточний час отримується від DS3231 через об'єкт `rtc` методом `now`, який повертає структуру `DateTime` із полями року, місяця, дня, години, хвилини та секунди. Ці дані форматуються у рядок `timestamp` у форматі `YYYY/MM/DD HH:MM:SS`, де для однозначних значень місяця, дня, години, хвилин і секунд додаються провідні нулі за допомогою умовного оператора `?:`, наприклад, 6 місяць форматується як `06`.

Надалі результати обробки даних передаються трьома різними каналами. На OLED-дисплеї попередній вміст очищається за допомогою методу `clearDisplay`, після чого курсор позиціонується в координатах `(0, 0)`, де виводиться текст "Voltage:" через команду `print`, а значення напруги з двома десятковими знаками відображається за допомогою `print(voltage, 2)`, завершуючись одиницею "V" через `println`, що здійснює перехід на новий рядок. Наступний рядок заповнюється міткою часу, сформованою командою `print(timestamp)`, а оновлення екрана виконується викликом `display`. У Serial Monitor подається схожа інформація в структурованому вигляді, наприклад, "Voltage: X.XX V, Time: YYYY/MM/DD HH:MM:SS", де операції `print` і `println` забезпечують послідовне додавання тексту та роздільники рядків, сприяючи зрозумілому логу для цілей налагодження. Через модуль Bluetooth дані передаються у форматі "V=X.XX,T=YYYY/MM/DD HH:MM:SS", де методи `bluetooth.print` і `bluetooth.println` формують відповідний рядок, який у симуляторі Wokwi відображається у віртуальному терміналі HC-06. Процес завершується затримкою 600 мс, реалізованою через команду `delay(600)`, що забезпечує оновлення даних приблизно кожні 0,4 секунди, підтримуючи баланс між оперативністю відображення та стабільністю роботи системи

На що варто звернути увагу при роботі з кодом. Точність вимірювань обмежена 10-бітною роздільністю АЦП Arduino, яка дає крок приблизно 4.88 мВ, що може бути недостатньо для високоточних застосувань, особливо при малих

змінах напруги. Для підвищення точності можна збільшити кількість зразків у усередненні або використати зовнішній АЦП, наприклад, ADS1115. У реальній схемі стабілітрон 5.1 В і конденсатор 100 нФ захищають пін A0 від перенапруги та фільтрують шуми, але в Wokwi ці компоненти не моделюються, і потенціометр видає ідеальний сигнал, що спрощує симуляцію, але не відображає реальних умов. Коефіцієнт дільника 0.4, який відповідає резисторам 100 кОм і 150 кОм, потрібно перевірити під час калібрування, підключивши відомі напруги, наприклад, 3 В на потенціометрі, що має відповідати 5 В після конвертації, і за потреби додати коригувальний множник у коді, наприклад, `voltage *= 1.02`. Ініціалізація дисплея і RTC є критичною, і помилки, викликані неправильною адресою I2C чи відсутністю зв'язку, зупинять програму, тому варто перевірити апаратне підключення чи адреса дисплея, яка може бути 0x3D у деяких модулях. Форматування мітки часу з провідними нулями через об'єкт String забезпечує читабельність, але займає додаткову оперативну пам'ять, що може стати проблемою при додаванні складних функцій, таких як логування чи IoT. У таких випадках можна оптимізувати код, використовуючи char-масиви замість String. Частота оновлення 600 мс підходить для моніторингу напруги, але її можна зменшити до 100 мс для швидшого відгуку або збільшити до 1000 мс для економії ресурсів у повільних системах. У Wokwi HC-06 виводить дані у віртуальний термінал, але в реальній схемі потрібно перевірити швидкість зв'язку 38400 бод і підключити дільник напруги для пина RXD HC-06, оскільки Arduino видає 5 В, а модуль зазвичай працює на 3.3 В. Код є модульним і дозволяє масштабування, наприклад, додавання SD-карти для збереження даних чи заміну Arduino на ESP32 для передачі даних через Wi-Fi, але це вимагатиме додаткових бібліотек, пінів і пам'яті. Варто протестувати код із різними значеннями напруги, задокументувати результати, наприклад, скріншоти дисплея чи логу Serial Monitor, і зазначити обмеження симуляції в Wokwi порівняно з реальною схемою.

3.5 Тестування проекту в емуляторі Wokwi

На представленій схемі (Рис. 3.1) відображено апаратну конфігурацію, яка складається з плати Arduino Uno, OLED-дисплея SSD1306 та датчика, ймовірно, модуля типу BMP180 чи іншого аналогічного I2C-пристрою, призначеного для моделювання змін напруги. Схема ілюструє електричні зв'язки між компонентами через кольорові провідники, що позначають лінії живлення, заземлення та сигнальних каналів. У симуляційному середовищі, такому як Wokwi, користувач має можливість взаємодіяти з окремими елементами, зокрема впливати на параметри системи через імітацію коливань напруги, що дає змогу оцінити її працездатність виходячи з типових характеристик подібних платформ. Оскільки схема має статичний характер, функціональні можливості взаємодії визначаються стандартними інструментами симуляції, де користувач може коригувати вхідні значення, пов'язані з напругою, та аналізувати відповідні зміни.

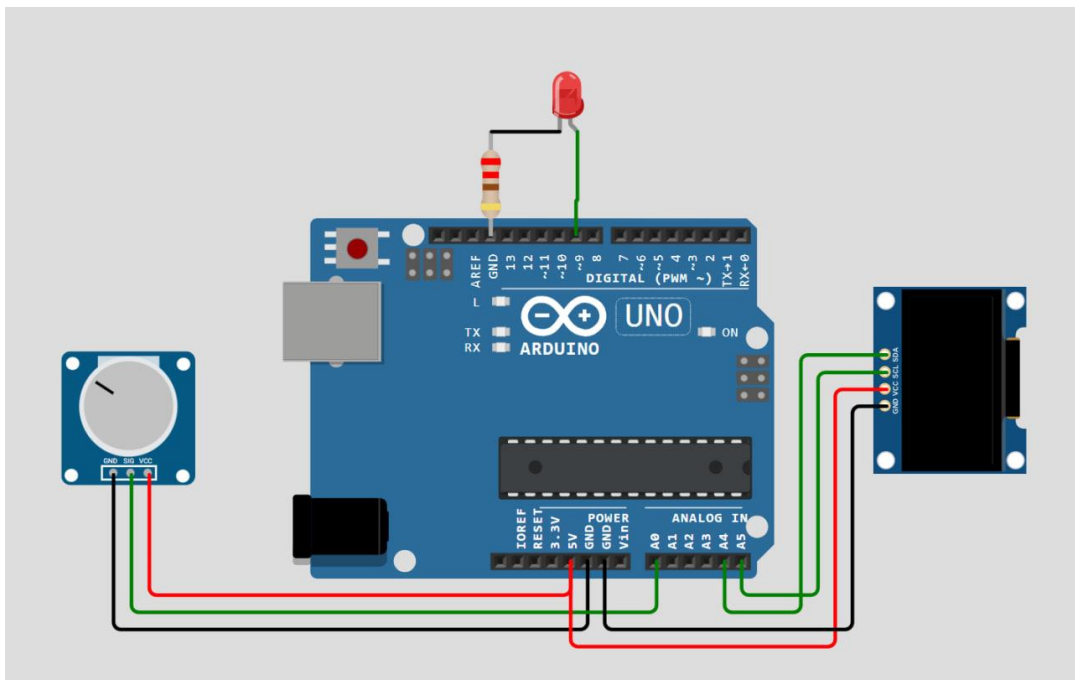


Рисунок 3.1 – Датчик напруги в емуляторі

Першим елементом, з яким можлива взаємодія, є потенціометр (Рис. 3.2), розміщений у лівій частині схеми. Його ідентифікують три провідні з'єднання:

зелений дріт підключений до аналогового входу A0 плати Arduino Uno, червоний до шини живлення 5V, а чорний до заземлення GND. У симуляційному середовищі цей модуль відтворює характеристики реального джерела сигналу, дозволяючи змінювати вихідний сигнал через інтерфейс симулятора. Зокрема, користувач може регулювати аналогове значення, яке надходить на пін A0, у межах від 0 до 1023, імітуючи коливання напруги. Така взаємодія дає змогу проаналізувати реакцію системи на різні входні параметри, відображаючи їх на дисплеї чи у виводі даних, що сприяє оцінці точності та чутливості апаратного забезпечення.

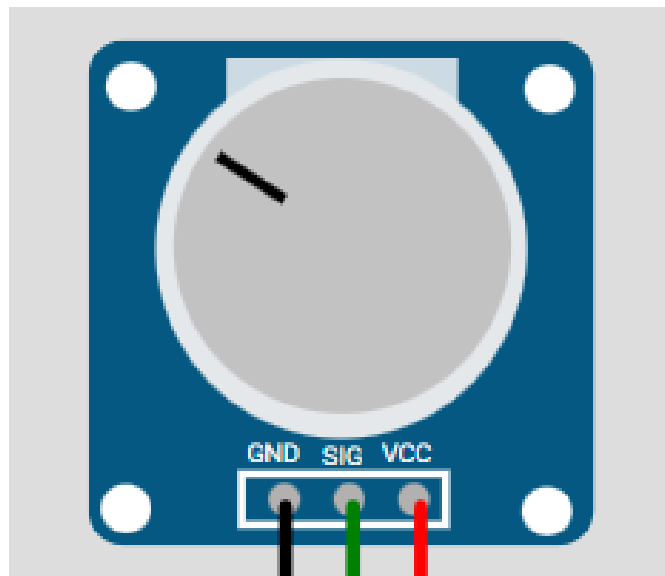


Рисунок 3.2 – Потенціометр

Другим компонентом є OLED-дисплей SSD1306 (Рис. 3.3), розміщений праворуч від плати. Його з'єднання з Arduino Uno здійснюється через інтерфейс I2C, де зелений дріт підключається до пін A4 (SDA), червоний - до шини живлення 5V, чорний - до заземлення GND, а пін A5 (SCL), хоч і не відображений на схемі, слугує стандартним каналом для цього протоколу. У симуляційному середовищі Wokwi дисплей відображає текстові дані, зокрема значення напруги, наприклад, "Voltage: 5.00 V", що формується внаслідок обробки сигналів програмним забезпеченням. Пряма взаємодія з дисплеєм у цьому середовищі не передбачена, однак користувач має можливість спостерігати за оновленням інформації в

реальному часі залежно від змін, які вводяться через потенціометр. Такий метод аналізу сприяє оцінці коректності форматування тексту, вибору розміру шрифту та розподілу вмісту на екрані з роздільною здатністю 128x64 пікселі, що має важливе значення для визначення зручності користування інтерфейсом. Вивчення даних, відображених на дисплеї, дає змогу не лише перевірити точність роботи програмного коду, але й оцінити раціональність використання обмеженого простору для передачі результатів.

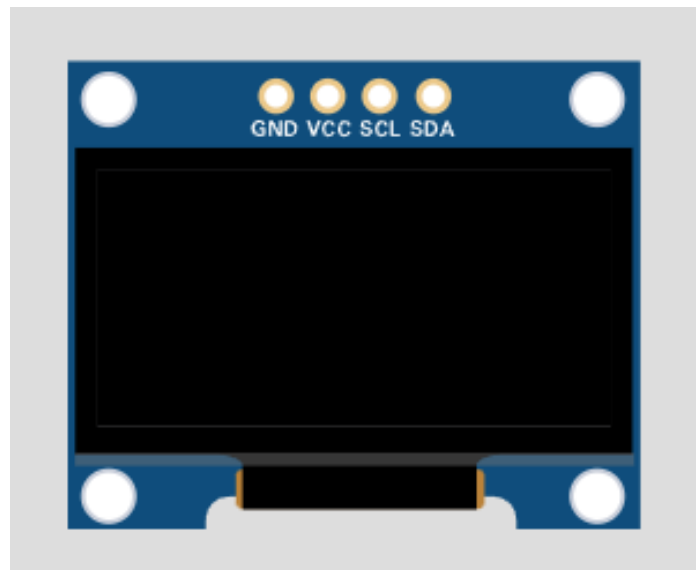


Рисунок 3.3 – OLED-дисплей

Третім елементом для непрямой взаємодії є плата Arduino Uno (Рис. 3.4), що виконує роль центрального процесорного вузла в описаній схемі. Вона з'єднана з потенціометром через пін A0 для зчитування аналогових сигналів, які імітують зміну напруги, а також підключена до дисплея через I2C-лінії для виведення результатів. У симуляційному середовищі пряма взаємодія з платою не передбачена, оскільки її функціональність реалізується шляхом виконання завантаженого програмного коду. Користувач має можливість оцінити працездатність Arduino, аналізуючи обробку сигналів від потенціометра та передачу даних на дисплей, що дозволяє перевірити логіку програмного забезпечення, зокрема алгоритми обчислення напруги чи перетворення даних, що

є суттєвим для гарантування надійності системи. Такий непрямий підхід сприяє оцінці ефективності обробки даних у реальному часі, що відіграє ключову роль у визначенні стабільності роботи. Гнучкість програмного коду дає змогу інтегрувати до платформи додаткові модулі, такі як SD-карта чи інші сенсори, розширюючи її функціональні можливості. Застосування симуляційного середовища полегшує виявлення потенційних недоліків у логіці програми до переходу до апаратної реалізації, що сприяє економії часу та ресурсів. Крім того, платформа підтримує коригування параметрів через оновлення програмного забезпечення, підвищуючи її адаптивність до змінених умов.

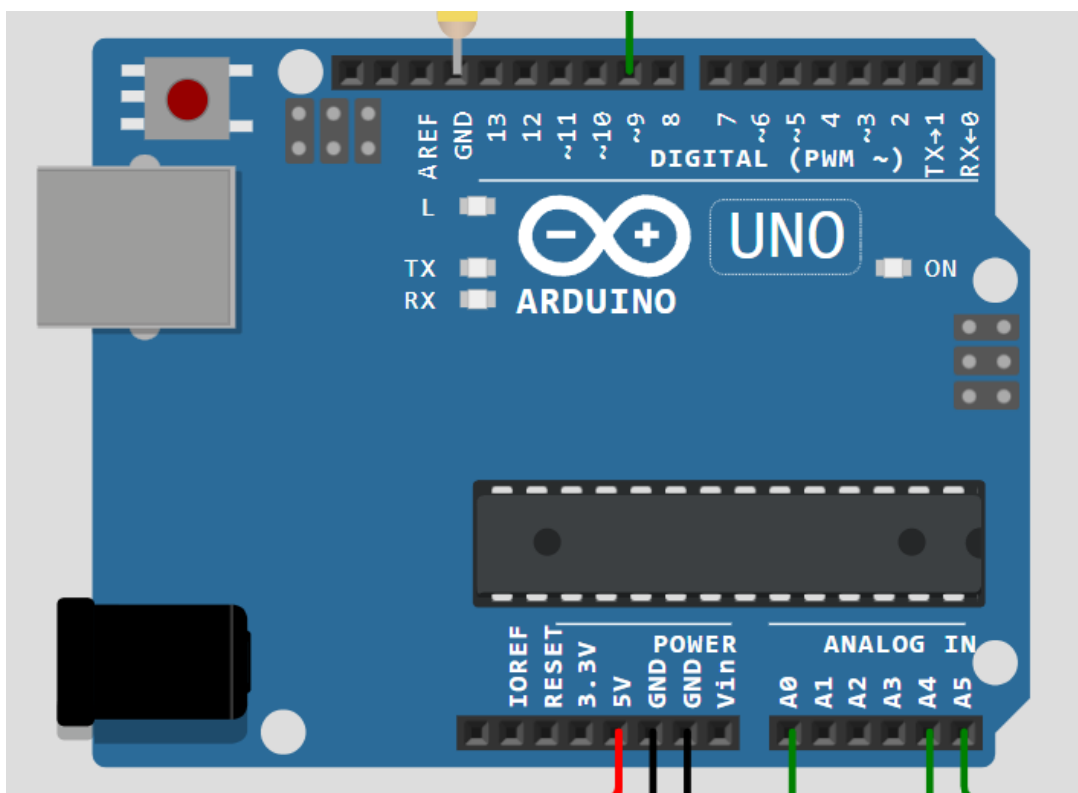


Рисунок 3.4 – Плата Arduino Uno

У запропонованій схемі реалізована можливість активної взаємодії з датчиком, що забезпечує зміну вхідних параметрів системи та дозволяє безпосередньо впливати на перебіг її роботи в реальному часі. Така можливість відкриває умови для більш гнучкого налаштування та динамічного дослідження реакції системи на змінні вхідні сигнали, що є особливо важливим у процесі

налагодження та тестування алгоритмів її функціонування. При цьому користувач має змогу змінювати параметри не лише на рівні програмного забезпечення, а й фізично - через відповідні сенсорні входи, що значно розширює спектр застосування схеми.

Паралельно з цим відбувається постійне оновлення інформації на OLED-дисплеї, який виконує функцію основного засобу візуалізації. Завдяки наочному відображенню результатів обробки даних користувач отримує можливість оперативно відстежувати поточний стан системи та її реакцію на зміни. Це, у свою чергу, значно спрощує процес верифікації правильності реалізації програмної логіки, оскільки всі ключові параметри можуть бути негайно оцінені без необхідності додаткових інструментів моніторингу.

Крім прямої візуалізації, передбачено також можливість непрямої оцінки роботи плати Arduino шляхом аналізу поведінки зовнішніх модулів, підключених до неї. Серед таких елементів можуть бути як виконавчі пристрої (наприклад, реле, двигуни чи світлодіоди), так і додаткові модулі збору даних або виводу. Поведінка цих пристроїв, у свою чергу, відображає ефективність виконання програмного коду на мікроконтролері. Завдяки такому підходу можна вчасно виявити збої або логічні помилки, що виникають у процесі обробки сигналів, без застосування складних діагностичних засобів.

У цілому така конфігурація формує логічно узгоджену, функціонально завершену систему, у якій кожен компонент відіграє чітко визначену роль. Датчик виступає джерелом первинної інформації, яка підлягає обробці; мікроконтролер Arduino здійснює аналіз отриманих вхідних даних, а OLED-дисплей забезпечує інформативне виведення результатів, доступне для користувача. Усі ці елементи діють у тісній взаємодії, створюючи умови для тестування системи в умовах симуляційного середовища. Такий підхід дає змогу здійснити комплексну перевірку працездатності розробленої логіки, а також оцінити ступінь її відповідності попередньо сформульованим функціональним і технічним вимогам.

3.6 Висновок до третього розділу

Огляд модулів та реалізація датчика напруги продемонстрували ефективність використання плати Arduino Uno як центрального процесорного вузла, потенціометра як імітації резистивного дільника з коефіцієнтом 0.4, OLED-дисплея SSD1306 для візуалізації даних, модуля DS3231 RTC для міток часу та модуля HC-06 Bluetooth для бездротової передачі. Ця комбінація дозволила реалізувати систему моніторингу напруги в діапазоні до 8.33 В, забезпечуючи обробку аналогових сигналів, додавання часової інформації та виведення результатів у кількох каналах. Обмеження, пов'язані з відсутністю стабілітрона 5.1 В і конденсатора 100 нФ у симуляції, не вплинули на базову функціональність, але підкреслюють необхідність їх інтеграції в реальній апаратній реалізації для захисту та фільтрації шумів.

Розробка засобу датчика напруги в емуляторі Wokwi довела практичність використання цього інструменту для моделювання електронних систем. Завдяки налаштуванню компонентів і з'єднань у файлі diagram.json вдалося відтворити схему, що точно імітує поведінку датчика, дозволяючи перевіряти логіку роботи без фізичних пристроїв. Симуляція показала коректність взаємодії між модулями, зокрема зчитування напруги з потенціометра та її відображення, що підтверджує доцільність Wokwi як платформи для попереднього тестування та оптимізації проєктів перед їх апаратною реалізацією.

Структура коду та функціонал diagram.json забезпечили чітке визначення компонентів і їх з'єднань, що є основою для візуалізації та симуляції схеми. Файл визначив типи модулів, їх ідентифікатори, координати розміщення та маршрути проводів, що дозволило Wokwi точно відтворити апаратну конфігурацію. Ця структура полегшила аналіз електричних зв'язків і дала змогу ефективно інтегрувати код із апаратними елементами, підкреслюючи важливість детального опису для забезпечення сумісності та стабільності системи в симуляційному середовищі.

Структура коду та функціонал файлу sketch.ino забезпечили реалізацію основного алгоритму обробки даних, який охоплює зчитування 10 зразків із аналогового входу A0, обчислення їхнього середнього значення для зменшення впливу шумів, перетворення отриманих даних у значення напруги з урахуванням коефіцієнта резистивного дільника, форматування міток часу, отриманих від модуля RTC, а також виведення результатів на OLED-дисплей, у Serial Monitor і через Bluetooth-модуль. Цей код виявив ознаки модульності та гнучкості, надаючи можливість проводити калібрування через введення коригувальних множників і адаптувати систему до додаткових функцій, таких як збереження даних у логах чи інтеграція з технологіями Інтернету речей (IoT). Водночас обмеження, зумовлені 10-бітною роздільною здатністю аналого-цифрового перетворювача (АЦП) і витратами пам'яті на форматування текстових рядків, підкреслюють потребу в подальшій оптимізації для забезпечення ефективності при виконанні більш складних завдань.

Тестування проєкту в емуляторі Wokwi засвідчило ефективну працездатність системи, продемонструвавши стабільне зчитування напруги в діапазоні від 0 до 8.33 В залежно від положення повзунка потенціометра, точне оновлення міток часу з урахуванням їхньої послідовності та синхронізоване виведення даних через різні канали. Взаємодія з потенціометром, модулем DS3231 для керування часом і модулем HC-06 для передачі даних забезпечила можливість оцінити динамічні зміни параметрів і процес передачі інформації, тоді як OLED-дисплей і Serial Monitor чітко відобразили результати без виявлення будь-яких збоїв. Проте обмеження симуляційного середовища, зокрема відсутність моделювання шумів і реального з'єднання Bluetooth, підкреслили необхідність проведення апаратного тестування для всебічної оцінки функціональності та надійності системи в реальних умовах.

ВИСНОВКИ

У рамках цієї дипломної роботи було успішно вирішено завдання розробки та тестування датчика напруги на базі мікроконтролера Arduino, що підтверджує досягнення поставленої мети.

У першому розділі було проведено всебічний аналіз існуючих аналогів, методів вимірювання напруги та сучасних технічних засобів, включаючи різні типи датчиків напруги, аналого-цифрові перетворювачі (АЦП), мікроконтролери та технології бездротової передачі даних. Виявлено, що Arduino є оптимальною платформою для даної розробки завдяки своїй доступності, гнучкості та широким можливостям інтеграції. Обґрунтовано вибір компонентів, таких як дільник напруги, RTC-модуль DS3231 для точних міток часу, HC-06 для Bluetooth-зв'язку та OLED-дисплей для візуалізації даних, що забезпечує необхідну функціональність та точність вимірювання.

Другий розділ присвячений розробці архітектури системи та її програмного забезпечення. Детально описано функціональну схему датчика, що включає схему підключення дільника напруги, реалізацію аналогового входу та інтеграцію периферійних модулів. Розроблено структурну схему програмного забезпечення, яка передбачає модульний підхід до обробки даних, включаючи ініціалізацію, зчитування, обробку, виведення та передачу інформації. Ключовим етапом стала розробка алгоритму зчитування аналогових даних, їх усереднення для підвищення точності, перетворення в значення напруги та додавання часових міток, що є фундаментом для коректної роботи системи.

У третьому розділі було здійснено практичну реалізацію розробленого програмного забезпечення та проведено його тестування. Код прошивки `sketch.ino` успішно реалізував основний алгоритм, включаючи ефективне зчитування та усереднення 10 зразків з аналогового входу A0, конвертацію у напругу з урахуванням коефіцієнта дільника, форматування міток часу від RTC та виведення результатів на OLED-дисплей, Serial Monitor та Bluetooth. Виявлена модульність та

гнучкість коду дозволяють легко калібрувати систему та адаптувати її під додаткові функції. Тестування проєкту в емуляторі Wokwi підтвердило працездатність системи, продемонструвавши стабільне зчитування напруги (від 0 до 8.33 В) залежно від положення потенціометра, коректне оновлення міток часу та синхронізоване виведення даних на всі інтерфейси. Взаємодія з потенціометром, DS3231 та HC-06 дозволила оцінити динамічну зміну параметрів та передачу інформації, а дисплей та Serial Monitor безпомилково відобразили результати. Хоча симуляція має певні обмеження (відсутність реальних шумів та фізичного Bluetooth-з'єднання), вона чітко підтвердила коректність логіки та функціоналу розробленого пристрою, вказавши на перспективність подальшого апаратного тестування.

Таким чином, розроблений датчик напруги на базі мікроконтролера Arduino є функціональним та ефективним рішенням для моніторингу електричних параметрів, демонструючи потенціал для широкого застосування в автоматизованих системах контролю та збору даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Орлов С. А. Мікроконтролери та їх програмування. Київ: Вища школа, 2017. 284 с.
2. Monk S. 30 Arduino Projects for the Evil Genius. McGraw-Hill Education, 2016. 345 с.
3. Маслюк М. А. Програмування мікроконтролерів Arduino та Raspberry Pi. Київ: ВПЦ "Київський університет", 2018. 238 с.
4. Артюхов І. І. Основи електроніки та мікропроцесорної техніки. Дніпро: ДНУ, 2019. 311 с.
5. Кучинський П. А. Датчики та перетворювачі інформації. Львів: Видавництво Львівської політехніки, 2020. 254 р.
6. Башков Є. В. Сучасні мікроконтролери та їх застосування. Одеса: Фенікс, 2017. 218 с.
7. Horowitz P., & Hill W. (2015). The Art of Electronics. Cambridge University Press. 1246 p.
8. Шпак Р. А. Arduino. Довідник програміста. Харків: Моноліт. Програмування мікроконтролерів Arduino. Київ: Технологічна школа, 2018. 247 с.
9. Светлов Ю. Основи електроніки та схемотехніки для інженерів. Львів: Експерт, 2019. 328 с.
10. Іванов О. В., Петренко С. Р. Розробка та тестування інтелектуального датчика напруги на базі Arduino. Вісник Хмельницького національного університету. Серія: Технічні науки. 2020 (3). С. 125-132.
11. Офіційна документація Arduino. Arduino.cc. URL: <https://www.arduino.cc/> (дата звернення: 11.05.2025).
12. SparkFun Electronics. Voltage Sensor Hookup Guide. URL: <https://learn.sparkfun.com/tutorials/voltage-sensor-hookup-guide/> (дата звернення: 11.05.2025).
13. Adafruit Learning System. Analog Input with Arduino. URL: <https://learn.adafruit.com/analog-input-with-arduino/> (дата звернення: 14.05.2025).

14. Instructables. *DIY* Arduino Voltage Sensor. URL: <https://www.instructables.com/DIY-Arduino-Voltage-Sensor/> (дата звернення: 14.05.2025).

15. ElectroPeak. Arduino Voltage Sensor Tutorial. URL: <https://electropeak.com/blog/arduino-voltage-sensor-tutorial/> (дата звернення: 15.05.2025).

16. DroneBot Workshop. Arduino Voltage Measurement. URL: <https://dronebotworkshop.com/arduino-voltage-measurement/> (дата звернення: 11.05.2025).

17. All About Circuits. Voltage Dividers Explained. URL: <https://www.allaboutcircuits.com/textbook/direct-current/chpt-5/voltage-divider-circuits/> (дата звернення: 02.05.2025).

18. Last Minute Engineers. Arduino ADC Explained. URL: <https://lastminuteengineers.com/arduino-adc-tutorial/> (дата звернення: 02.05.2025).

19. PJRC. Teensy and Arduino ADC Resolution. URL: https://www.pjrc.com/teensy/td_libs_AnalogInput.html (дата звернення: 05.05.2025).

20. *DS3231* Real Time Clock Module with Arduino. URL: <https://learn.microgen.com/learn/ds3231-real-time-clock-module-with-arduino/> (дата звернення: 19.05.2025).

21. Circuit Basics. Arduino OLED Display Tutorial. URL: <https://www.circuitbasics.com/arduino-oled-display-tutorial/> (дата звернення: 05.05.2025).

22. Wokwi. Online Arduino Simulator. URL: <https://wokwi.com/> (дата звернення: 17.05.2025).

23. Stack Overflow. Arduino Voltage Divider Calculation. URL: <https://stackoverflow.com/questions/example-arduino-voltage-divider> (дата звернення: 01.05.2025).

24. GitHub. Arduino Libraries for Sensors. URL: <https://github.com/arduino-libraries> (дата звернення: 15.05.2025).

25. The Engineering Projects. Arduino Voltage Measurement using ACS712. URL: <https://www.theengineeringprojects.com/2020/12/arduino-voltage-measurement-using-ac712.html> (дата звернення: 18.05.2025).

26. Hackster.io. Arduino based Smart Energy Meter. URL: <https://www.hackster.io/arduino/arduino-based-smart-energy-meter-9b0d2a> (дата звернення: 13.05.2025).

27. Towards Data Science. Introduction to Analog-to-Digital Converters (ADCs). URL: <https://towardsdatascience.com/introduction-to-analog-to-digital-converters-adcs-891c9f7b1e4f> (дата звернення: 10.05.2025).

28. Seeed Studio Wiki. Grove – Voltage Divider. URL: https://wiki.seeedstudio.com/Grove-Voltage_Divider/ (дата звернення: 03.05.2025).

29. Embedded.com. Understanding ADC Resolution and Accuracy. URL: <https://www.embedded.com/understanding-adc-resolution-and-accuracy/> (дата звернення: 19.05.2025).

30. Medium. Bluetooth Low Energy (BLE) with Arduino. URL: <https://medium.com/example-ble-arduino> (дата звернення: 19.05.2025).

31. ScienceDirect. Journals on Measurement Systems and Microcontrollers. URL: <https://www.sciencedirect.com/> (дата звернення: 06.05.2025).

32. IEEE Xplore. Conference Papers on Arduino Applications. URL: <https://ieeexplore.ieee.org/> (дата звернення: 08.05.2025).

33. National Instruments. Fundamentals of Data Acquisition. URL: <https://www.ni.com/en-us/innovations/white-papers/10/fundamentals-of-data-acquisition.html> (дата звернення: 05.05.2025).

34. Learn Robotics. How to Read Analog Sensors with Arduino. URL: <https://www.learnrobotics.org/blog/read-analog-sensors-arduino/> (дата звернення: 08.05.2025).

35. Circuit Fun. Arduino Voltage Measurement Tutorial with Code. URL: <https://circuit.fun/arduino-voltage-measurement-tutorial-with-code/> (дата звернення: 09.05.2025).

36. DIY Electronics. Arduino Current and Voltage Sensor. URL: <https://diyelectronics.com/arduino-current-voltage-sensor/> (дата звернення: 01.05.2025).

37. Tutorials Point. Arduino – Analog Read. URL: https://www.tutorialspoint.com/arduino/arduino_analog_read.htm (дата звернення: 14.05.2025).

38. Programmer's Ranch. Arduino Data Logging to SD Card. URL: <https://www.programmersranch.com/arduino-data-logging-to-sd-card/> (дата звернення: 17.05.2025).

39. Geek Gurl Diaries. Arduino OLED Display SSD1306 Tutorial. URL: <https://www.geekgurldiaries.com/arduino-oled-display-ssd1306-tutorial/> (дата звернення: 08.05.2025).

40. Arduino Serial. URL: <https://www.martyncurrey.com/arduino-serial-a-look-at-the-different-serial-libraries/> (дата звернення: 02.05.2025).

41. Стабілізатор. URL: <https://car-led.org/ua/p1685699174-ks170a-stabilitron-643.html> (дата звернення: 03.05.2025).

42. Швидкодіючий TVS-діод. URL: <https://radio-market.biz/ua/p1288625077-diod-ultra-bystrodejstvuyuschij.html> (дата звернення: 10.05.2025).

43. Резистор. URL: <https://tempus.prom.ua/ua/p1599011078-rezistor-2vt-100.html> (дата звернення: 02.05.2025).

44. Конденсатор керамічний. URL: <https://www.rcscomponents.kiev.ua/catalog/vysokovoltni-keramichni-kondensatory/84> (дата звернення: 02.05.2025).

45. Макетна плата. URL: <https://arduino.ua/prod218-maketnaya-plata-besraechnaya-mb-102-400-tochek> (дата звернення: 03.05.2025).

46. З'єднувальні дроти типу Male-to-Male. URL: <https://a-radio.com.ua/soedinitelnye-provoda/> (дата звернення: 09.05.2025).

47. Батарея 9 В із конектором. URL: <https://nocher-techn.in.ua/perehodnik-adapter-pitaniya-pod-batareyku-krona-9v-s-konnektorom-55mm-arduino> (дата звернення: 07.05.2025).

48. OLED-дисплей 0.96". URL: <https://www.mini-tech.com.ua/oled-spi-0-96-128x64-graficheskij-displej-white> (дата звернення: 07.05.2025).

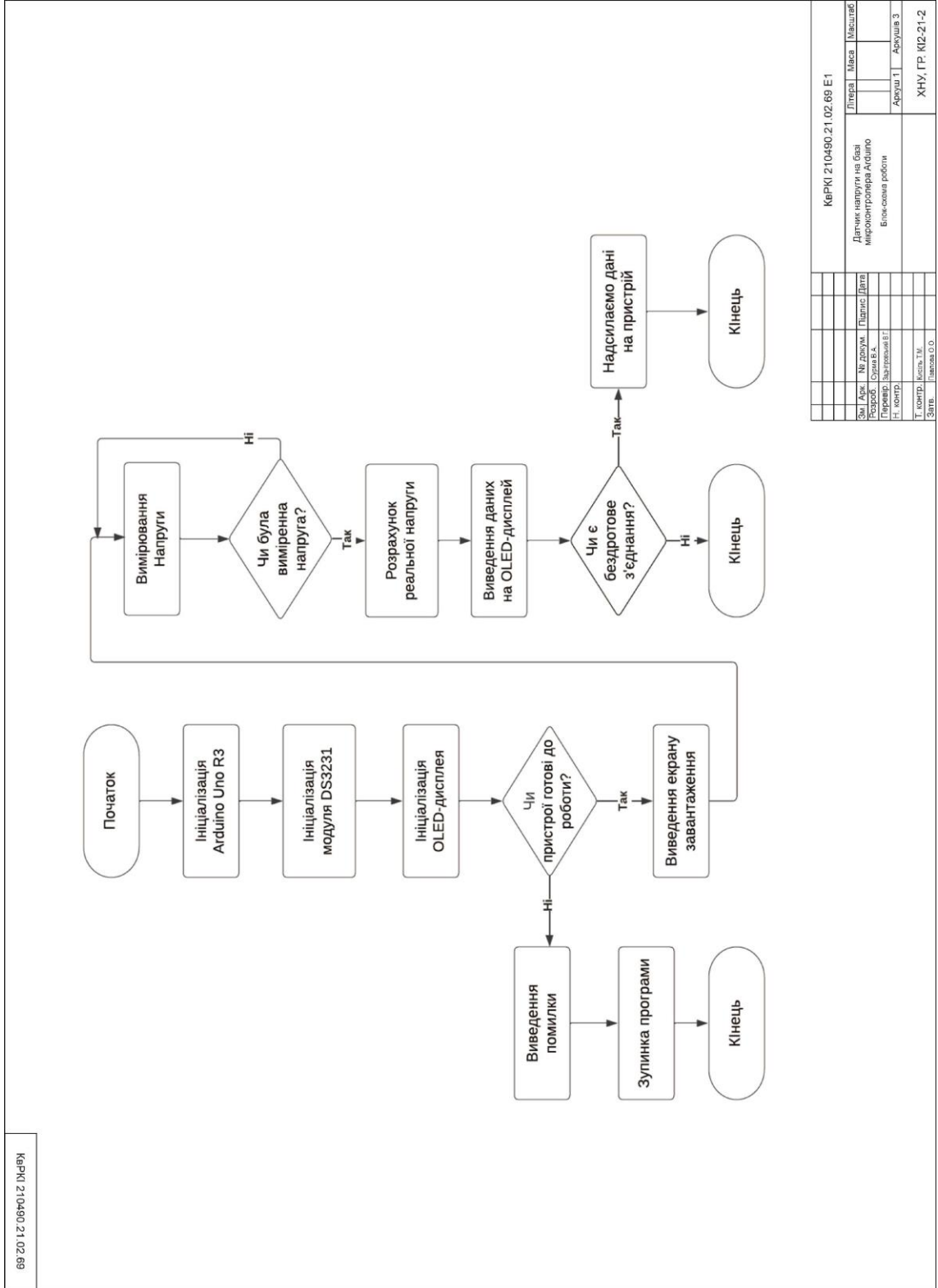
49. Модуль DS3231 RTC. URL: <http://ardushop.in.ua/arduino/real-time-clock-module-ds3231-rtc> (дата звернення: 13.05.2025).

50. Модуль HC-06 (Bluetooth). URL: <https://arduino.ua/prod241-bluetooth-modyl-hc-06?srsltid> (дата звернення: 14.05.2025).

51. Оптрон. URL: <https://www.electronics-tutorials.ws/blog/optocoupler.html> (дата звернення: 15.05.2025).

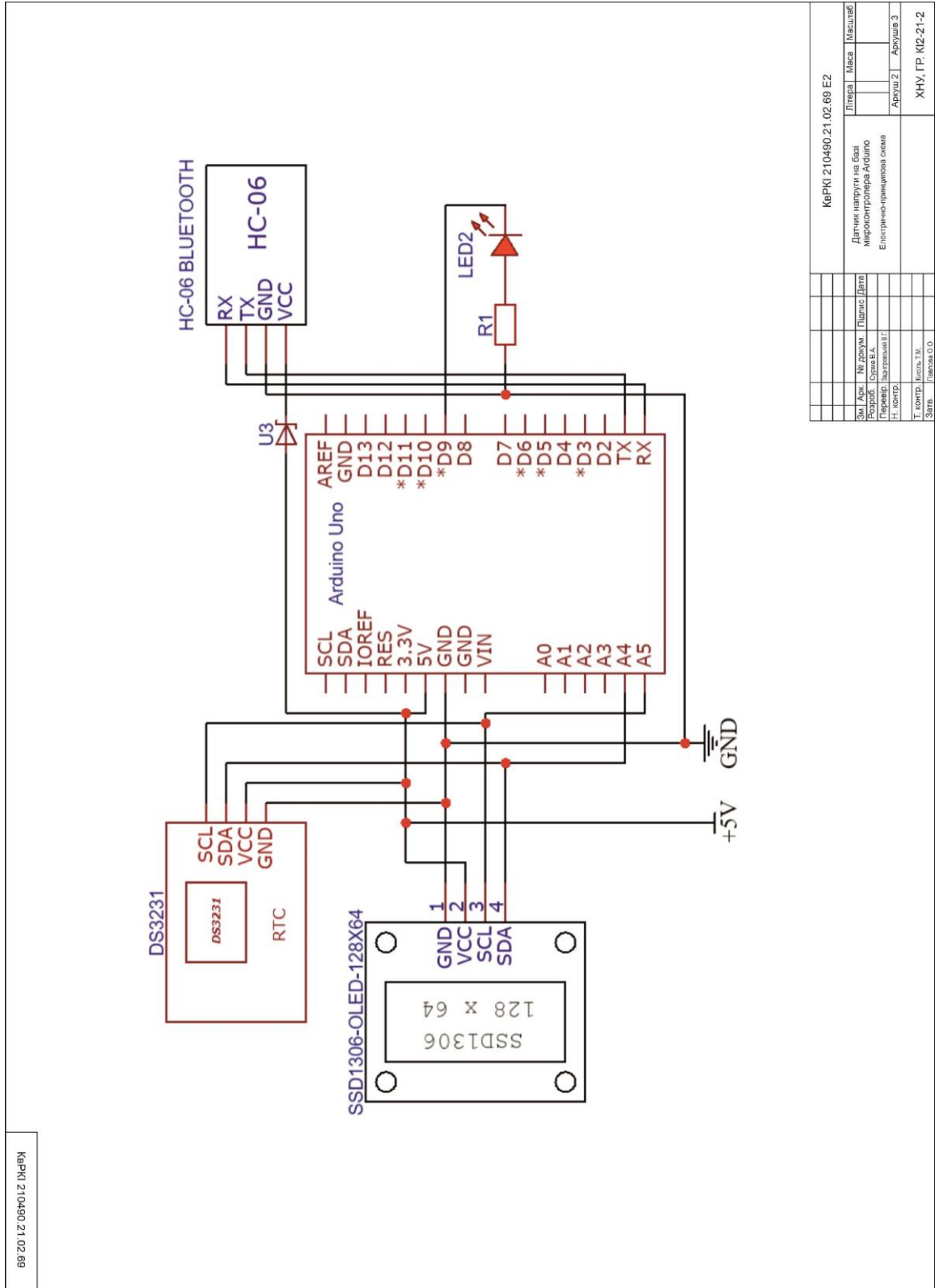
Додаток А (обов'язковий)

КОПІЯ КРЕСЛЕННЯ «БЛОК-СХЕМА РОБОТИ ПЗ»



Додаток Б (обов'язковий)

КОПІЯ КРЕСЛЕННЯ «ЕЛЕКТРИЧНО-ПРИНЦИПОВА СХЕМА»

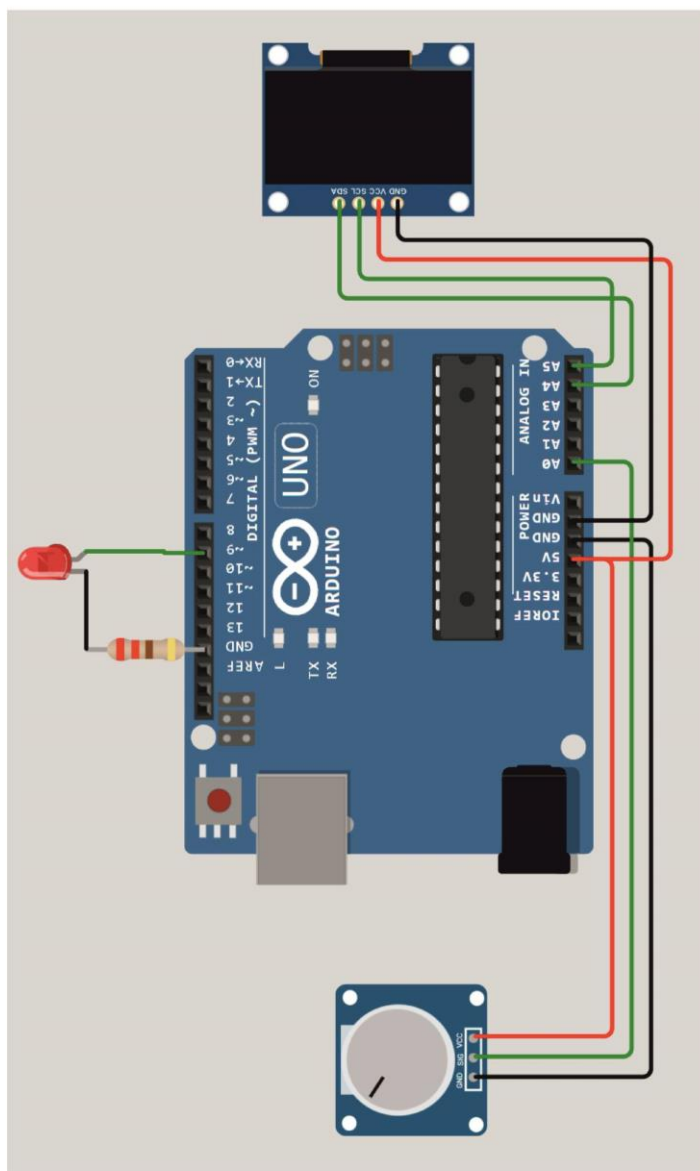


69 20 12 069012 12069012

| | | | |
|--|-------------------------|---------|--|
| КвРКІ 210490.21.02.09 E2 | | | |
| Листопад | Місяць | | |
| Датчик напруги на базі мікроконтролера Arduino | Електроімітаторна схема | | |
| Архив | Архив 2 | Архив 3 | |
| ХНУ, ГР. КІ2-21-2 | | | |

Додаток В (обов'язковий)

КОПІЯ КРЕСЛЕННЯ «ЕМУЛЯЦІЯ У WOKWI»



КерРКІ 210490.21.02.69

| | | | | | | | | | | |
|---------------------------|-----------|-----------------|-------|---------|-------|----------|---|-------------------|----------|--|
| КерРКІ 210490.21.02.69 ЕЗ | | | | | | | | | | |
| Змі. Акт. | № розум. | Підпис: | Дата: | Літера: | Маса: | Масштаб: | | | | |
| Розроб. | Схемат. | | | | | | Датчик напруги на базі мікроконтролера Arduino Емуляція у Wokwi | | | |
| Н. контр. | Паралель: | Загальнооб'єкт. | | | | | Аркуш. 3 | | Аркуш. 3 | |
| Т. контр. | Масштаб: | | | | | | | ХНУ, ГР. КІ2-21-2 | | |
| Звіт. | Повторяє: | | | | | | | | | |

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

Кваліфікаційна робота бакалавра

Кваліфікаційна робота на тему: Датчик напруги
на базі мікроконтролера Arduino

Виконав: студент IV курсу

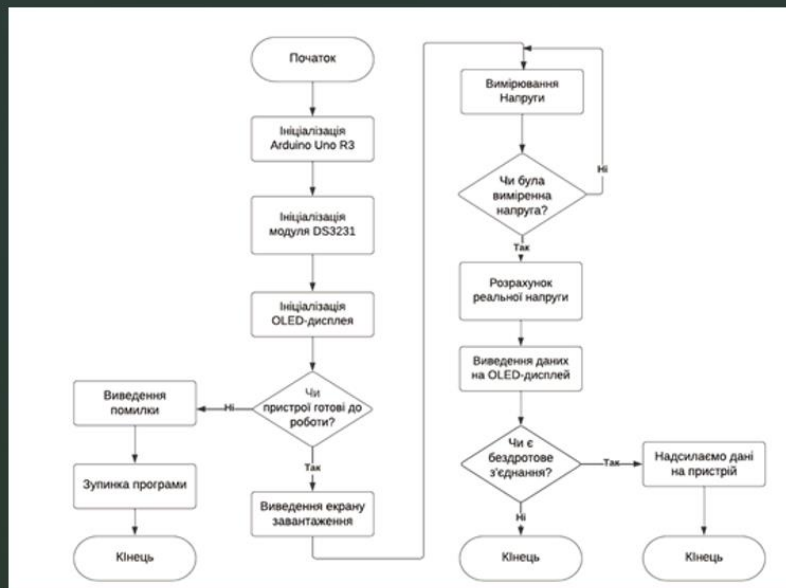
кафедри комп'ютерної інженерії та
інформаційних систем, групи KI2 – 21 – 2 Сурма В.А.

Керівник: Вадим Задніпровський, ст. викладач

Мета та завдання

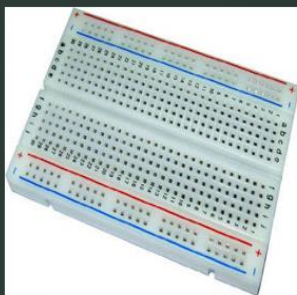
Метою роботи є створення та тестування датчика напруги на платформі Arduino для моніторингу електричних параметрів. Для досягнення цієї мети визначено завдання: проаналізувати предметну область і особливості Arduino, спроектувати апаратну та програмну частини системи, реалізувати датчик із інтеграцією дільника напруги, дисплея, модуля реального часу та Bluetooth, а також протестувати систему в емуляторі Wokwi для перевірки її функціональності.

Блок-схема роботи пз

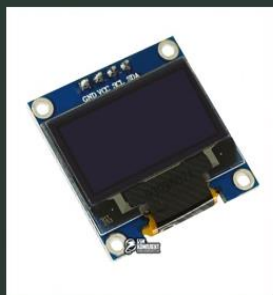


Модулі які застосовувались

Макетна плата



OLED-дисплей



Модуль DS3231



Резистор 100 кОм



Батарея 9 В



Модуль HC-06



Висновки

Розроблено функціональний датчик напруги на базі Arduino, який успішно виконує зчитування, обробку та виведення даних через OLED-дисплей, Serial Monitor і Bluetooth. Датчик має практичне застосування в системах моніторингу електроживлення, слугує навчальним зразком для вивчення електроніки та програмування, а також є основою для масштабування, наприклад, для захисту від перенапруги

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: 246840 Title: БКР Датчик напруги на базі мікроконтролера Arduino Added in a DB: 2025-06-19 Authors: Володимир СУРМА Heads: Владим ЗАДНІПРОВСЬКИЙ Consultants: Opponents: | Document | | Sum coincidence on the DB | |
| | Symbols | Lexemes | Symbols | Lexemes |
| | 110285 | 732 | 975 (1%) | 15 (2%) |

| Plagiarism sources | | Plagiarism presence in the document | |
|--------------------|-------------|-------------------------------------|---------|
| ID | Description | Symbols | Lexemes |
| | | | |

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 СУРМА

Співавтор:

Назва: Сурма_Датчик напруги на базі мікроконтролера Arduino

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 1.1%

Коефіцієнт подібності 2: 0.2%

Мікропробіли: 5

Заміна букв: 1

Інтервали: 0

Білі знаки: 30

Дата створення звіту: 2025-06-19 06:31:54.0

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 процедур. Таким чином робота не приймається.

Обґрунтування:

2025-06-19

Дата



Доцент Андрій Нічепорук

експерт

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА
ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Датчик напруги на базі мікроконтролера Arduino
Автор Володимир СУРМА
Освітня програма Комп'ютерна інженерія та програмування
Рівень вищої освіти перший (бакалаврський) рівень
Спеціальність 123– Комп'ютерна інженерія
Науковий керівник: Вадим ЗАДНІПРОВСЬКИЙ, ст. викладач.

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

| № | Висновок | Позначка про відповідність |
|-----|---|----------------------------|
| 1 | Ознаки академічного плагіату | |
| 1.1 | Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту. | Відповідає |
| 1.2 | Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. | |
| 1.3 | Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 1.4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |
| 2 | Інші види порушень академічної доброчесності | Не виявлено |

Підтвердження:

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

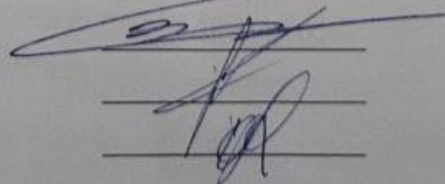
- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 1.1%, 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



Вадим ЗАДНІПРОВСЬКИЙ

Андрій НІЧЕПОРУК

Ольга ПАВЛОВА

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Сурма Володимир Анатолійович

Тема: Датчик напруги на базі мікроконтролера Arduino

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 64

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розробка Програмно-технічного датчика напруги на базі мікроконтролера Arduino

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

3. У роботі за результатами виконаних теоретичних та практичних досліджень було розроблено програмно-технічний засіб датчика напруги на базі Arduino, який забезпечує точне вимірювання електричної напруги в діапазоні, визначеному апаратною конфігурацією

У першому розділі було проведено всебічний аналіз існуючих аналогів, методів вимірювання напруги та сучасних технічних засобів, включаючи різні типи датчиків напруги, аналого-цифрові перетворювачі (АЦП), мікроконтролери та технології бездротової передачі даних.

Другий розділ присвячений розробці архітектури системи та її програмного забезпечення. Детально описано функціональну схему датчика, що включає схему підключення дільника напруги, реалізацію аналогового входу та інтеграцію периферійних модулів. Розроблено структурну схему програмного забезпечення, яка передбачає модульний підхід до обробки даних, включаючи ініціалізацію, зчитування, обробку, виведення та передачу інформації.

У третьому розділі було здійснено практичну реалізацію розробленого програмного забезпечення та проведено його тестування. Код прошивки sketch.ino успішно реалізував основний алгоритм, включаючи ефективне зчитування та усереднення 10 зразків з аналогового входу A0, конвертацію у напругу з

урахуванням коефіцієнта дільника, форматування міток часу від RTC та виведення результатів на OLED-дисплей, Serial Monitor та Bluetooth.

Таким чином, розроблений датчик напруги на базі мікроконтролера Arduino є функціональним та ефективним рішенням для моніторингу електричних параметрів, демонструючи потенціал для широкого застосування в автоматизованих системах контролю та збору даних.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатньо глибокий аналіз фільтрації та обробки біосигналів.

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 технічному рівні.


8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно

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

Щаршук Володимир Володимирович, д.т.н.,
професор, зав. каф. АКІТрР

“19” *серпня* 2025 р.

 (підпис)

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

Володимира СУРМИ

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-2

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.

18.06. 2025 року

