

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут автоматики і електротехніки

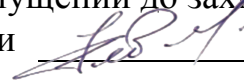
(повне найменування інституту, назва факультету)

Кафедра програмованої електроніки, електротехніки і телекомунікацій

(повна назва кафедри)

«Допущений до захисту»

Завідувач кафедри



В. М. Рябенський

д.т.н., професор

« 15 » грудня 2025 р.

Кваліфікаційна робота

на здобуття ступеня вищої освіти «магістр»

(освітньо-кваліфікаційний рівень)

на тему: Розробка програмного забезпечення для суднових вбудованих систем збору та обробки даних

Виконав: 6321м групи
студент



С.А. Івановський

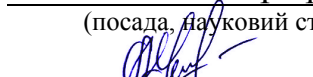
(підпис)

Керівник роботи:

професор каф. ПЕЕТ, д.т.н.,

професор

(посада, науковий ступінь, вчене звання)



О.О. Ушкаренко

(підпис)

м. Миколаїв–2025року

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут автоматики та електротехніки

Кафедра	програмованої електроніки, електротехніки і телекомунікацій
Спеціальність	171«Електроніка»
Освітня програма	Електронні системи

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

В. М. Рябенський
(підпис)

« 09 » жовтня 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти «магістр» (освітньо-кваліфікаційний рівень)

Студенту Івановському Святославу Анатолійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка програмного забезпечення для суднових вбудованих систем збору та обробки даних
керівник роботи Ушкаренко Олександр Олегович, професор каф. ПЕЕТ, д.т.н., професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ректора № 1217 від « 14 » жовтня 2025 року.

2. Термін подання роботи: 09.12.2025 р.

3. Вихідні дані по роботі: технічні характеристики суднових вбудованих систем керування, збору та обробки даних, вимоги до надійності та роботи в реальному часі, специфікації сенсорів і підсистем, з якими взаємодіє програмне забезпечення

4. Перелік питань, що належать до розробки (найменування розділів)

1. Аналіз предметної області розробки ПЗ в контексті суднових вбудованих систем
2. Проектування ПЗ для суднової системи керування синхронним генератором
3. Реалізація програмного забезпечення для забезпечення паралельної роботи ССГ
4. Охорона праці
5. Перелік презентаційних матеріалів
1. Принципова схема вбудованої системи керування синхронним генератором
2. Блок-схеми алгоритмів керування судновим синхронним генератором
3. Інтерфейс програмного забезпечення АРМ оператора для керування генератором
4. Принципова схема МП системи підключення ССГ на загальне навантаження
5. Блок-схеми алгоритмів обробки переривань та формування сигналів керування

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Тубальцев А.М.		

7. Дата видачі завдання « 15 » жовтня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз предметної області та формулювання вимог до програмного забезпечення для судових вбудованих систем керування, збору та обробки даних	04.11.2024	виконано
2.	Огляд і порівняння існуючих рішень для судових вбудованих систем збору та обробки даних	30.12.2024	виконано
3.	Опис апаратної частини вбудованої системи керування синхронним генератором	17.02.2025	виконано
4.	Проектування та опис алгоритмів керування, збору й обробки даних	31.03.2025	виконано
5.	Реалізація програмного забезпечення в обраному середовищі	26.05.2025	виконано
6.	Проведення тестування та аналіз отриманих результатів	29.09.2025	виконано
7.	Оптимізація та вдосконалення програмного забезпечення за результатами тестування	17.11.2025	виконано
8.	Підготовка розділу з охорони праці, оформлення кваліфікаційної роботи, підготовка графічного матеріалу	20.12.2025	виконано

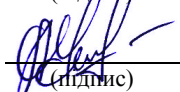
Студент


(підпис)

С.А. Івановський

(прізвище та ініціали)

Керівник роботи


(підпис)

О.О. Ушкаренко

(прізвище та ініціали)

Анотація

У магістерській кваліфікаційній роботі розглянуто питання створення програмного забезпечення для суднових вбудованих систем керування, збору та обробки даних. Розроблене рішення дозволяє здійснювати збір інформації з різних сенсорів та підсистем судна, виконувати її фільтрацію, попередній аналіз і передавати результати для подальшого використання у системах керування. Запропоновано архітектуру програмного забезпечення, яка забезпечує ефективність роботи у реальному часі та враховує особливості функціонування суднових вбудованих систем. Реалізовано методи оптимізації використання обчислювальних ресурсів та захисту даних від втрат у випадку збоїв. Практичне значення роботи полягає у можливості впровадження розробленого програмного забезпечення у сучасні суднові комплекси для підвищення їх надійності, безпеки та зручності експлуатації.

Ключові слова: суднові вбудовані системи, збір даних, обробка даних, програмне забезпечення, реальний час, автоматизація.

Abstract

This master's thesis is focused on the development of software for embedded shipboard data acquisition and processing systems. The proposed solution enables the collection of information from various ship sensors and subsystems, provides data filtering and preliminary analysis, and transmits results for further use in control systems.

The architecture of the developed software ensures real-time performance and takes into account the specific requirements of marine embedded systems. Methods for optimizing computational resource usage and protecting data against losses in case of failures have been implemented.

The practical significance of this research lies in the possibility of integrating the developed software into modern ship systems in order to enhance their reliability, safety, and operational efficiency.

Keywords: embedded ship systems, data acquisition, data processing, software, real-time, automation.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В КОНТЕКСТІ СУДНОВИХ ВБУДОВАНИХ СИСТЕМ	10
1.1. Сучасний стан і тенденції розвитку суднових вбудованих систем збору та обробки даних	11
1.2. Аналіз архітектур суднових автоматизованих систем управління	13
1.3. Вимоги до програмного забезпечення вбудованих систем	15
1.4. Аналіз існуючих програмних засобів для автоматизації суднових електроенергетичних систем	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМ КЕРУВАННЯ СУДНОВИМ СИНХРОННИМ ГЕНЕРАТОРОМ	24
2.1. Опис апаратних засобів вбудованої мікропроцесорної системи	25
2.2. Розробка алгоритму програми налаштування параметрів регулятора системи керування ССГ	29
2.3. Розробка програмного забезпечення автоматизованого робочого місця оператора для керування судновим синхронним генератором	33
2.4. Програмна реалізація регулятора та методика його налаштування	35
Висновки до розділу	37
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАБЕЗПЕЧЕННЯ ПАРАЛЕЛЬНОЇ РОБОТИ СУДНОВОГО СИНХРОННОГО ГЕНЕРАТОРА	39
3.1. Програмно-апаратне забезпечення підключення генератора до загального навантаження	40

3.2. Обґрунтування параметрів дільників для забезпечення роботи таймерів-лічильників	44
3.3. Програмна організація алгоритму керування мікропроцесорною системою	46
3.4. Розробка програмного інтерфейсу для взаємодії з блоком синхронізації	54
Висновки до розділу	56
РОЗДІЛ 4. ОХОРОНА ПРАЦІ	57
4.1. Вступ	58
4.2. Аналіз небезпечних і шкідливих факторів, впливаючих на людину, при розробці програмного забезпечення	59
4.3. Розрахунок захисного заземлення	60
ВИСНОВКИ	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	68
ДОДАТОК А. Програмний код для мікроконтролера у складі вбудованої системи керування судновим синхронним генератором, збору та обробки даних	73
ДОДАТОК Б. Програмний код системи синхронізації генераторів	77

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

АРМ – автоматизоване робоче місце
АСУ – автоматизована система управління
АЦП – аналого-цифровий перетворювач
ЕРС – електрорушійна сила
ІС – інтегральні схеми
МК – мікроконтролер
МП – мікропроцесор
САПР – система автоматизації проектування
СГ – синхронний генератор
СЕЕС – суднова електроенергетична система
СЗ – система збудження
ССГ – судновий синхронний генератор
ПЗ – програмне забезпечення
ПІД – пропорційно-інтегрально-диференціальний регулятор
ПК – персональний комп'ютер
MCU (Microcontroller Unit) – мікроконтролер
UART (Universal Asynchronous Receiver-Transmitter) – універсальний асинхронний приймач-передавач
IDE (Integrated Development Environment) – інтегроване середовище розробки

					171.6321м.КР.ПЗ.Скорочення	Арк.
						6
Змн.	Арк.	№ документу	Підпис	Дата		

ВСТУП

Сучасний розвиток морської галузі неможливий без впровадження інноваційних інформаційних технологій, які забезпечують підвищення безпеки та ефективності експлуатації суден. Суднові енергетичні системи вимагають високого рівня надійності та точності керування для забезпечення стабільної роботи всіх бортових споживачів електроенергії. Ключовим елементом таких систем є судновий синхронний генератор, ефективна робота якого значною мірою залежить від правильної організації процесу збудження та синхронізації з іншими генераторами на загальне навантаження. Вбудовані системи керування, збору та обробки даних на борту суден відіграють ключову роль у моніторингу технічного стану обладнання, контролі параметрів навколишнього середовища, а також в автоматизації керування різними підсистемами. Такі системи дозволяють своєчасно виявляти відхилення у роботі механізмів, зменшувати ймовірність аварійних ситуацій, оптимізувати витрати палива та забезпечувати відповідність міжнародним стандартам морської безпеки.

Програмне забезпечення є основою функціонування вбудованих систем, оскільки воно визначає алгоритми збору, збереження та обробки інформації, а також інтерфейси взаємодії з оператором та іншими підсистемами. Високі вимоги до надійності, продуктивності та захисту від збоїв потребують застосування сучасних методів розробки та тестування програмного забезпечення, орієнтованого на роботу в реальному часі. У цьому контексті особливої важливості набуває розробка програмного забезпечення для вбудованої мікропроцесорної системи керування судновим синхронним генератором, яке забезпечує стабільність напруги та частоти, контроль режимів роботи генератора і автоматизацію регулювання збудження.

Не менш актуальною є розробка програмного забезпечення для мікропроцесорної системи керування введення генератора на паралельну

					171.6321м.КР.ПЗ.Вступ	Арк.
						7
Змн.	Арк.	№ документу	Підпис	Дата		

роботу з іншими генераторами. Система синхронізації генераторів є складним технічним комплексом, який дозволяє забезпечити рівномірний розподіл навантаження між енергоблоками та безпечне підключення нового генератора до загальної електричної мережі судна. Від точності роботи таких систем залежить ефективність енергопостачання, запобігання аварійним режимам та продовження ресурсу обладнання, що робить автоматизовані рішення особливо актуальними.

Наукова й практична актуальність теми полягає у необхідності створення ефективних програмних рішень для судових вбудованих систем, які здатні забезпечити керування елементами судової електроенергетичної системи, зокрема судовим синхронним генератором, збір та аналіз даних у складних умовах експлуатації.

Об'єктом дослідження є судові електроенергетичні системи та процеси функціонування судових вбудованих систем збору та обробки даних.

Предметом дослідження є методи, алгоритми та програмні засоби для вбудованих мікропроцесорних систем керування судовим синхронним генератором.

Метою роботи є розробка програмного забезпечення для судових вбудованих систем керування, збору та обробки даних, яке забезпечує автоматизацію процесів збудження, підтримання стабільної напруги та введення генераторів на паралельну роботу.

При виконанні роботи використано наступні методи досліджень:

- метод системного аналізу – для формування вимог до архітектури програмного забезпечення та опису взаємодії між підсистемами.

- моделювання алгоритмів – для проектування процесів керування, збору, збереження та попередньої обробки даних у судовій електроенергетичній системі.

					171.6321м.КР.ПЗ.Вступ	Арк.
						8
Змн.	Арк.	№ документу	Підпис	Дата		

– методи об’єктно-орієнтованого програмування – при розробці архітектури та реалізації програмного забезпечення автоматизованого робочого місця оператора.

– методи оптимізації – для підвищення ефективності використання обчислювальних ресурсів вбудованої мікропроцесорної системи .

Новизна отриманих результатів полягає у тому, що запропоноване та реалізоване програмне забезпечення для суднових вбудованих систем забезпечує адаптивний підхід до збору та фільтрації даних з урахуванням специфіки морських умов, інтеграцію модулів попередньої обробки даних у режимі реального часу, оптимізацію обчислювальних ресурсів вбудованої системи з можливістю масштабування та модифікації для різних типів суднових підсистем.

Зв’язок з науковими програмами, темами, планами: робота була складовою держбюджетної НДР «Розробка енергоефективних засобів генерації та перетворення електроенергії для систем розмагнічування малих кораблів» (ДР №0124U001522), в яких автор брав участь як виконавець.

					171.6321м.КР.ПЗ.Вступ	Арк.
						9
Змн.	Арк.	№ документу	Підпис	Дата		

					171.6321м.КР.ПЗ.Р1			
Змн.	Арк.	№ документу	Підпис	Дата	Аналіз предметної області розробки програмного забезпечення в контексті суднових вбудованих	Літ.	Арк.	Аркуші
Розробник		Івановський С.А.						
Консультант							10	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Ушкаренко О.О.						
Зав. каф.		Рябенський В. М.						

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В КОНТЕКСТІ СУДНОВИХ ВБУДОВАНИХ СИСТЕМ

1.1. Сучасний стан і тенденції розвитку суднових вбудованих систем збору та обробки даних

Останні роки у світі активно досліджується тема цифровізації морського транспорту та створення вбудованих систем збору й обробки даних. Такі системи дають змогу автоматично отримувати інформацію з сенсорів і підсистем судна, обробляти її на місці та передавати далі у системи керування або берегові центри. В оглядах літератури відзначається, що основною проблемою є обмежені канали зв'язку між судном і берегом, тому дедалі більше уваги приділяється обробці даних безпосередньо на борту судна [4]. Це означає, що програмне забезпечення має бути легким, працювати в реальному часі й не перевантажувати обчислювальні ресурси.

Важливим питанням є стандартизація. Міжнародні стандарти ІЕС 61162-1/-2 описують класичні інтерфейси для передачі даних між обладнанням, тоді як ІЕС 61162-450 і ІЕС 61162-460 регламентують роботу через Ethernet та вимоги до кібербезпеки [1], [2]. У практичних дослідженнях зазначається, що перехід до нових стандартів відбувається поступово, і на реальних судах часто одночасно працюють як старі, так і нові протоколи [3]. Це створює труднощі для розробників програмного забезпечення, адже необхідно забезпечити сумісність різних систем.

Чимало досліджень зосереджено на сенсорних мережах NMEA 2000, які використовуються для моніторингу допоміжних систем судна. Роботи українських авторів також підкреслюють, що важливо не тільки збирати дані, а й перевіряти їх достовірність і захищати від спотворення [12], [13]. Це пов'язано з тим, що від правильності показників залежить безпека плавання.

Велика кількість наукових статей присвячена обробці даних у реальному часі. Наприклад, у роботі [4] описано методи так званої «edge-

					171.6321м.КР.ПЗ.Р1	Арк.
						11
Змн.	Арк.	№ документу	Підпис	Дата		

аналітики», які дозволяють зменшувати обсяг даних для передачі без втрати корисності. Інші дослідження пропонують методи стискання або фільтрації даних [5]. Проте автори зазначають, що існує проблема нестачі відкритих наборів реальних судових даних для перевірки та порівняння таких алгоритмів.

Окремим напрямом є технічна діагностика та моніторинг стану обладнання. У багатьох роботах, у тому числі українських [14], [15], розглядаються методи прогнозування несправностей судових двигунів та генераторів. Це дозволяє перейти від планового технічного обслуговування до обслуговування «за станом», що робить експлуатацію економнішою.

Досить активно в останні роки обговорюється питання кібербезпеки. У зарубіжних статтях описуються приклади атак на судові інформаційні системи та пропонуються методи захисту [9], [10]. Українські дослідники також звертають увагу на цю проблему, особливо у контексті відповідності міжнародним вимогам [12]. Проте поки що бракує практичних рекомендацій, як саме реалізувати кіберзахист у вбудованому програмному забезпеченні.

Українські автори приділяють увагу і створенню інформаційних систем для морського спостереження та збирання метеоданих [13], [14]. Хоча ці роботи здебільшого носять оглядовий характер, вони підтверджують актуальність теми та необхідність подальшої розробки ефективного програмного забезпечення для судових систем.

Таким чином, аналіз літератури показав, що сьогодні існують напрацювання у сфері збору та обробки даних на борту судна, але лишаються невирішені проблеми – забезпечення сумісності різних протоколів, підвищення рівня кібербезпеки та створення відкритих інструментів для тестування. Це підтверджує доцільність розробки програмного забезпечення, присвяченого саме цим завданням.

					171.6321м.КР.ПЗ.Р1	Арк.
						12
Змн.	Арк.	№ документу	Підпис	Дата		

1.2. Аналіз архітектур суднових автоматизованих систем управління

Автоматизована система управління (АСУ) – це комплекс обладнання та програмного забезпечення, який забезпечує контроль і управління різними процесами в межах технологічного процесу. Залежно від організації управління, АСУ поділяють на децентралізовані (рис. 1.1, в), централізовані (рис. 1.1, а, б) та ієрархічні (рис. 1.1, г) [22].

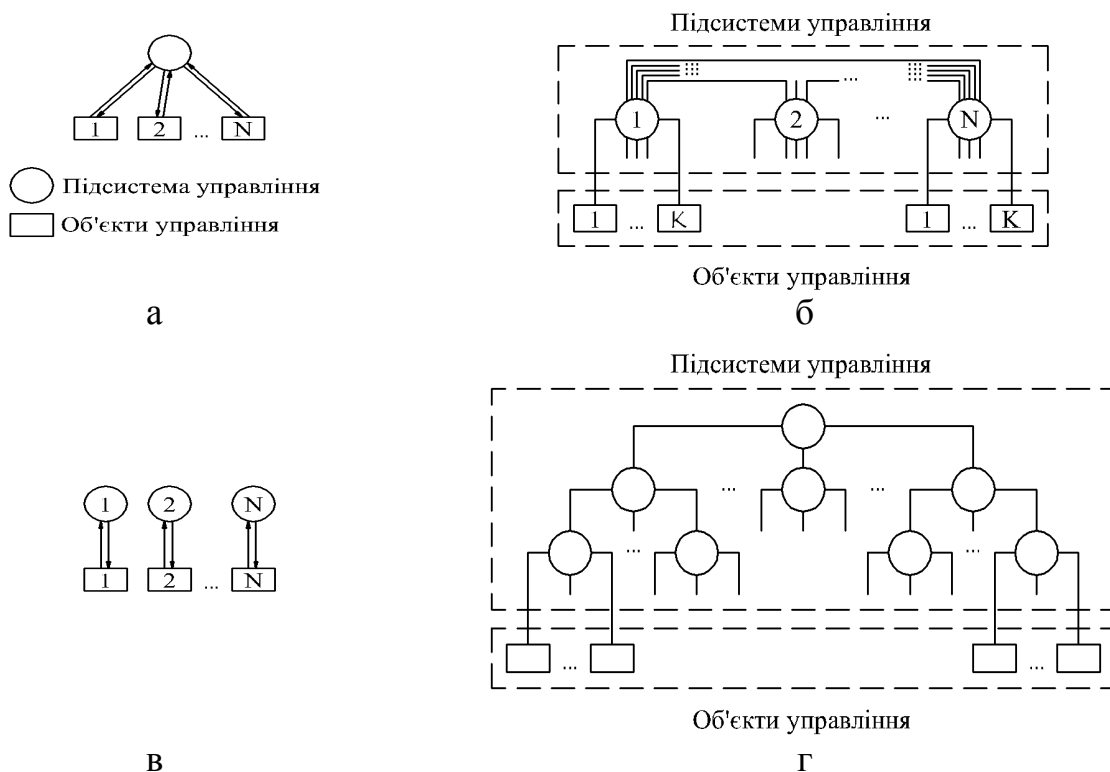


Рисунок 1.1 – Структури автоматизованих систем управління: а – централізована; б – централізована розосереджена; в – децентралізована; г – ієрархічна

Децентралізовані АСУ (рис. 1.1, в) передбачають розподіл функцій управління між окремими елементами системи. Такі системи можуть ефективно працювати лише тоді, коли об'єкти управління незалежні між собою за ресурсами — матеріальними, енергетичними чи інформаційними. У разі взаємозалежності елементів використовують централізовані або ієрархічні системи.

У централізованих АСУ (рис. 1.1, а) всі процеси управління здійснюються через одну центральну підсистему. Переваги такої структури

полягають у простоті інформаційної взаємодії, можливості оптимально управляти всією системою, легкому коригуванні змінних параметрів і досягненні високої ефективності при мінімальному обсязі технічних засобів. Недоліки ж включають високу вимогу до надійності та продуктивності центрального органу управління, великі відстані каналів зв'язку при розосереджених об'єктах і складність підвищення функціональності.

Щоб зменшити навантаження на центральний орган і скоротити довжину каналів зв'язку, було запропоновано централізовані розосереджені АСУ (рис. 1.1, б), де єдина підсистема управління поділяється на кілька взаємопов'язаних підсистем. Це дозволяє частково розвантажити центральну підсистему, проте ускладнює інформаційні процеси та потребує додаткових технічних засобів для обробки даних. Через це централізовані системи підходять головним чином для простих об'єктів.

Щоб подолати ці обмеження, застосовують ієрархічні АСУ (рис. 1.1, г) – багаторівневі системи, де функції управління розподіляються між різними підсистемами. Основні особливості таких систем:

- вертикальна підпорядкованість підсистем, тобто підсистеми розташовані на різних рівнях;
- пріоритет дій: верхні рівні керують загальною поведінкою системи, але не реагують безпосередньо на локальні зміни, які контролюють нижчі рівні;
- залежність часу формування керуючих впливів від рівня системи: чим вище рівень, тим більша невизначеність і довший час прийняття рішення.

Дослідження показують, що сучасні системи управління електроенергетичними установками здебільшого мають трирівневу ієрархічну структуру:

					171.6321м.КР.ПЗ.Р1	Арк.
						14
Змн.	Арк.	№ документу	Підпис	Дата		

1. Нижній рівень – локальні системи управління генераторами та виконавчими механізмами, пристрої первинної обробки аналогових і дискретних сигналів;

2. Середній рівень – відповідає за координацію взаємодії силових агрегатів, оптимізацію їх роботи; включає програмовані логічні контролери (ПЛК), системи синхронізації та розподілу навантажень;

3. Верхній рівень – алгоритми управління всією електростанцією та інформаційна підтримка оператора.

1.3. Вимоги до програмного забезпечення вбудованих систем

При створенні програмного забезпечення важливо враховувати як теоретичні основи аналізу електричних кіл та електронних систем, так і практичні підходи до побудови мікропроцесорних пристроїв і цифрових модулів. У навчальному посібнику [30] розглянуті методи програмної реалізації розрахунку та аналізу електричних кіл, що може стати основою для алгоритмів збору й обробки даних у суднових системах. Більш широкий підхід до моделювання електронних, електротехнічних та енергетичних систем подано в [33], де увага акцентується на створенні математичних моделей, які зручно реалізовувати у вигляді програмних рішень. Такі матеріали допомагають зрозуміти, як моделювання може використовуватися під час розробки програмного забезпечення для підвищення точності обчислень та аналізу даних.

У підручнику [32] описані основні принципи роботи мікропроцесорної техніки, що дозволяє краще зрозуміти, як саме реалізуються вбудовані системи збору інформації. У роботі [37] більш детально висвітлюються питання програмованих електронних систем збору та обробки даних, де наголошується на важливості адаптивності та надійності. Це особливо важливо для суднових умов, де система повинна працювати без збоїв при різних навантаженнях.

					171.6321м.КР.ПЗ.Р1	Арк.
						15
Змн.	Арк.	№ документа	Підпис	Дата		

Праці [34] та [36] зосереджені на цифровій схемотехніці та проєктуванні пристроїв на ПЛІС із використанням середовища MAX+plus II і мови Verilog. Використання таких технологій дозволяє реалізовувати окремі алгоритми обробки даних не тільки програмним способом, але й апаратно, що значно підвищує швидкодію системи. Це створює основу для поєднання апаратних і програмних рішень у розробці вбудованих систем.

Дослідження [31] показує, що навіть невеликі коливання швидкості приводних двигунів можуть негативно впливати на розподіл потужності, і пропонує методи для підвищення точності роботи таких систем. Це підтверджує важливість програмного аналізу та обробки даних у режимі реального часу. Додатково, у статті [35] подано формальний опис елементів автоматизованого робочого місця оператора, що може бути використано при створенні програмних інтерфейсів для зручної та безпечної роботи персоналу.

Таким чином, аналіз літературних джерел [30-37] показує, що ефективне програмне забезпечення для суднових вбудованих систем повинно поєднувати математичне моделювання, оптимізацію алгоритмів під мікропроцесорні системи, а також передбачати створення зручних і надійних інтерфейсів для оператора. Це дозволить забезпечити стабільну та точну роботу систем збору і обробки даних у складних умовах експлуатації суден.

Програмне забезпечення для вбудованих суднових систем збору та обробки даних має відповідати ряду вимог, які впливають зі складності таких систем і умов їх експлуатації.

Системний підхід. У процесі проєктування необхідно враховувати взаємодію всіх елементів системи, а також особливості їх інтеграції в єдине апаратно-програмне середовище. Для цього застосовується модульно-ієрархічний підхід у створенні апаратної частини та об'єктно-орієнтований підхід у розробці програмної частини. Це дозволяє розділяти складну систему на окремі рівні та підсистеми.

					171.6321м.КР.ПЗ.Р1	Арк.
						16
Змн.	Арк.	№ документа	Підпис	Дата		

Функціональні вимоги. Основною метою програмного забезпечення є забезпечення збору, контролю та обробки даних від сенсорів, моніторинг стану обладнання, а також надання оператору зручного людино-машинного інтерфейсу. Програмні засоби повинні виконувати систематизацію, збереження та візуалізацію даних, забезпечуючи швидкий доступ до необхідної інформації.

Ієрархічність. Умовно програмно-апаратний комплекс можна поділити на три рівні:

верхній – збір, обробка та відображення інформації;

середній – організація та підтримка мережевої взаємодії;

нижній – робота з датчиками, виконавчими модулями та іншими пристроями.

Модульність. Конструктивна реалізація апаратної та програмної частин має бути модульною. Для програмного забезпечення це означає поділ функцій між окремими модулями чи класами, що спрощує оновлення та супровід.

Принцип розвитку. Програмне забезпечення має бути масштабованим, з можливістю підключення нових модулів, зміни алгоритмів та оптимізації системи під нові умови експлуатації.

Зв'язність і зворотний зв'язок. Програма повинна не лише обробляти вхідні дані, а й враховувати взаємодію підсистем, забезпечуючи контроль результатів і стабільність роботи навіть при випадкових змінах параметрів.

Надійність та гнучкість. Враховуючи роботу в умовах невизначеності та неповних вихідних даних, програмне забезпечення має забезпечувати стійкість до збоїв, передбачати роботу з неповними або некоректними даними, підтримувати адаптивні алгоритми прийняття рішень.

Зручність інтерфейсу. Людино-машинний інтерфейс має будуватися на основі принципів лаконічності, узагальнення та акцентування важливої інформації. Використання мнемосхем, кольорових виділень і звичних

					171.6321м.КР.ПЗ.Р1	Арк.
						17
Змн.	Арк.	№ документа	Підпис	Дата		

асоціацій допомагає оператору швидко орієнтуватися в ситуації та приймати правильні рішення.

1.4. Аналіз існуючих програмних засобів для автоматизації суднових електроенергетичних систем

Автоматизація суднових електроенергетичних систем (СЕЕС) є одним з ключових напрямів розвитку морської техніки. В умовах зростання складності суднового обладнання та підвищених вимог до безпеки плавання роль програмного забезпечення, що забезпечує збір, обробку і аналіз даних, постійно зростає [16]. У сучасних судах такі системи не лише підтримують роботу енергетичних установок, а й виконують діагностичні, контрольні та прогностичні функції.

Існуючі рішення для автоматизації СЕЕС можна поділити на декілька груп:

- вбудоване програмне забезпечення систем управління двигунами і генераторами, яке забезпечує локальний контроль параметрів;
- системи моніторингу і діагностики, що збирають дані з різних підсистем та дозволяють відстежувати їхній технічний стан у реальному часі;
- інтегровані суднові системи автоматизації (IAS), які об'єднують дані від силових установок, допоміжного обладнання та систем безпеки в єдине інформаційне середовище [17], [18].

Більшість сучасних IAS розробляються провідними міжнародними компаніями (Kongsberg, Wärtsilä, ABB, Siemens). Наприклад, система Kongsberg K-Chief 700 (рис. 1.2) забезпечує централізований моніторинг та управління більшістю суднових підсистем, підтримує модульність і масштабованість [19].

					171.6321м.КР.ПЗ.Р1	Арк.
						18
Змн.	Арк.	№ документу	Підпис	Дата		



Рисунок 1.2 – Інтерфейс ПЗ системи Kongsberg K-Chief 700

Компанія Wärtsilä пропонує рішення UNIMACS (рис. 1.3), яке поєднує автоматизацію енергетичних установок із можливістю інтеграції даних у берегові центри управління [20].

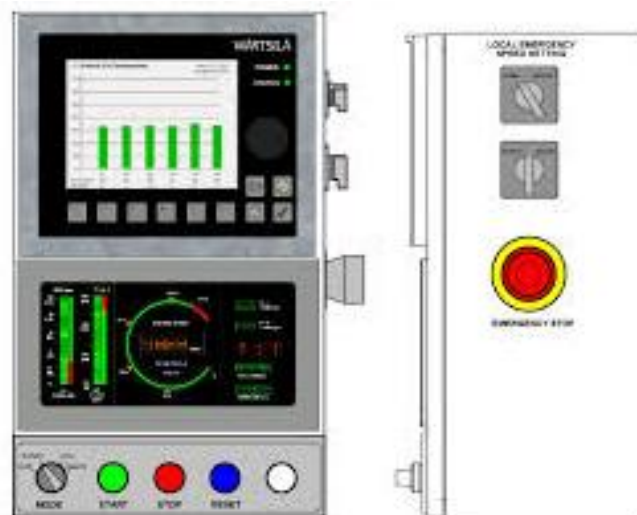


Рисунок 1.3 – Система Wärtsilä 34DF

У системах ABB акцент зроблено на енергоефективності та зниженні витрат палива завдяки оптимізації режимів роботи двигунів і генераторів [21].

					171.6321м.КР.ПЗ.Р1	Арк.
Змн.	Арк.	№ документа	Підпис	Дата		19

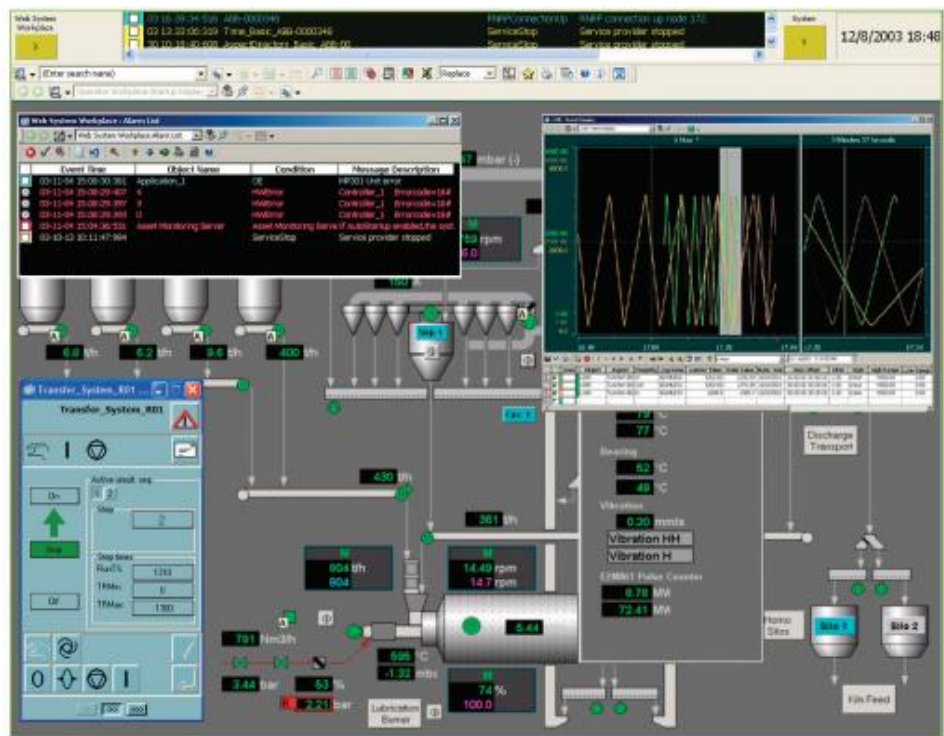


Рисунок 1.4 – Програмні засоби АСУ компанії АВВ

Вітчизняні розробки у цій сфері поступаються за масштабом, однак мають прикладний характер. Дослідження українських науковців [22], [23] акцентують увагу на розробці програмних модулів для моніторингу параметрів дизельних генераторів та прогнозування можливих відмов (рис. 1.5). У цих роботах особлива увага приділяється адаптації програмного забезпечення до умов обмежених обчислювальних ресурсів, що характерно для вбудованих систем.

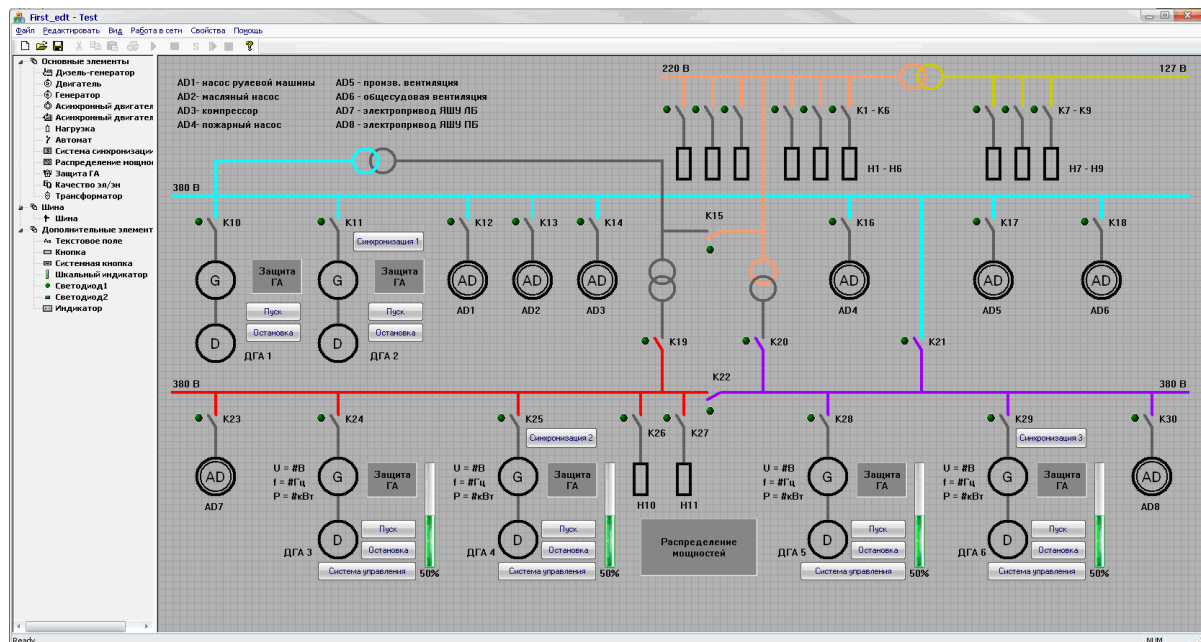


Рисунок 1.5 – Інтерфейс програмного забезпечення контролю параметрів та керування СЕЕС

Існуючі комерційні програмні продукти мають низку переваг:

- модульність і масштабованість – можливість підключення додаткових підсистем без зміни базового ядра;
- зручний операторський інтерфейс – реалізація мнемосхем, індикаторів, звукових і візуальних сигналів;
- сумісність із міжнародними стандартами (IEC, NMEA, Modbus), що дозволяє інтегрувати обладнання різних виробників;
- високий рівень надійності та резервування, що критично для морських умов [17], [19], [21].

Разом із тим, аналіз літератури показує й певні обмеження:

1. Закритість архітектури. Більшість систем є пропрієтарними, а їхня модифікація сторонніми розробниками обмежена.
2. Висока вартість. Це робить подібні рішення малодоступними для невеликих судноплавних компаній.

3. Проблеми з інтеграцією старого обладнання. На практиці на судах одночасно працюють старі та нові протоколи, що ускладнює обмін даними [24].

4. Кібербезпека. Незважаючи на стандарти ІЕС 61162-460, у ряді робіт [25], [26] наголошується на недостатній захищеності суднових інформаційних систем від кіберзагроз.

Українські науковці приділяють значну увагу створенню алгоритмів для діагностики та прогнозування стану суднових енергетичних установок. Наприклад, у роботі [27] запропоновано методи контролю технічного стану дизель-генераторів, які базуються на аналізі трендів основних параметрів. У статті [28] розглянуто підходи до інтеграції сенсорних систем моніторингу з єдиною інформаційною платформою. Автори [29] наголошують на важливості врахування людського фактору при розробці інтерфейсу програмного забезпечення, оскільки саме від зручності та зрозумілості інтерфейсу часто залежить ефективність експлуатації системи.

Важливим напрямом в українських дослідженнях є також питання адаптації програмного забезпечення до умов роботи в обмежених обчислювальних середовищах [22], [27]. Це особливо актуально для невеликих суден, де немає потужних серверів, а більшість процесів має виконуватися у режимі реального часу.

На основі аналізу можна зробити висновок, що сучасні системи автоматизації СЕЕС вже досягли значного рівня розвитку, проте залишаються завдання, які потребують вирішення. Серед основних напрямів подальших досліджень і розробок можна виділити:

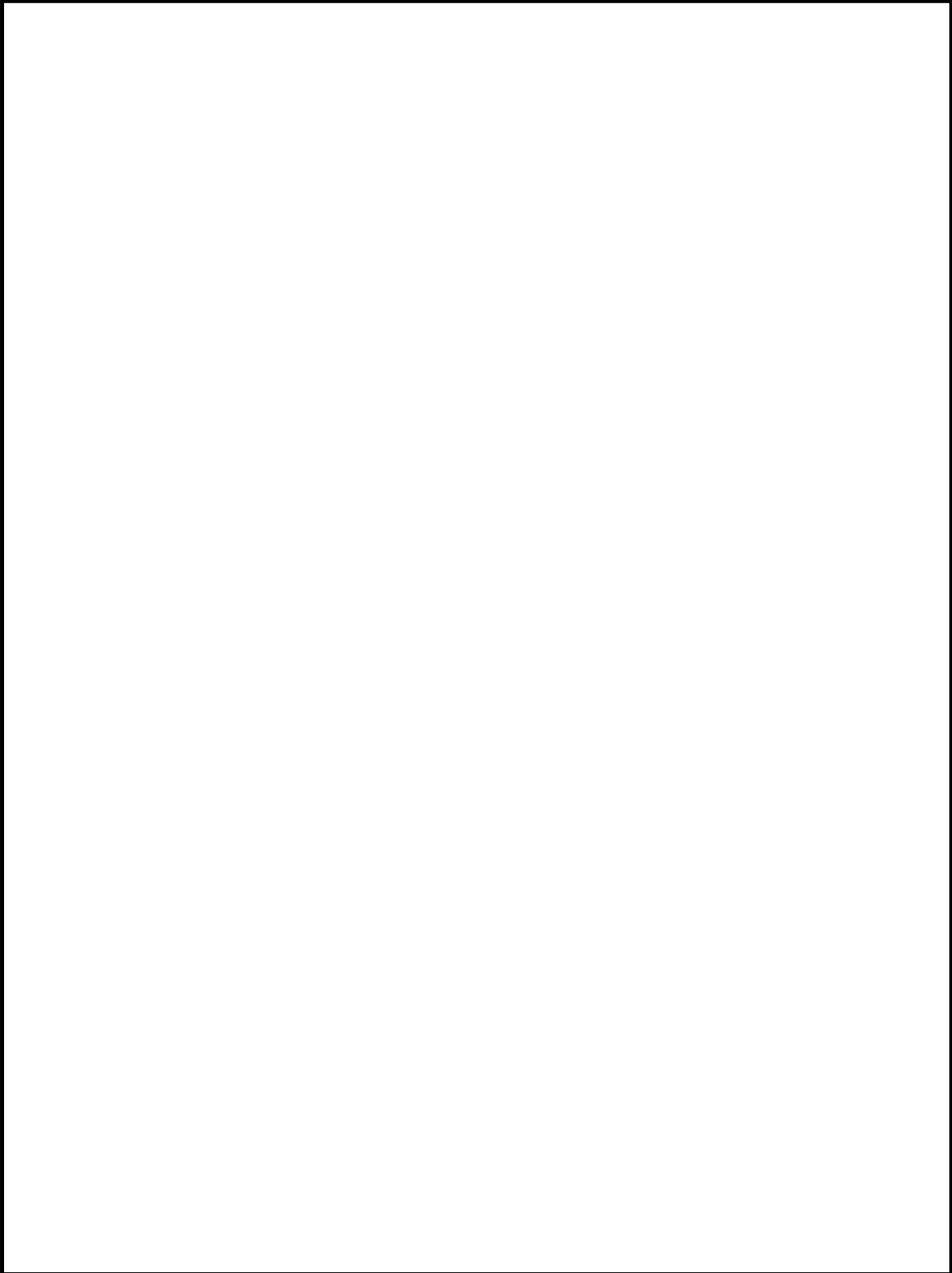
- створення відкритих і доступних рішень із можливістю модифікації під конкретні завдання;
- підвищення рівня кіберзахисту суднових інформаційних систем;
- розробку алгоритмів прогнозової діагностики для переходу від планового обслуговування до обслуговування «за станом»;

					171.6321м.КР.ПЗ.Р1	Арк.
						22
Змн.	Арк.	№ документа	Підпис	Дата		

- використання методів машинного навчання та інтелектуального аналізу даних для більш точного прогнозування відмов;
- розвиток інтерфейсів людино-машинної взаємодії, що враховують ергономічні аспекти та мінімізують ймовірність операторських помилок.

Таким чином, аналіз існуючих програмних засобів показав, що вони здатні забезпечити високий рівень автоматизації та безпеки, проте потребують подальшої адаптації до нових викликів – цифровізації, кіберзахисту та інтеграції інтелектуальних технологій. Для України особливо актуальним є створення доступних рішень, які можна ефективно застосовувати на різних типах суден із урахуванням специфічних умов їх експлуатації.

					171.6321м.КР.ПЗ.Р1	Арк.
						23
Змн.	Арк.	№ документу	Підпис	Дата		



					171.6321м.КР.ПЗ.Р2			
Змн.	Арк.	№ документа	Підпис	Дата	Проектування програмного забезпечення для систем керування судновим синхронним генератором	Літ.	Арк.	Аркуші
Розробник		Івановський С.А.						
Консультант							24	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Ушкаренко О.О.						
Зав. каф.		Рябенський В. М.						

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМ КЕРУВАННЯ СУДНОВИМ СИНХРОННИМ ГЕНЕРАТОРОМ

2.1. Опис апаратних засобів вбудованої мікропроцесорної системи керування судновим синхронним генератором

У самозбудному синхронному генераторі (ССГ) живлення обмотки збудження ротора здійснюється за допомогою статичного перетворювача. На принциповій схемі (рис. 2.1) наведено процес самозбудження, що реалізується через випрямлений струм, сформований діодним випрямлячем VD1, який отримує живлення від лінійної напруги генератора.

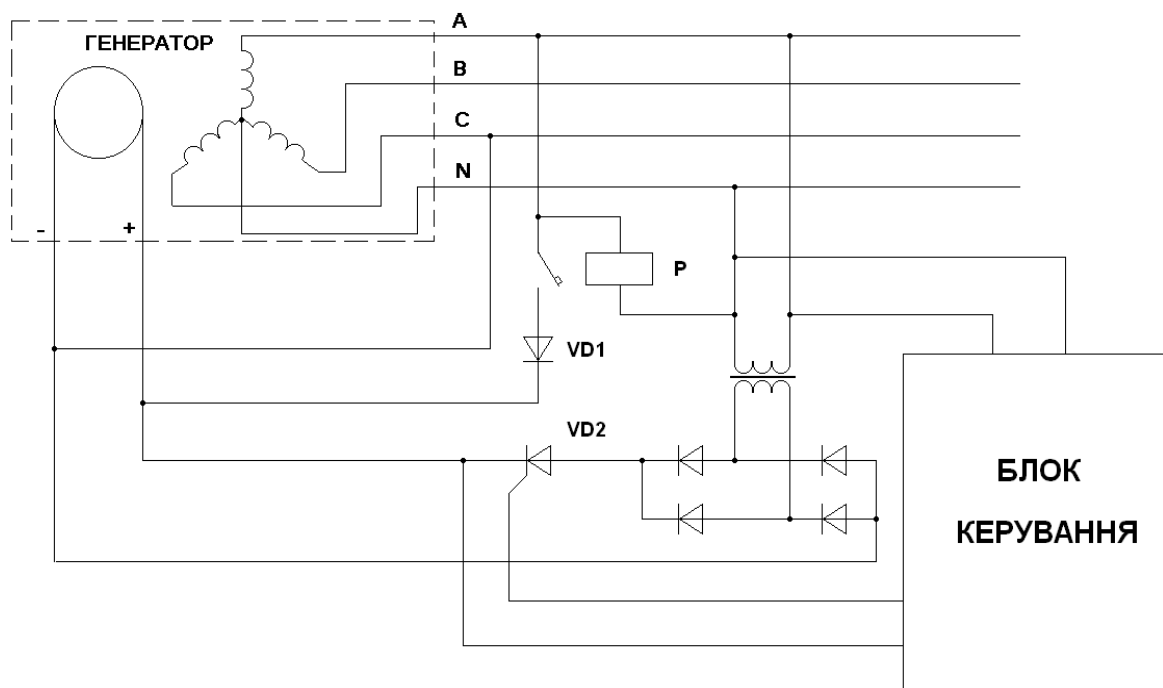


Рисунок 2.1 – Принципова схема самозбудження синхронного генератора

При розкручуванні генератора до номінальної частоти обертання (наприклад, за допомогою дизельного двигуна), за рахунок залишкового магнітного потоку на його виводах виникає ЕРС, що становить лише кілька відсотків від номінального значення. Цього недостатньо для спрацювання реле P, тому у початковому режимі діод VD1 підключається до кола обмотки

збудження. У результаті з'являється струм збудження i_b , який створює додатковий магнітний потік $\Phi_{дб}$. Сумарний потік визначається як:

$$\Phi_{рез} = \Phi_{ост} + \Phi_{дб}$$

Зростання магнітного потоку призводить до підвищення ЕРС і відповідно струму збудження. Процес має характер позитивного зворотного зв'язку та триває до тих пір, поки напруга генератора не досягне рівня холостого ходу $U_{хх}$. Подальший ріст обмежується опором кола збудження, який стає вищим за критичний.

Коли напруга генератора досягає значення $U_{ср}$, достатнього для спрацювання реле Р, ланцюг діода VD1 розмикається, і керування збудженням переходить під контроль системи автоматичного регулювання (САР). Від цього моменту живлення обмотки збудження здійснюється від фазної напруги через трансформатор напруги, випрямляч та тиристор VD2.

Принципова схема системи автоматичного регулювання збудження (рис. 2.2) базується на живленні від мережі змінного струму 220 В. Напруга надходить на понижувальний трансформатор Tr1.1, після чого випрямляється мостом VD1–VD4. Далі вона подається на інтегральний стабілізатор 7812, що формує стабільну напругу 12 В. Це живлення використовується для імпульсних трансформаторів та додаткового стабілізатора 7805, на виході якого формується напруга 5 В для цифрової частини системи.

					171.6321м.КР.ПЗ.Р2	Арк.
						26
Змн.	Арк.	№ документу	Підпис	Дата		

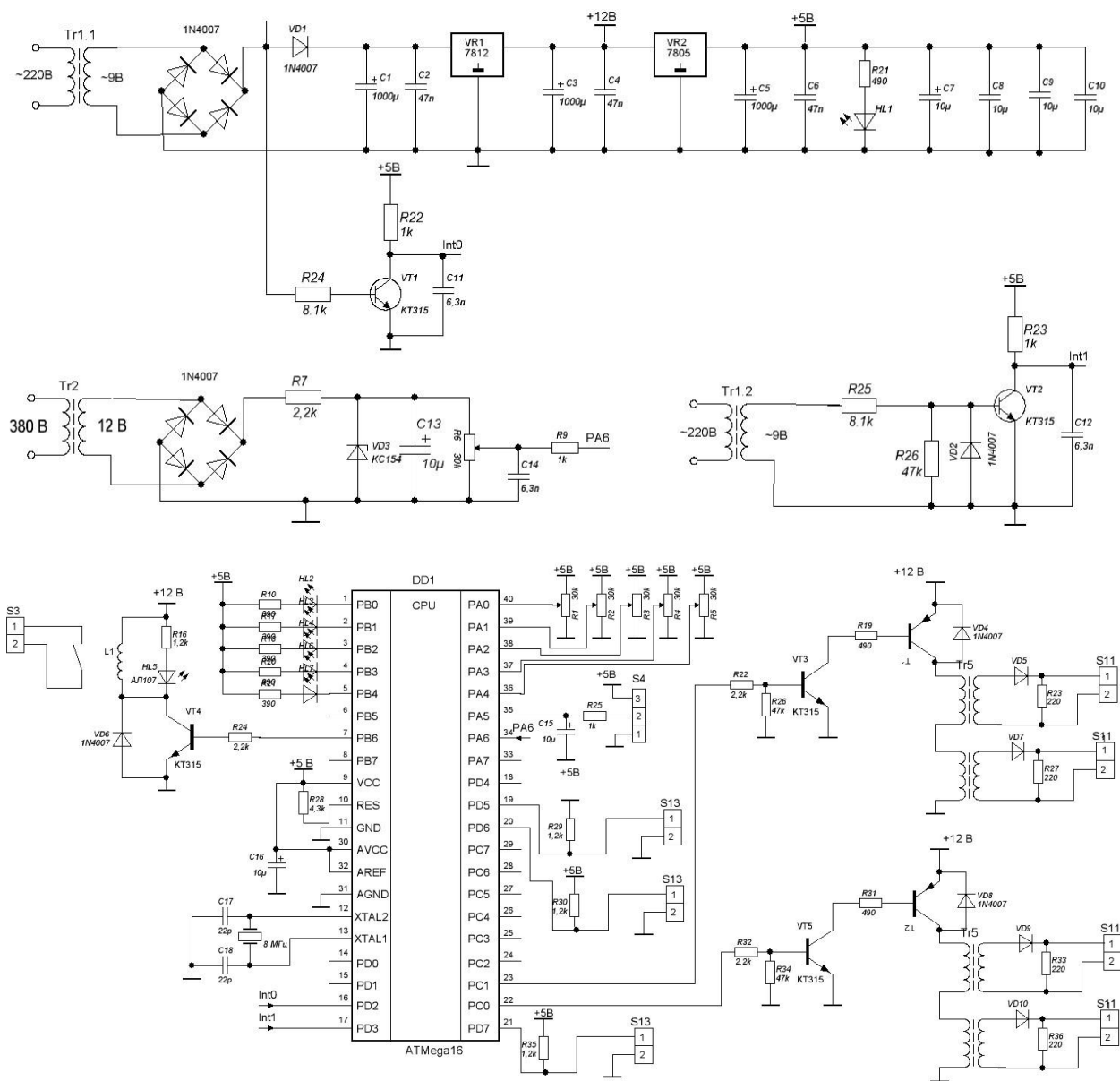


Рисунок 2.2 – Принципова схема вбудованої мікропроцесорної системи керування судновим синхронним генератором

Для синхронізації з мережею застосовується транзисторний каскад, з колектора якого імпульсні сигнали надходять на вхід зовнішнього переривання Int0 мікроконтролера. Базове керування транзистором здійснюється через струмообмежувальний резистор R24, який отримує сигнал із діодного випрямляча. Для визначення полярності напруги використовується трансформатор Tr1.2, вихід якого з'єднаний з базою

транзистора VT2. З його колектора формуються імпульси, що надходять на вхід Int1 мікроконтролера.

Налаштування параметрів регулювання, зокрема коефіцієнтів ПІД-регулятора, здійснюється за допомогою змінних резисторів R1–R6. Для гальванічної розв'язки силової частини від цифрової застосовуються імпульсні трансформатори.

Таким чином, апаратна частина вбудованої мікропроцесорної системи забезпечує самозбудження генератора, автоматичне регулювання процесу збудження, стабілізацію живлення електронних вузлів та коректну взаємодію мікроконтролера із силовою частиною.

На рисунку 2.3 наведено компонування елементів керування системи збудження, а також розташування вхідних клем для підключення живлення, сигналів зворотного зв'язку та зовнішніх органів керування. На рисунку 2.4 подано електричну схему підключення системи збудження.

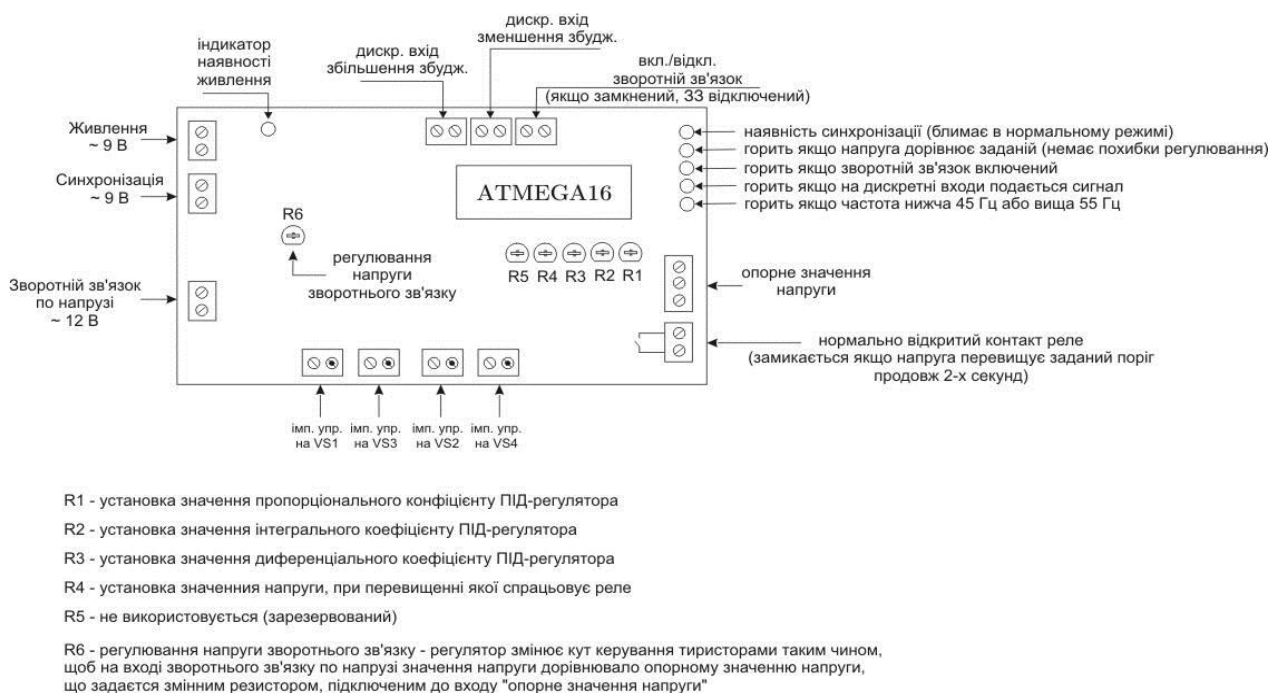


Рисунок 2.3 – Компонування елементів в системі та опис призначення входів

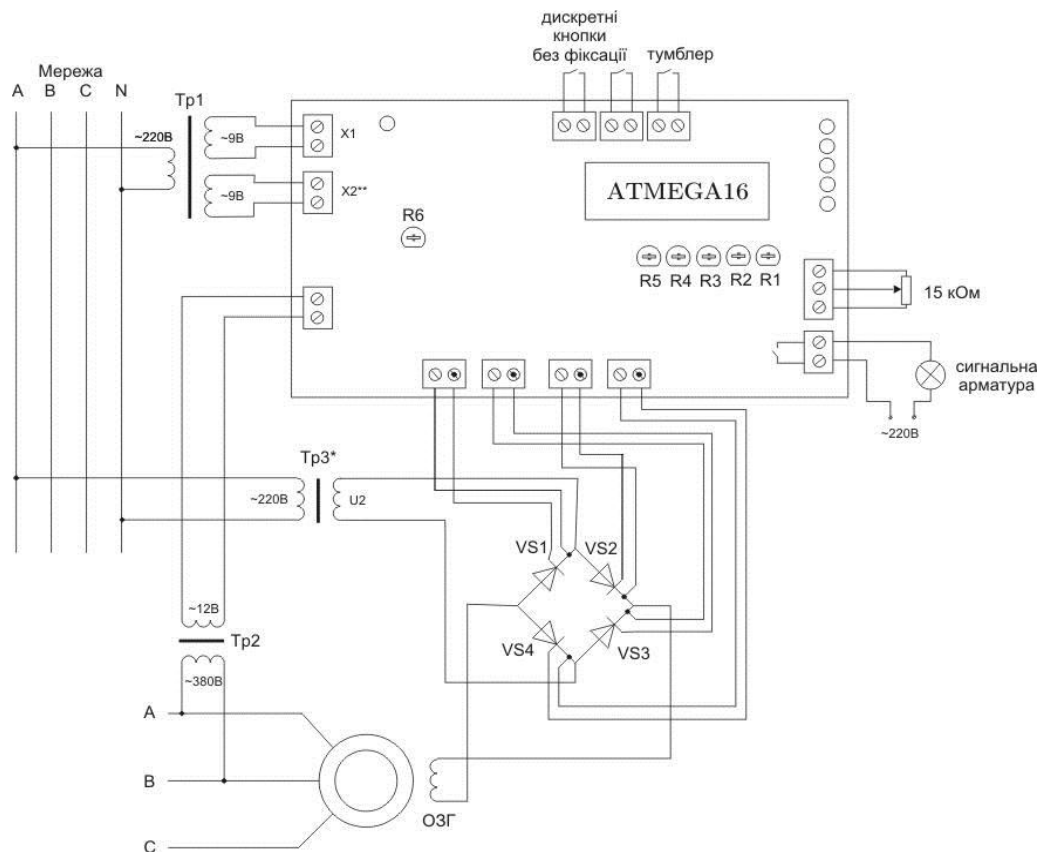


Рисунок 2.4 – Схема підключення вбудованої системи керування до суднового генератора

Регулювання коефіцієнтів ПІД-регулятора здійснюється за допомогою потенціометрів, розташованих на лицьовій панелі системи збудження. Така конструктивна реалізація забезпечує можливість оперативного налаштування параметрів регулятора обслуговуючим персоналом під час експлуатації, що підвищує ефективність і точність керування напругою збудження.

2.2. Розробка алгоритму програми налаштування параметрів регулятора системи керування ССГ

Програмна частина мікроконтролера, що входить до складу блоку керування збудженням суднового синхронного генератора, повинна забезпечувати виконання основних функцій системи автоматизації. До них належать:

- реалізація можливості вибору автоматичного або ручного режиму керування;
- підтримання вихідної напруги генератора на рівні, заданому користувачем;
- формування керуючих дій відповідно до ПД-закону регулювання.

Основним завданням програмного забезпечення є керування кутом відкриття тиристора, що потребує формування керуючих імпульсів. З точки зору алгоритму, мікроконтролер повинен генерувати на керуючий вивід тиристора імпульси прямокутної форми тривалістю близько 100 мкс. Це реалізується за допомогою вбудованих таймерів-лічильників.

Алгоритм роботи полягає в наступному: у регістр таймера заноситься значення, що відповідає тривалості імпульсу $t_{\text{имп}}$. Перед початком відліку вихід мікроконтролера встановлюється у високий логічний рівень. Після закінчення проміжку $t_{\text{имп}}$ генерується переривання від таймера, в обробнику якого вихід переводиться у низький стан. Таким чином формується керуючий сигнал на базу транзистора (рис. 2.5).

Не менш важливою складовою програми є реалізація алгоритму обчислення сигналу керування відповідно до дискретизованого ПД-закону регулювання. У неперервній формі цей закон описується рівнянням (2.1).

$$U(t) = K_p(e(t) + 1/T_N \int_0^t e dt + T_d \frac{de}{dt}) \quad (2.1)$$

Однак для використання в мікроконтролері доцільно застосовувати його дискретний аналог (2.2).

$$u(k) = K_p \left[e(k) + 1/T_N \sum_{i=0}^k e(i) T_x + T_d \frac{e(k) - e(k-1)}{T_x} \right] \quad (2.2)$$

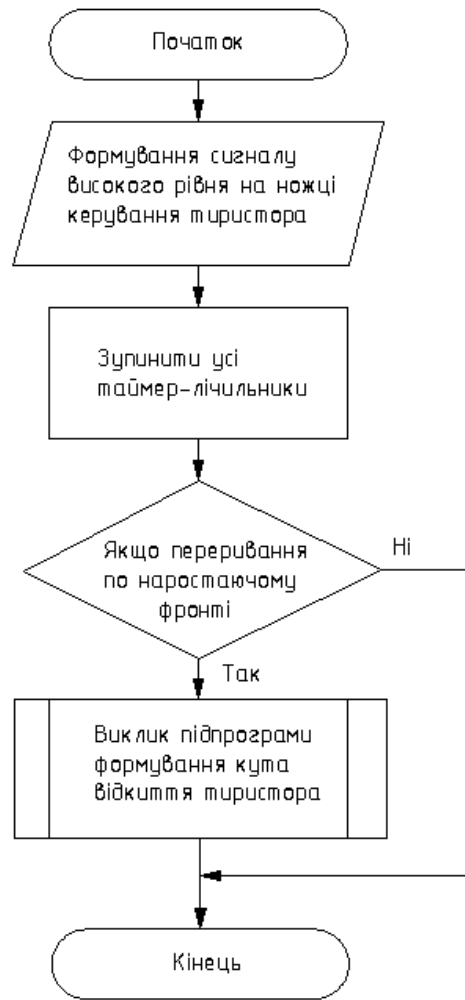


Рисунок 2.5 – Блок схема алгоритму створення імпульсу керування

Ще більш ефективною з точки зору обчислень є форма (2.3), де розраховується не сам сигнал керування, а його приріст.

$$u(k) = u(k-1) + K_p (d_0 e(k) + d_1 e(k-1) + d_2 e(k-2)) \quad (2.3)$$

де позначено

$$d_0 = 1 + \frac{T_k}{T_H} + \frac{T_d}{T_k}, \quad d_1 = -1 - 2 \frac{T_d}{T_k}, \quad d_2 = \frac{T_d}{T_k}$$

Це дозволяє значно зменшити обсяг обчислень і водночас зберегти необхідну якість регулювання (рис. 2.6).

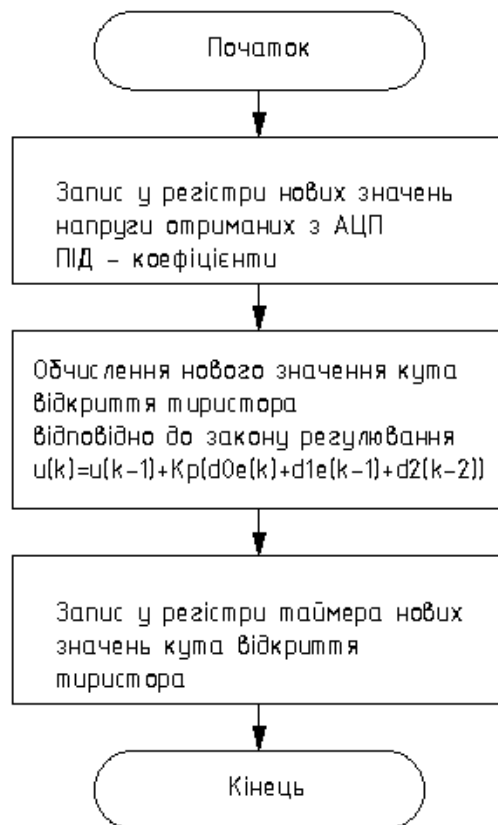


Рисунок 2.6 – Блок-схема алгоритму реалізації ПІД-закону керування ССГ

Коефіцієнти d_0 , d_1 , d_2 визначаються експериментально під час налагодження системи для конкретного генератора. Після їхнього підбору значення розраховуються та зберігаються у пам'яті мікроконтролера, залишаючись незмінними під час подальшої експлуатації. Такий підхід дозволяє оптимізувати використання обчислювальних ресурсів, оскільки основна операція зводиться до множення, яке апаратно підтримується мікроконтролером.

Проміжні значення, необхідні для реалізації алгоритму (попередній сигнал керування $u(k-1)$, похибки $e(k)$, $e(k-1)$, $e(k-2)$, доцільно зберігати у оперативній пам'яті (стеку). Для обраного мікроконтролера, що має достатній обсяг оперативної пам'яті, це не створює обмежень для реалізації обчислювальних процедур.

2.3. Розробка програмного забезпечення автоматизованого робочого місця оператора для керування судновим синхронним генератором

Розробка програмного забезпечення здійснювалася з урахуванням вимог до суднових вбудованих систем: надійності, відмовостійкості, зручності експлуатації в умовах обмеженого простору машинного відділення. Програмний інтерфейс було спроектовано у вигляді інтуїтивно зрозумілого віконного додатку з інтегрованими елементами керування, орієнтованими на персонал середньої кваліфікації.

На етапі проєктування особлива увага приділялася:

- мінімізації часу реакції системи на зміну налаштувань;
- реалізації алгоритмів синхронізації між ПЗ та мікроконтролером;
- забезпеченню стійкого обміну даними через послідовний інтерфейс;
- інтеграції функцій моніторингу (візуалізація параметрів, відображення динаміки зміни напруги);
- можливості подальшої модернізації для збору статистики та діагностичних даних з метою їх збереження у базі даних.

Завдяки цьому програмні засоби не лише виконують функцію налаштування ПД-регулятора, а й стають частиною комплексної системи збору та обробки експлуатаційних даних суднового синхронного генератора. Це відкриває можливості для інтеграції у більш широкі інформаційні системи судна, зокрема системи технічної діагностики та централізованого енергетичного менеджменту.

Перед початком роботи з програмними засобами керування необхідно перевірити правильність апаратного з'єднання блоку керування з персональним комп'ютером. Для організації обміну даними застосовується нуль-модемний кабель, який забезпечує надійний двосторонній зв'язок між мікроконтролером системи збудження та комп'ютером. На початковому етапі користувач у вікні програми обирає СОМ-порт, до якого підключено пристрій, і натискає кнопку «Відкрити порт». Після встановлення зв'язку

					171.6321м.КР.ПЗ.Р2	Арк.
						33
Змн.	Арк.	№ документа	Підпис	Дата		

інтерфейс підтверджує з'єднання: список доступних портів блокується, а кнопка змінює назву на «Закрити порт».

Далі оператор запускає приводний двигун генератора за допомогою органів керування на передній панелі системи. Після досягнення номінальної частоти обертання активується живлення обмотки збудження. Це супроводжується вмиканням сигнальної лампи жовтого кольору, що свідчить про готовність системи до подальшої роботи у режимі автоматичного регулювання.

Програмні засоби (рис. 2.7) забезпечують зручний графічний інтерфейс для налаштування й контролю параметрів системи збудження.

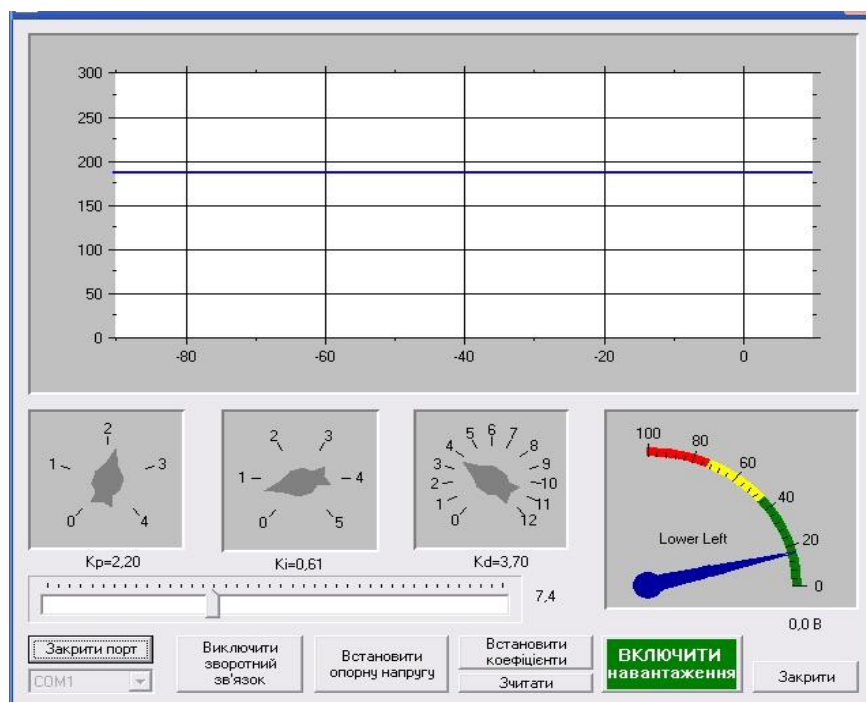


Рисунок 2.7 – Інтерфейс програмних засобів керування, збору та обробки даних для суднового синхронного генератора

Оператор має можливість у інтерактивному режимі задавати значення коефіцієнтів ПД-регулятора (K_p , K_i , K_d) за допомогою стилізованих елементів керування. Після натискання кнопки «Встановити коефіцієнти» мікроконтролер переходить на роботу з новими налаштуваннями.

Передбачено також зворотну операцію – зчитування актуальних параметрів («Зчитати»), при якій графічні елементи автоматично синхронізуються з даними, що зберігаються у пам'яті системи.

Опорне значення напруги задається через елемент «Повзунок» (Slider Control), тоді як поточне значення напруги відображається у вікні, стилізованому під осцилограф. Це дозволяє в реальному часі спостерігати за зміною вихідної напруги генератора, відслідковувати появу коливань та оцінювати якість налаштування регулятора. Таким чином програмні засоби виконують не лише функцію керування, але й функцію збору та первинної обробки даних про роботу генератора, що є надзвичайно важливим у контексті суднових енергетичних систем.

2.4. Програмна реалізація регулятора та методика його налаштування

Програмна реалізація алгоритму ПД-регулятора вбудованої мікропроцесорної системи ССГ побудована таким чином, щоб забезпечити обчислення пропорційної, інтегральної та диференційної складових керуючого сигналу в режимі реального часу. В основі лежить функція, яка на вхід отримує сигнал похибки та повертає значення керуючої дії для силової частини системи збудження.

У функції передбачено:

- виділення пам'яті для збереження проміжних значень складових регулятора (пропорційної, інтегральної, диференційної);
- масштабування коефіцієнтів K_p , K_i , K_d , що зчитуються через вбудовані аналого-цифрові перетворювачі мікроконтролера;
- розрахунок пропорційної складової шляхом множення коефіцієнта K_p на сигнал похибки;
- визначення диференційної складової на основі різниці між поточним та попереднім значеннями похибки;

					171.6321м.КР.ПЗ.Р2	Арк.
						35
Змн.	Арк.	№ документа	Підпис	Дата		

- накопичення інтегральної складової із захистом від явища «перенасичення» (антивіндап);
- формування підсумкового значення керуючого сигналу як суми трьох складових.

Для підвищення надійності алгоритму реалізовано перевірку вихідного сигналу на перевищення граничних значень. У разі виходу за допустимі межі спрацьовує механізм обмеження (сатурація), що гарантує коректну роботу силової частини та захист тиристорного перетворювача від аварійних режимів.

Особливістю реалізації є використання апаратної підтримки операцій множення у мікроконтролері сімейства AVR, що дозволяє значно зменшити час виконання обчислень. Для зберігання значень похибок у попередні моменти часу ($e(k)$, $e(k-1)$, $e(k-2)$) та проміжних керуючих дій використовується оперативна пам'ять (стек), обсяг якої є достатнім для реалізації дискретного алгоритму.

Налаштування коефіцієнтів ПД-регулятора виконується під час експлуатації за допомогою зовнішніх програмних засобів, які забезпечують інтерактивний доступ до параметрів регулятора. Користувач має змогу задати значення K_p , K_i , K_d , через інтерфейс керування, після чого нові значення автоматично завантажуються у мікроконтролер.

Методика налаштування передбачає:

1. Встановлення пропорційної складової – підбір коефіцієнта K_p , що визначає швидкість реакції системи на відхилення.
2. Підбір інтегральної складової (K_i) – забезпечує усунення статичної похибки, проте потребує обмеження для запобігання перенасиченню інтегратора.
3. Налаштування диференційної складової (K_d) – формує демпфуючий ефект, зменшуючи коливання та переналаштування системи.

					171.6321м.КР.ПЗ.Р2	Арк.
						36
Змн.	Арк.	№ документа	Підпис	Дата		

4. Експериментальну перевірку – визначення оптимальних параметрів для конкретного суднового генератора шляхом аналізу перехідних процесів на виході.

Підібрані параметри зберігаються у пам'яті мікроконтролера, залишаючись незмінними при наступних сеансах роботи системи. Це дозволяє уникнути повторного налаштування та забезпечує стабільність режимів регулювання. Текст програми наведено у Додатку А.

У результаті така програмна реалізація регулятора забезпечує:

- підтримання стабільної напруги на виході суднового синхронного генератора;
- можливість оперативного налаштування під конкретні умови експлуатації;
- оптимальне використання обчислювальних ресурсів мікроконтролера;
- інтеграцію з програмними засобами збору та обробки даних, що відповідає концепції розробки сучасних суднових вбудованих систем.

Висновки до розділу

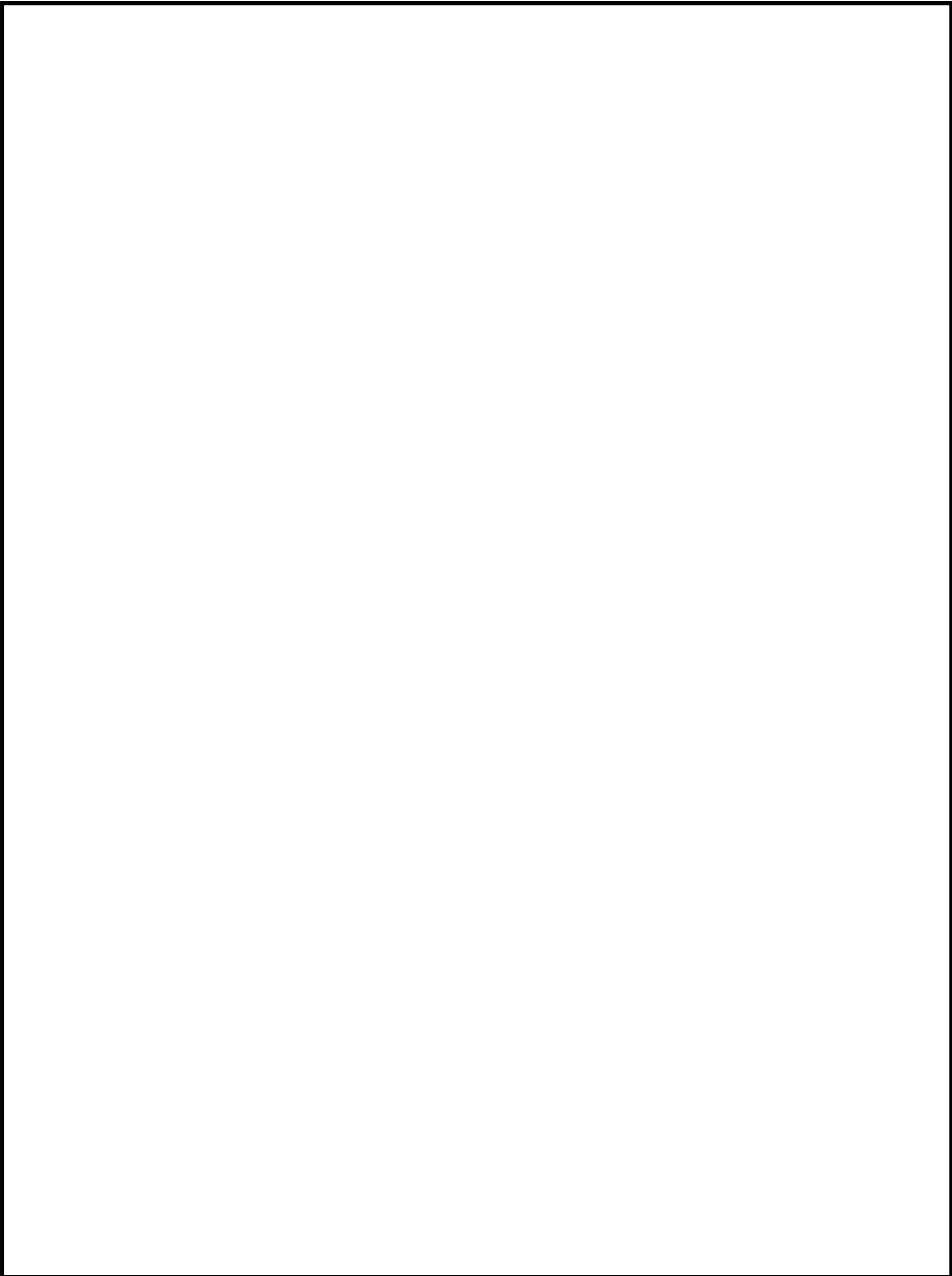
У розділі проаналізовано принципи роботи та побудови вбудованої системи керування судновим синхронним генератором, яка забезпечує стабільну напругу та передачу керування до мікропроцесорної системи керування, збору та обробки даних. Програмна реалізація ПІД-регулятора дозволяє обчислювати пропорційну, інтегральну та диференціальну складові в реальному часі з урахуванням обмежень на вихідний сигнал і захисту інтегратора від перенасичення. Використання апаратної підтримки множення забезпечує точність та швидкодію регулювання. Налаштування регулятора здійснюється експериментально через розроблений програмний інтерфейс, що дозволяє оптимізувати коефіцієнти для конкретних умов експлуатації та

					171.6321м.КР.ПЗ.Р2	Арк.
						37
Змн.	Арк.	№ документа	Підпис	Дата		

підтримувати стабільну роботу вбудованої системи керування судновим синхронним генератором.

Таким чином, інтеграція апаратних і програмних компонентів у рамках мікропроцесорної системи керування, збору та обробки даних забезпечує ефективне керування генератором, високу надійність та точність регулювання, а також створює основу для подальшої інтеграції в комплексні автоматизовані судові системи.

					171.6321м.КР.ПЗ.Р2	Арк.
						38
Змн.	Арк.	№ документу	Підпис	Дата		



					171.6321м.КР.ПЗ.РЗ			
Змн.	Арк.	№ документа	Підпис	Дата				
Розробник		Івановський С.А.			Реалізація програмного забезпечення для забезпечення паралельної роботи суднового синхронного	Літ.	Арк.	Аркуші
Консультант							39	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Ушкаренко О.О.						
Зав. каф.		Рябенський В. М.						

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАБЕЗПЕЧЕННЯ ПАРАЛЕЛЬНОЇ РОБОТИ СУДНОВОГО СИНХРОННОГО ГЕНЕРАТОРА

3.1. Програмно-апаратне забезпечення підключення генератора до загального навантаження

Блок системи синхронізації є ключовим елементом вбудованої мікропроцесорної системи керування судновими синхронними генераторами, адже від його роботи залежить стабільність та надійність функціонування всієї електроенергетичної мережі судна. Підключення до генераторів здійснюється через стандартні вимірювальні трансформатори напруги та струму, після чого блок синхронізації формує сигнал на замикання контактів автоматичного вимикача. Це дозволяє здійснювати введення генератора у паралельну роботу з іншими агрегатами за умови дотримання встановлених критеріїв синхронізації.

До основних умов синхронізації належить узгодженість частот, амплітуд напруг та фазових кутів напруги генераторів. Ідеально рівні параметри на практиці недосяжні, тому використовується підхід, що допускає роботу з невеликими відхиленнями: різниця фаз $\Delta\varphi \leq 10^\circ$, відхилення напруг $\Delta U \leq 5\%$, різниця частот $\Delta f \leq 1,5\%$. Дотримання цих меж гарантує відсутність небезпечних перехідних процесів і забезпечує плавний розподіл навантаження.

Недоліком більш ранніх синхронізаторів було значне збільшення часу вирівнювання частот, оскільки керуючий сигнал формувався періодично і не враховував реальної величини Δf . Це призводило до небажаних затримок або навіть до явища «зависання», коли при малій різниці частот підключення генератора ставало неможливим. Запропонований у рамках даної роботи алгоритм мікропроцесорної синхронізації усуває ці проблеми завдяки безперервному вимірюванню Δf та формуванню сигналів керування у реальному часі. При цьому у дозволеному інтервалі для включення

					171.6321м.КР.ПЗ.РЗ	Арк.
						40
Змн.	Арк.	№ документу	Підпис	Дата		

автоматичного вимикача сигнали на корекцію частоти не видаються, що запобігає помилкам і скорочує час синхронізації.

Особливу увагу приділено усуненню негативних наслідків так званого «негативного ковзання» ($\Delta f < 0$), коли підключуваний генератор може перейти у режим двигуна і викликати спрацювання захисту від зворотної потужності. Запропонована система працює виключно при позитивному ковзанні ($\Delta f > 0$), що гарантує коректність синхронізації.

У розробленому блоці враховано також затримку автоматичного вимикача (10–90 мс залежно від його типу), що дозволяє точно визначати момент подачі сигналу на замикання контактів. На вході блок синхронізації отримує сигнали від формувачів імпульсів при переході напруги через нуль, а на виході формує команди на зміну збудження, регулювання обертів первинного двигуна та підключення генератора до мережі.

Реалізація такого алгоритму стала можливою завдяки використанню сучасної мікропроцесорної та мікроелектронної бази, що забезпечила високу швидкодію, компактність апаратної реалізації та простоту інтеграції у систему збору та обробки даних. Це дозволяє підвищити надійність, точність та безпеку роботи суднових енергетичних установок, а також забезпечує сумісність із автоматизованим робочим місцем оператора для моніторингу та налаштування параметрів синхронізації.

На рис. 3.1 наведено принципову схему системи підключення генератора до загального навантаження.

					171.6321м.КР.ПЗ.РЗ	Арк.
						41
Змн.	Арк.	№ документу	Підпис	Дата		

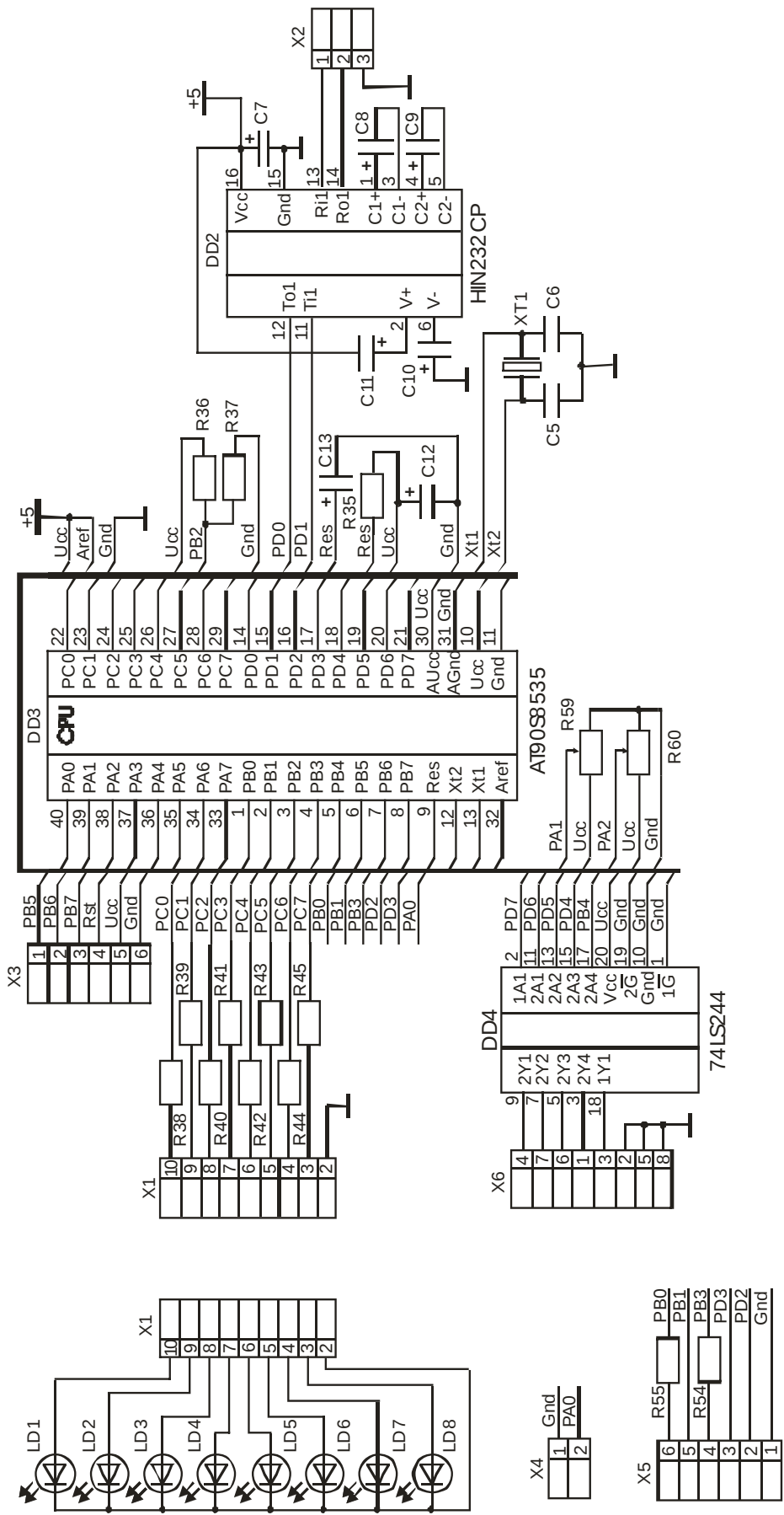


Рисунок 3.1 – Принципова схема системи підключення генератора до загального навантаження

Програмно-апаратна реалізація блоку керування паралельним підключенням суднового синхронного генератора функціонує у режимі безперервно повторюваних обчислювальних циклів. У кожному циклі мікропроцесорна система виконує визначення величини та знаку різниці частот (Δf), обчислює зсув фаз ($\Delta \varphi$), а також оцінює швидкість зміни частоти і фазового кута. На основі цих даних формується рішення щодо оптимального моменту подачі команди на замикання контактів автоматичного вимикача.

Точність вимірювання різниці частот забезпечується цифровим принципом обробки сигналів. Похибка визначення Δf , яка зумовлена дискретністю вимірювального процесу та можливими коливаннями частоти імпульсів формувача, не перевищує $\pm 0,08$ Гц, що є достатнім для надійного функціонування системи. При цьому нестабільність опорних імпульсів практично не впливає на кінцевий результат, оскільки алгоритм передбачає корекцію похибок.

Загальний принцип дії блоку можна описати таким чином: система формує послідовність імпульсів синхронізації, вимірює їх тривалість для визначення фазового зсуву, ідентифікує фазову гілку (зростаючу чи спадну) та порівнює отримані результати з допустимими уставками. Якщо всі критерії синхронізації виконані (різниця фаз, частот та напруг знаходяться у межах допустимих значень), формується вихідний сигнал для ввімкнення автоматичного вимикача, що здійснює підключення генератора на паралельну роботу. У випадку відхилення хоча б одного з параметрів за межі допустимих значень, команда на підключення не подається.

Таким чином, запропонований підхід дозволяє реалізувати точний та надійний алгоритм автоматизованого введення суднового синхронного генератора у паралельну роботу в складі вбудованої мікропроцесорної системи керування, збору та обробки даних.

					171.6321м.КР.ПЗ.РЗ	Арк.
						43
Змн.	Арк.	№ документа	Підпис	Дата		

3.2. Обґрунтування параметрів дільників для забезпечення роботи таймерів-лічильників

Як зазначалося у попередньому параграфі, мікроконтролер виконує функції центрального керуючого елемента вбудованої мікропроцесорної системи керування судновим синхронним генератором. Програмна логіка його роботи реалізована на принципах асинхронного виконання, тоді як синхронізація окремих процесів досягається через спеціальні події. До таких подій відносяться: зовнішні переривання, сигнали від аналогового компаратора, події переповнення таймерів-лічильників, перетворення вбудованого АЦП, а також обмін даними через UART.

Тактовою основою для роботи мікроконтролера є кварцовий генератор із частотою 7.3728МГц, яка у багато разів перевищує робочі частоти досліджуваних сигналів. Це створює необхідність попереднього розрахунку оптимальних значень дільників для таймерів-лічильників, аналого-цифрового перетворювача та універсального асинхронного приймача-передавача з метою узгодження часових параметрів апаратної платформи з характеристиками електричних сигналів генераторів.

Мікроконтролер має у своєму складі три таймери-лічильники. Два з них застосовуються для вимірювання частот основного та підключуваного генераторів. Для коректної роботи прийнято значення дільника, рівне 256, що дозволяє отримати робочу частоту таймерів за співвідношенням (3.1).

$$f_T = \frac{f_{\kappa}}{256} = 28800(\Gamma\text{ц}). \quad (3.1)$$

Період та час переповнення таймера розраховуються відповідно до формул (3.2-3.3), що дає можливість зіставити кількість відліків лічильника із реальною частотою напруги генератора.

$$T_T = \frac{1}{f_T} = \frac{1}{28800} = 3,472 \cdot 10^{-5}(\text{с}). \quad (3.2)$$

$$T_{\text{пер}} = T_T \cdot 2^8 = 1,389 \cdot 10^{-4} \cdot 256 = 0,00889(\text{с}). \quad (3.3)$$

					171.6321м.КР.ПЗ.РЗ	Арк.
						44
Змн.	Арк.	№ документа	Підпис	Дата		

Для частоти 50 Гц, а також для граничних відхилень (48 Гц і 52 Гц), визначаються відповідні значення таймерів (3.4-3.6).

$$N_{50} = \frac{256}{T_{\text{пер}} \cdot f_{\text{ген}}} = \frac{256}{0,00889 \cdot 50} = 576, \quad (3.4)$$

$$N_{48} = \frac{256}{T_{\text{пер}} \cdot f_{48}} = \frac{256}{0,00889 \cdot 48} = 600, \quad (3.5)$$

$$N_{52} = \frac{256}{T_{\text{пер}} \cdot f_{52}} = \frac{256}{0,00889 \cdot 52} = 553. \quad (4.6)$$

На основі цих величин розраховано абсолютну та відносну похибки вимірювання частоти (3.7-3.8), які підтверджують достатню точність запропонованого підходу.

$$\Delta f = \frac{f_{50} - f_{48}}{N_{48} - N_{50}} = \frac{50 - 48}{600 - 576} = \frac{2}{24} = 0,083(\text{Гц}). \quad (3.7)$$

$$\delta = \frac{\Delta f}{f_{50}} = \frac{0,083}{50} = 0,17\%. \quad (3.8)$$

Третій таймер-лічильник використовується для визначення різниці фаз між напругами генераторів. Його розрядність становить 16 біт, що забезпечує можливість роботи з великим часовим діапазоном. Обравши дільник частоти 64, отримуємо робочу частоту таймера (3.9) та тривалість одного тактового імпульсу (3.10).

$$f_{T2} = \frac{f_{\text{кв}}}{64} = \frac{7372800}{64} = 1,152 \cdot 10^5 (\text{Гц}), \quad (3.9)$$

$$T_2 = \frac{1}{f_{T2}} = \frac{1}{1,152 \cdot 10^5} = 8,68 \cdot 10^{-6} (\text{с}). \quad (3.10)$$

Далі визначаються часові інтервали, що відповідають допустимим фазовим відхиленням: від 10 електричних градусів (межа для синхронізації) до 180 електричних градусів (максимально можливий зсув). На основі цього розраховуються значення таймера (3.12-3.13) та похибки визначення фазових параметрів (3.14-3.15).

$$t_{10} = \frac{\varphi}{360 \cdot f_{\text{ген}}} = \frac{10}{360 \cdot 50} = 5,56 \cdot 10^{-4} (\text{с}). \quad (3.11)$$

					171.6321м.КР.ПЗ.РЗ	Арк.
						45
Змн.	Арк.	№ документа	Підпис	Дата		

$$N_{10} = \frac{t_{10}}{T_2} = \frac{5,56 \cdot 10^{-6}}{8,68 \cdot 10^{-6}} = 64. \quad (3.12)$$

$$N_{180} = N_{10} \cdot \frac{180}{10} = 1152. \quad (3.13)$$

$$\Delta\varphi = \frac{10}{64} = 0,156(\text{град}). \quad (3.14)$$

$$\delta = \frac{\Delta\varphi}{180} \cdot 100\% = 0,087\%. \quad (3.15)$$

Для обміну даними з операторським автоматизованим робочим місцем використовується універсальний асинхронний приймач-передавач (UART). Швидкість передачі даних визначається за формулою (3.16).

$$B = \frac{f_{\text{кв}}}{16 \cdot (UBRR + 1)}, \quad (3.16)$$

Прийнявши значення 115200 біт/с, розраховано відповідний дільник (3.17). Така швидкість гарантує нульову похибку та дає змогу передати 8 байт інформації за час (3.18), який суттєво менший за період вимірювання частот генераторів. Це забезпечує своєчасне відображення на комп'ютері актуальних даних про частоти, фазові відхилення та напруги.

$$UBRR = \frac{f_{\text{кв}}}{16 \cdot B} - 1 = \frac{7372800}{16 \cdot 115200} - 1 = 3. \quad (3.17)$$

$$t_8 = \frac{8 \cdot 10}{115200} = 6,94 \cdot 10^{-4} (\text{с}). \quad (3.18)$$

Таким чином, розрахунок параметрів дільників таймерів-лічильників та UART підтверджує можливість ефективної реалізації алгоритмів мікропроцесорної системи керування, збору та обробки даних для автоматизованої синхронізації суднових синхронних генераторів.

3.3. Програмна організація алгоритму керування мікропроцесорною системою

Програмне забезпечення, розроблене для мікроконтролера, що є ядром вбудованої мікропроцесорної системи керування судновим синхронним генератором, побудоване на принципі структурної декомпозиції. Основна

					171.6321м.КР.ПЗ.РЗ	Арк.
						46
Змн.	Арк.	№ документу	Підпис	Дата		

логіка організована у вигляді циклу «отримати подію – обробити подію», що забезпечує безперервність роботи та високу надійність обробки сигналів.

Крім основного циклу, система передбачає асинхронне виконання обробників зовнішніх апаратних переривань, що дозволяє оперативно реагувати на критичні події. Програма поділяється на функціональні модулі, кожен з яких відповідає за окрему задачу: вимірювання частоти генераторів, визначення фазового зсуву, регулювання збудження, формування сигналів для зміни обертів дизель-генератора, а також видачу команди на замикання контактора за умови дотримання параметрів синхронізації.

Алгоритм роботи включає такі основні етапи:

- обчислення частоти працюючого та підключуваного генератора;
- аналіз різниці частот та формування відповідних керуючих сигналів;
- вимірювання фазового зсуву та його швидкості зміни;
- контроль напруги генератора, що вводиться в роботу;
- формування команди на включення автоматичного вимикача з урахуванням часу його спрацьовування;
- індикація поточного стану синхронізації;
- передача даних у комп'ютер через послідовний канал для відображення параметрів на автоматизованому робочому місці оператора.

Програмний алгоритм (рис. 3.2) не має чітко визначеного початку чи завершення – він активується одразу після запуску або перезапуску контролера, та повторює робочий цикл доти, доки не буде виконано умови синхронізації. Після видачі сигналу на замикання контактора відбувається заиклення, а повторний процес може бути ініційований командою від оператора.

Першим етапом роботи є ініціалізація апаратних ресурсів: реєстрів, стека та параметрів випередження для врахування затримки автоматичного вимикача. Значення цих уставок задаються за допомогою аналогових сигналів через АЦП, що надходять зі змінних резисторів на панелі пристрою.

					171.6321м.КР.ПЗ.РЗ	Арк.
						47
Змн.	Арк.	№ документа	Підпис	Дата		

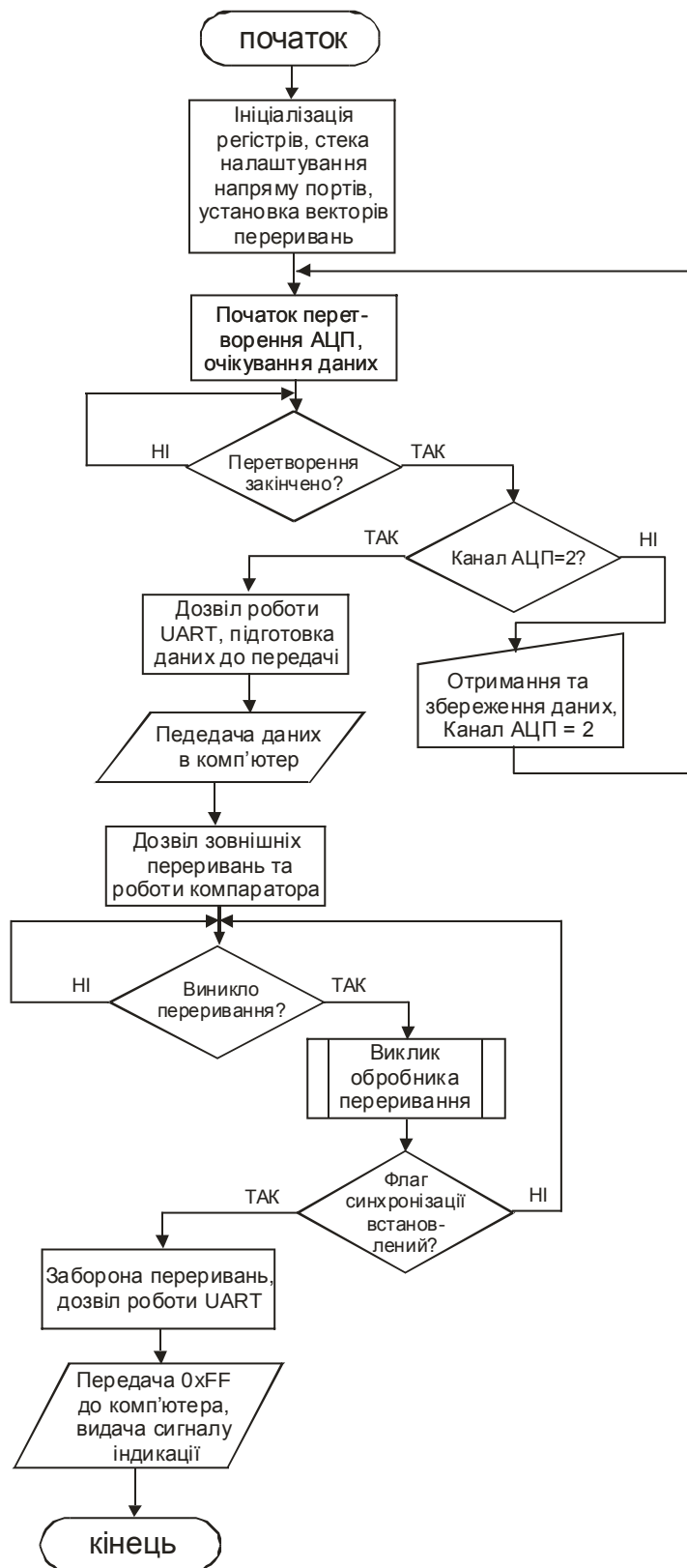


Рисунок 3.1 – Алгоритм роботи головного блоку програми

Далі активується процедура вимірювання частот. Це реалізується шляхом запуску таймерів-лічильників у моменти надходження зовнішніх

переривань. При цьому алгоритм перевіряє прапори вимірювання та готовності даних, забезпечуючи коректність результатів. Якщо виконуються умови допустимих відхилень, система дозволяє синхронізацію; у протилежному випадку формуються сигнали для корекції швидкості обертання дизеля. Блок-схема алгоритмк представлена на рис. 3.3.

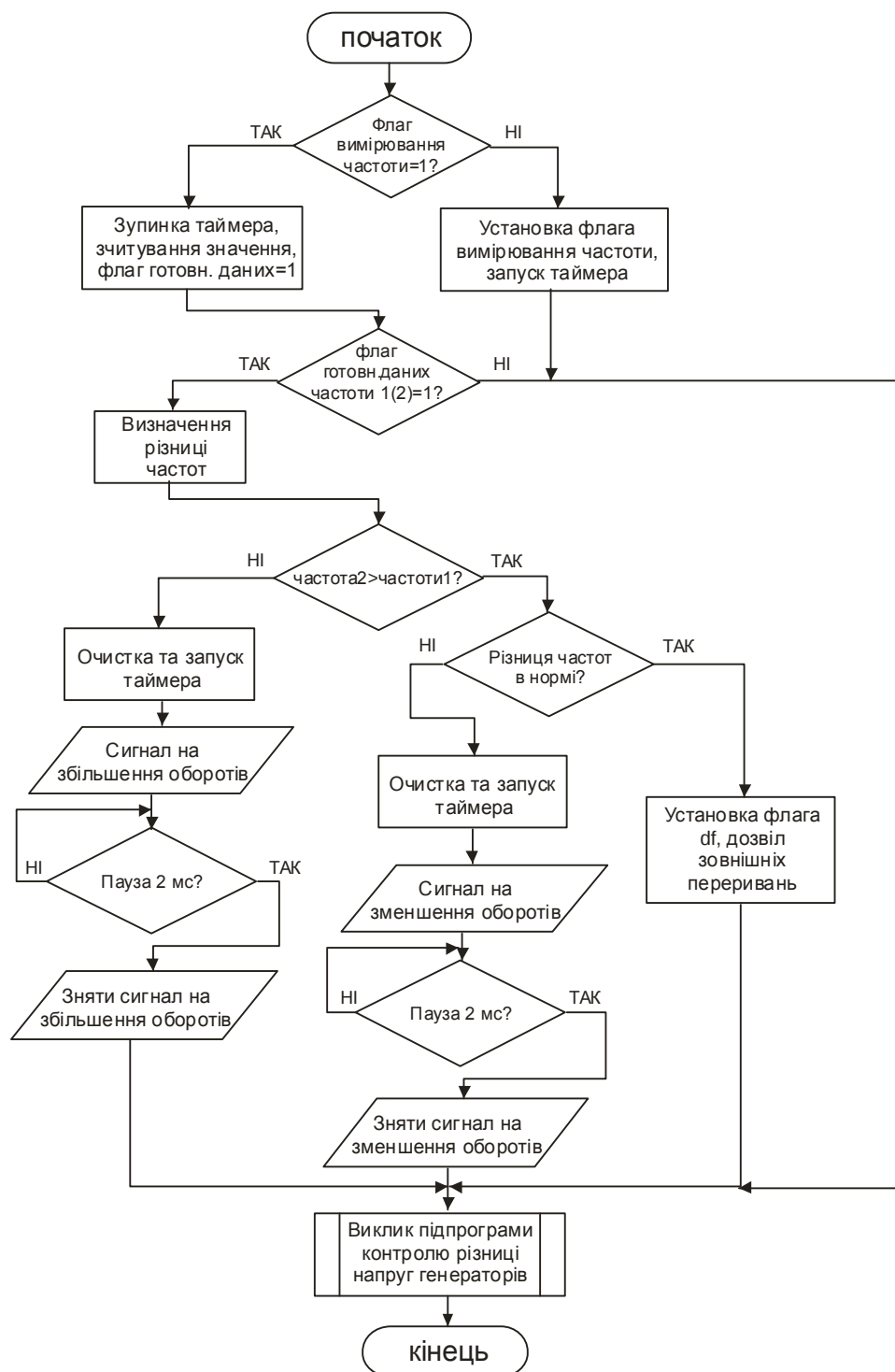


Рисунок 3.3 – Алгоритм підпрограми обробки зовнішнього переривання

На рис. 3.4 представлена блок-схема алгоритму роботи підпрограми визначення різниці напруг.

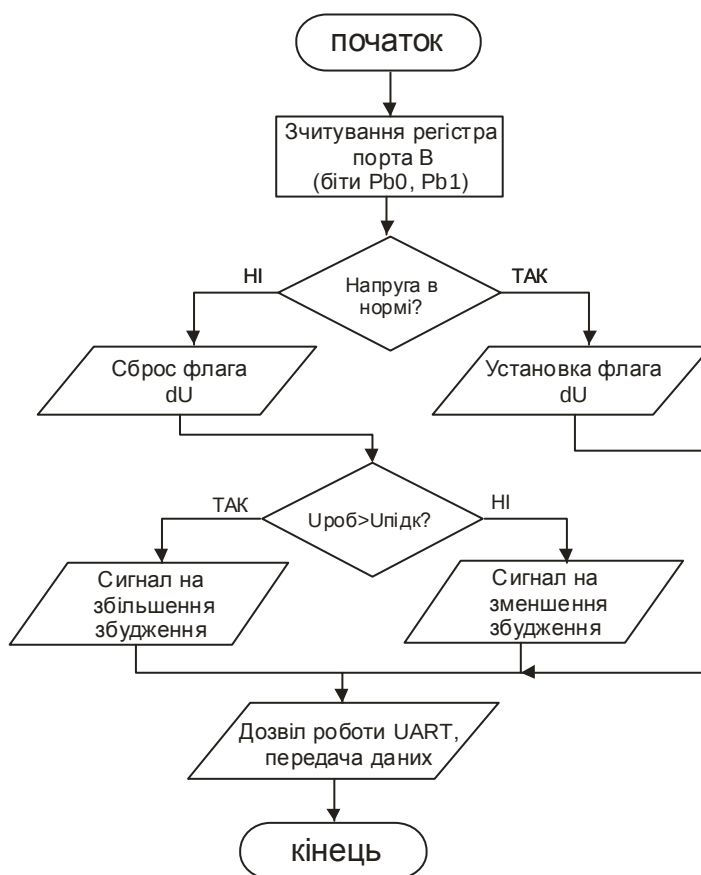


Рисунок 3.4 – Алгоритм підпрограми визначення різниці напруг

Розрахунок різниці фаз, а також першої похідної різниці фаз виконується в обробнику переривання від аналогового компаратора. Алгоритм роботи оброблювача переривання від вбудованого аналогового компаратора наведено на рис. 3.5.

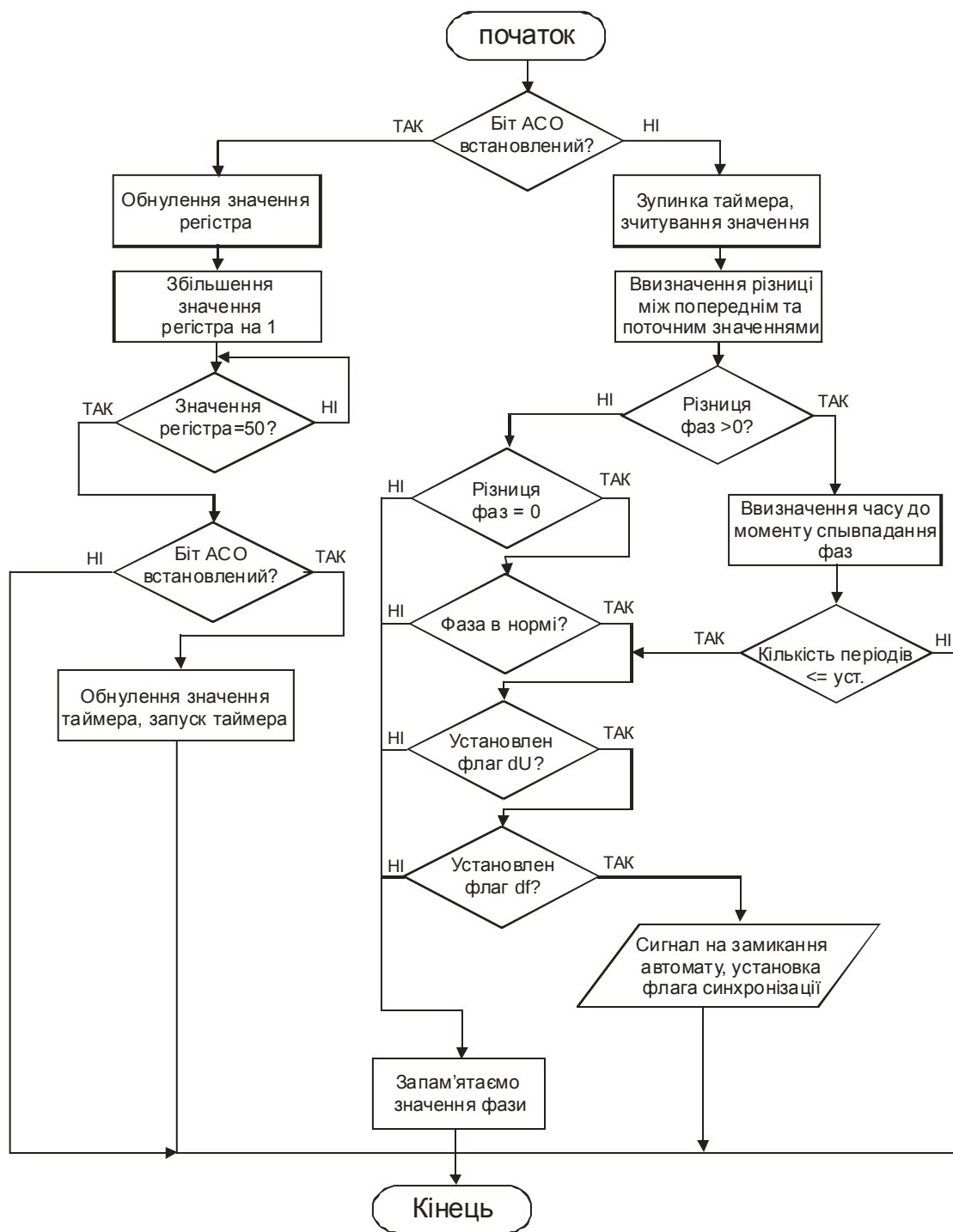


Рисунок 3.5 – Блок-схема алгоритму обробника переривання

Таким чином, програмна реалізація алгоритму забезпечує інтегровану взаємодію всіх підсистем: від збору та обробки даних до формування керуючих впливів, гарантуючи надійність синхронізації суднових генераторів у складі єдиної енергетичної мережі. Текст програми для мікроконтролера, що реалізує описані алгоритми, наведено в Додатку Б.

3.4. Розробка програмного інтерфейсу для взаємодії з блоком синхронізації

Для реалізації відображення осцилограм напруг підключеного та працюючого генераторів, а також для налаштування параметрів вбудованої мікропроцесорної системи керування синхронізацією, було використано персональний комп'ютер IBM PC. Розробка візуального інтерфейсу виконана мовою C++ у середовищі Visual Studio, що дало можливість забезпечити гнучкість та масштабованість програмного забезпечення.

Архітектура програми базується на трьох ключових класах:

CSynchronApp – клас прикладного екземпляра програми;

CSynchronDlg – клас, що відповідає за діалогове вікно з елементами керування та відображення даних;

CComThread – клас окремого потоку для обробки прийому даних.

Використання окремого потоку дозволило уникнути ситуацій, коли відсутність вхідних даних (через відключений блок або несправність лінії зв'язку) сприймалася б як зависання програми.

Головне вікно інтерфейсу містить елементи для встановлення часу спрацювання автоматичного вимикача та допустимого значення різниці частот. Після задання параметрів формується байт даних, який передається у мікропроцесорний блок. При цьому старша тетрада відповідає різниці частот, а молодша – часу випередження. Передбачено також можливість зчитування реальних параметрів із регуляторів на панелі пристрою, після чого положення елементів керування у програмі автоматично змінюється. Команда на початок синхронізації надсилається у вигляді коду 0xFF.

Відображення інформації реалізоване як у текстовій, так і в графічній формі. Оператор отримує дані про частоти обох генераторів, фазовий зсув і рівень напруги. Індикація стану здійснюється за допомогою кольорових світлодіодів: зелений сигналізує про готовність до синхронізації, червоний – про необхідність корекції параметрів. Додатково у правій частині вікна

					171.6321м.КР.ПЗ.РЗ	Арк.
						52
Змн.	Арк.	№ документа	Підпис	Дата		

виводяться осцилограми сигналів, що відображають роботу вихідних каналів (керування дизелем та збудженням генератора). Стан автоматичного вимикача індикуюється спеціальним маркером. Вигляд інтерфейсу програми наведено на рис. 3.6.

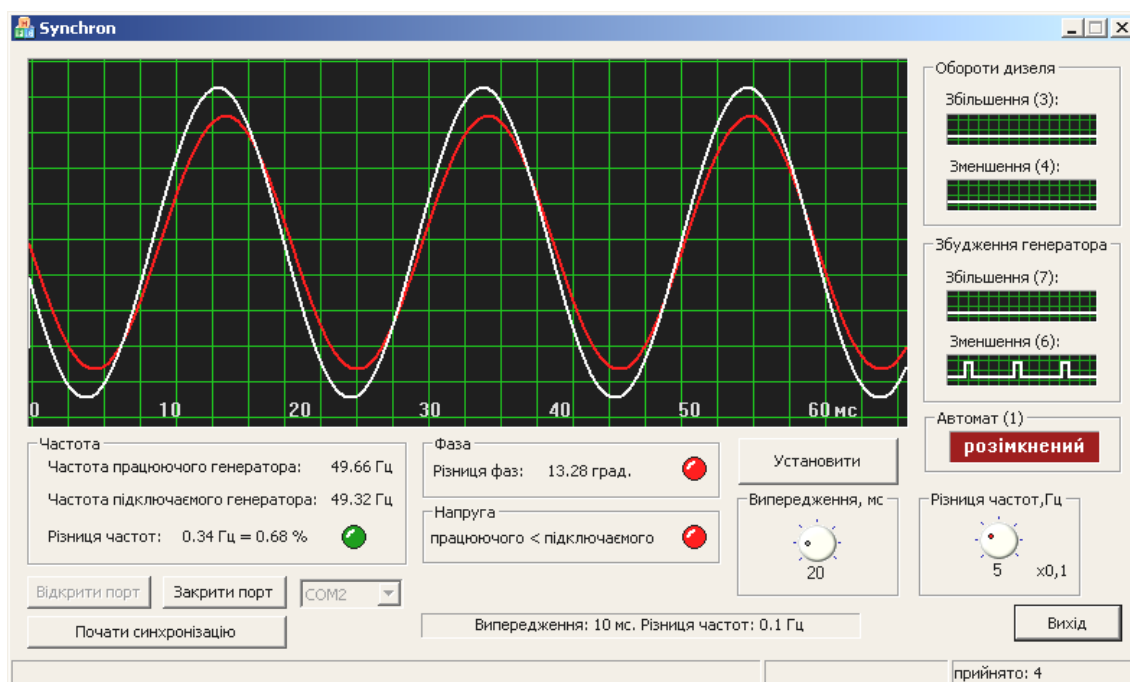


Рисунок 3.6 – Вигляд інтерфейсу програми

Передача даних із класу потоку до інтерфейсу здійснюється через функцію OnRecieveData(), де виконується попередня обробка інформації та її підготовка до візуалізації. Для мінімізації мерехтіння екрану застосовується метод подвійної буферизації: малювання відбувається у віртуальному контексті пам'яті, після чого зображення одним кроком копіюється у вікно. Це забезпечує швидке й стабільне оновлення даних.

По завершенні синхронізації користувач має закрити порт і, за потреби, завершити роботу програми. Крім того, передбачена можливість збереження у форматі CSV журналу зміни параметрів (частоти, фази), що дозволяє у середовищі MS Excel будувати графіки та аналізувати ефективність роботи системи, а також коригувати параметри алгоритму синхронізації.

Висновки до розділу

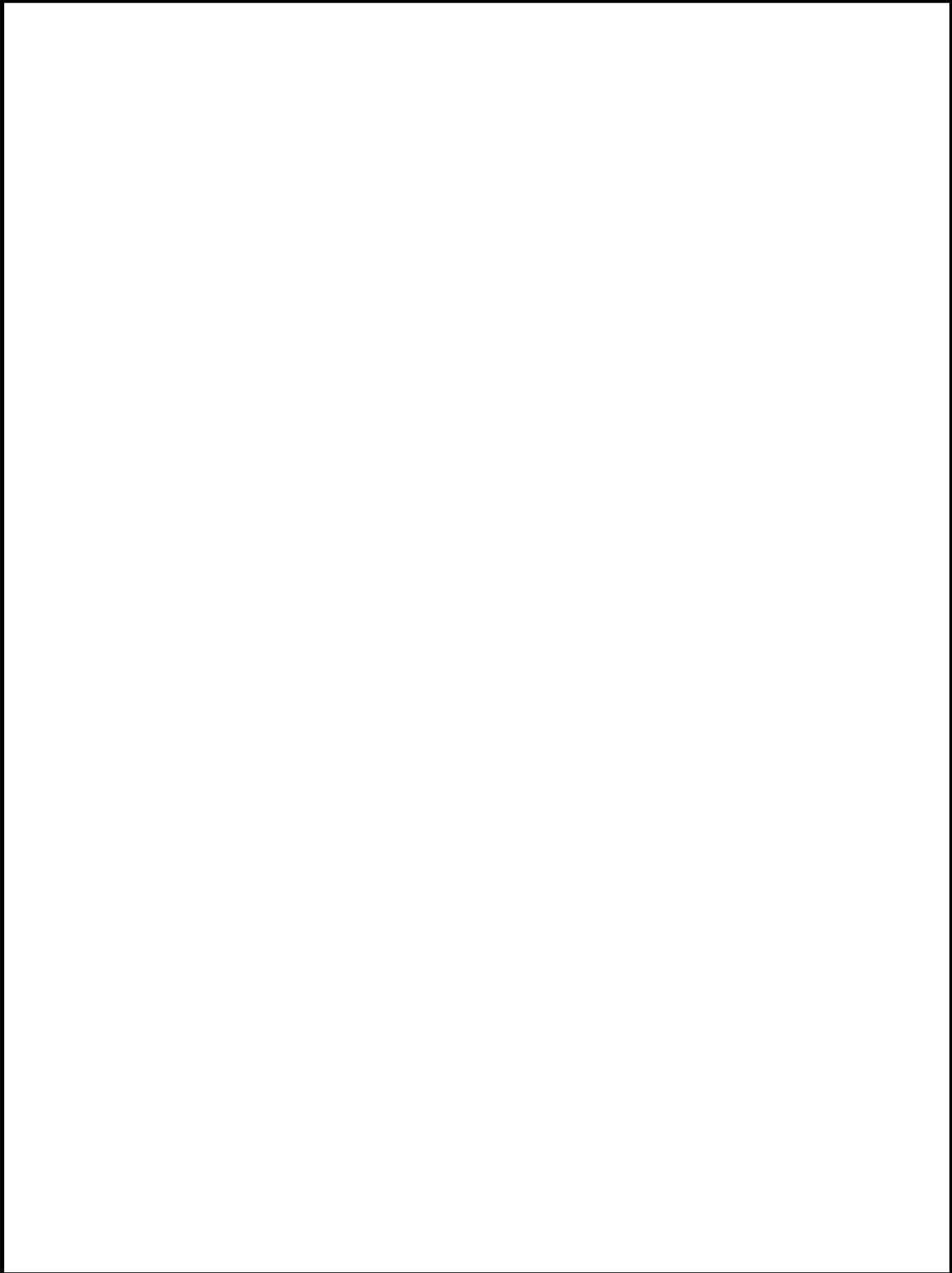
У результаті виконаної роботи було розроблено програмне забезпечення для вбудованої системи керування процесом підключення генератора до загального навантаження, що поєднує мікроконтролерну частину та програмний інтерфейс для персонального комп'ютера. Запропонований алгоритм побудовано за принципом функціональної декомпозиції, що забезпечує поділ завдань на окремі логічні блоки та їх асинхронне виконання. Використання апаратних переривань дало можливість реалізувати своєчасну реакцію на зміну параметрів та підвищити точність вимірювань частоти й фази.

Програмне забезпечення мікроконтролера виконує всі функції контролю й регулювання процесу синхронізації: вимірює параметри генераторів, визначає їх різницю, формує сигнали для регулювання обертів дизеля та збудження, а також подає команду на замикання автоматичного вимикача за умови виконання критеріїв синхронізації.

Окремо було створено програмний інтерфейс на базі мови C++ у середовищі Visual Studio, що забезпечує візуалізацію параметрів у вигляді числових показників і графічних осцилограм.

Таким чином, розроблена система поєднує високу точність вимірювань, надійність виконання функцій та зручність користування. Її можна розглядати як практичну основу для подальшої модернізації та застосування у сучасних багатофункціональних системах керування електроенергетичними установками.

					171.6321м.КР.ПЗ.РЗ	Арк.
						54
Змн.	Арк.	№ документу	Підпис	Дата		



					171.6321м.КР.ПЗ.Р4			
Змн.	Арк.	№ документу	Підпис	Дата				
Розробник		Івановський С.А.			Охорона праці	Літ.	Арк.	Аркуші
Консультант		Губальцев А. М.					55	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Ушкаренко О.О.						
Зав. каф.		Рябенський В. М.						

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1. Вступ

Охорона праці – це система правових, економічних, організаційних, технічних, санітарно-гігієнічних і медико-профілактичних заходів, спрямованих на збереження життя, зміцнення здоров'я та підтримку працездатності працівників у процесі виконання трудових функцій.

Роботодавцем вважається власник підприємства чи його представник, незалежно від форми власності та виду діяльності, а також фізична особа, яка наймає працівників.

Працівник – це особа, яка виконує обов'язки на підставі трудового договору чи контракту в межах організації чи установи.

Сфера дії законодавства

Закон України про охорону праці поширюється на всі види господарської діяльності, включаючи юридичних і фізичних осіб-підприємців, які використовують найману працю, та охоплює всіх працівників без винятку.

Правове забезпечення

Регулювання питань охорони праці базується на таких актах:

- Закон України «Про охорону праці»;
- Кодекс законів про працю України;
- Закон України «Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання»;
- Інші підзаконні нормативно-правові документи.

Якщо міжнародні угоди, ратифіковані Україною, передбачають інші норми, застосовуються положення міжнародного права.

Державна політика у сфері охорони праці

					171.6321м.КР.ПЗ.Р4	Арк.
Змн.	Арк.	№ документу	Підпис	Дата		56

Основою державної стратегії є Конституція України, а головними завданнями – створення безпечних умов праці, профілактика травматизму та попередження професійних захворювань.

Основні принципи державної політики:

- пріоритет охорони життя та здоров'я працівника;
- повна відповідальність роботодавця за безпеку на виробництві;
- постійний контроль за технічним станом робочих місць;
- впровадження національних і регіональних програм із безпеки праці;
- соціальний захист осіб, які постраждали на роботі;
- єдині стандарти охорони праці для всіх роботодавців;
- врахування фізичних та психологічних можливостей працівників при організації роботи;
- економічне стимулювання роботодавців до дотримання правил безпеки та фінансування профілактичних заходів з боку держави;
- навчання та підвищення кваліфікації працівників з питань охорони праці;
- співпраця між державними органами, роботодавцями та профспілками;
- використання міжнародного досвіду та партнерство з глобальними організаціями у сфері безпеки праці.

4.2. Аналіз небезпечних і шкідливих факторів, впливаючих на людину, при розробці програмного забезпечення

Небезпечні виробничі фактори

До небезпечних виробничих факторів належать умови та елементи робочого середовища, які в певних ситуаціях можуть спричинити травму чи різке погіршення стану здоров'я працівника. Їхнім наслідком найчастіше стають виробничі нещасні випадки.

					171.6321м.КР.ПЗ.Р4	Арк.
						57
Змн.	Арк.	№ документа	Підпис	Дата		

Одним із найбільш критичних ризиків під час створення технічних систем є ураження електричним струмом. Потрапляючи в організм людини, струм може призвести до тяжких пошкоджень, а в окремих випадках – до смерті. Характер дії визначається силою й видом струму (змінний чи постійний), його частотою, напругою, часом впливу та електричним опором тіла.

Види уражень електричним струмом

Місцеві електротравми: опіки, характерні мітки на шкірі, металізація поверхні шкіри від електричної дуги, механічні пошкодження, електроофтальмія.

Електричні удари (ступені тяжкості):

I – судоми м'язів без втрати свідомості;

II – втрата свідомості при судомах;

III – непритомність із порушенням роботи серця;

IV – клінічна смерть.

Постійний струм є менш небезпечним за змінний (50 Гц) у 4–5 разів, але це правило діє лише до напруги 500 В. Вище цього рівня він становить ще більшу загрозу.

Небезпечні діапазони сили змінного струму (50 Гц):

0,6–1,5 мА – межа відчуття;

10–15 мА – «невідпускаючий» струм (судоми рук, голосових зв'язок);

25–50 мА – можливе припинення дихання;

50–100 мА – фібриляція серця й загроза його зупинки.

Основні причини електротравм:

- випадкове доторкання до частин під напругою;
- пошкодження ізоляції, що призводить до появи напруги на металевих елементах;
- подача живлення на обладнання, яке мало бути знеструмленим;
- крокова напруга при замиканні на землю.

					171.6321м.КР.ПЗ.Р4	Арк.
						58
Змн.	Арк.	№ документа	Підпис	Дата		

Близько 60% випадків пов'язані з прямим контактом зі струмопровідними елементами, ще понад 25% – з дотиком до металевих частин, які в нормі не повинні бути під напругою. Головний чинник – порушення правил експлуатації електроустановок.

Заходи підвищення електробезпеки:

- обмеження доступу до небезпечних елементів;
- використання якісної ізоляції та контроль її стану;
- застосування низьковольтних схем, де це можливо;
- монтаж заземлення;
- захист від статичної електрики;
- захист низьковольтних кіл від високовольтних перенапруг;
- суворе дотримання інструкцій і регламентів;
- регулярне технічне обслуговування.

Шкідливі виробничі фактори

Шкідливий фактор – це умова праці, яка поступово може призвести до розвитку хронічних хвороб або зниження функціональних можливостей організму. Його дія накопичувальна та довготривала.

При проєктуванні програмного забезпечення для вбудованих систем керування значна частина роботи здійснюється за комп'ютером. Це створює низку ризиків: вплив електромагнітних і електростатичних полів, світлових відблисків і випромінювання.

Вплив електромагнітних полів (ЕМП):

- міліметрові хвилі поглинаються шкірою;
- сантиметрові впливають на поверхневі тканини;
- дециметрові проникають до 10 см у глибину організму.

Теплова дія ЕМП: підвищення температури в тканинах із недостатнім кровопостачанням (очний кришталік, печінка, шлунок), що може провокувати хронічні захворювання.

Біологічна дія:

					171.6321м.КР.ПЗ.Р4	Арк.
						59
Змн.	Арк.	№ документа	Підпис	Дата		

- порушення роботи нервової та ендокринної систем;
- зміни електричної активності мозку;
- розвиток астенії, уповільнення серцебиття;
- головний біль, зниження працездатності.

Електростатичні поля

Утворюються під час роботи дисплея, спричиняють накопичення пилу на шкірі, що може викликати подразнення та збільшувати ризик проблем із зором (зокрема катаракти).

Світлові фактори

Відблиски й надмірна яскравість екрана провокують зорову втому, погіршення гостроти зору, розвиток короткозорості, що негативно позначається на ефективності праці.

4.3. Розрахунок захисного заземлення

Як уже зазначалося, одним із головних ризиків під час роботи з обладнанням при налагодженні розробленого програмного забезпечення є можливість ураження електричним струмом. З метою запобігання цьому необхідно виконати розрахунок параметрів захисного заземлення (ЗЗ) для даного обладнання.

Вихідні дані для розрахунку:

1. Електроустановка з напругою до 1000 В;
2. Нормативне допустиме значення опору ЗЗ: 4 Ом ;
3. Використовуваний заземлювач – кутник, заглиблений у ґрунт (довжина $l = 2,5$ м, ширина $b = 43$ мм);
4. Тип ґрунту – садова земля;
5. Глибина прокладання заземлювального пристрою від поверхні: 0,7 м;
6. З'єднувальний елемент – сталева смуга круглого перерізу, прокладена в землі (діаметр 25 мм).

					171.6321м.КР.ПЗ.Р4	Арк.
						60
Змн.	Арк.	№ документа	Підпис	Дата		

Відомо, що нормативне значення опору захисного заземлення становить 4 Ом.

Питомий опір садової землі визначається за табличними даними, і складає 50 Ом/м. Далі виконуємо розрахунок опору розтікання струму для кутникового заземлювача, заглибленого в ґрунт.

$$R_1 = \frac{\rho}{2 \cdot \pi \cdot l} \cdot \left(\ln \frac{2.1 \cdot l}{b} + \frac{1}{2} \cdot \ln \frac{4.2 \cdot t + l}{4.2 \cdot t - l} \right), \text{ де}$$

t – глибина закладення металевого заземлювача, м;

l – довжина металевого кутка, м;

b – ширина металевого кутка, м;

t_0 – глибина закладення металевого кутка від рівня поверхні землі, м.

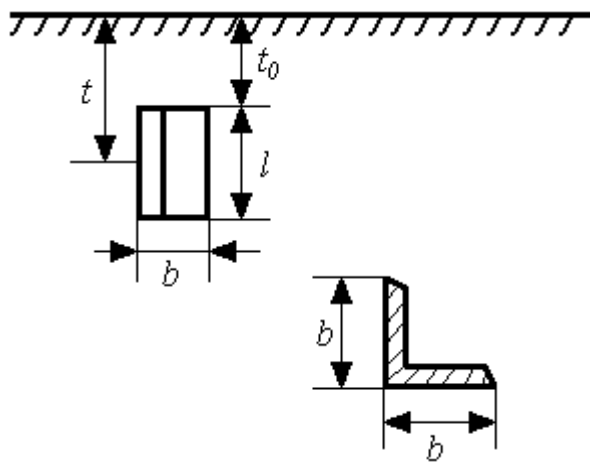


Рисунок 4.1 – Креслення електроду даного типу

$$t = t_0 + \frac{1}{2} \cdot l = 0.7 + \frac{1}{2} \cdot 2.5 = 1.95 \text{ (м)}.$$

$$\text{Тоді: } R_1 = \frac{50}{2 \cdot 3.1416 \cdot 2.5} \cdot \left(\ln \frac{2.1 \cdot 2.5}{0.043} + \frac{1}{2} \cdot \ln \frac{4.2 \cdot 1.95 + 2.5}{4.2 \cdot 1.95 - 2.5} \right) = 16.298 \text{ (Ом)}.$$

Оскільки опір одиночного заземлювача має розмір більше допустимого ($R_1 > R_d$), тоді використовуємо штучні заземлювачі, що з'єднуються паралельно.

Необхідна кількість паралельно з'єднаних заземлювачів:

$$n' = \frac{R_1}{\eta_b \cdot R_d} = \frac{16.298}{1 \cdot 4} = 4.074,$$

					171.6321м.КР.ПЗ.Р4	Арк.
						61
Змн.	Арк.	№ документа	Підпис	Дата		

де η_v – коефіцієнт використання вертикальних заземлювачів, що враховує їх взаємне екранування ($\eta_v = 1$).

Отримане число заземлювачів округляємо до цілого числа n , тоді: $n \approx 4$

З таблиці (методичні вказівки – С. 17) знаходиться фактичне значення коефіцієнту використання заземлювачів η_ϕ та визначається фактичний опір вертикальних заземлювачів $R_{v,\phi}$.

$$\eta_\phi = 0.76, \text{ отже, } R_{v,\phi} = \frac{R_1}{n \cdot \eta_\phi} = \frac{16.298}{4 \cdot 0.76} = 5.361 \text{ (Ом)}.$$

Для вертикальних з'єднувачів застосовується з'єднувальна смуга круглого перетину в ґрунті діаметром 25 мм (з похідних даних).

Визначаємо значення опору розтіканню струму для смуги. Розрахунок проводиться аналогічно розрахунку опору при розтіканні струму одиночного заземлювача.

$$R_{c,\Pi} = \frac{\rho}{2 \cdot \pi \cdot L_{c,\Pi}} \cdot \ln \frac{L_{c,\Pi}^2}{D \cdot T_{c,\Pi}}, \text{ де}$$

$L_{c,\Pi}$ – довжина металевої смуги, що з'єднує заземлювачі, м;

$L_{c,\Pi} = 1.05 \cdot a \cdot n$ (де a – відстань між заземлювачами);

$T_{c,\Pi}$ – глибина закладання з'єднувальної смуги в землю, м;

D – діаметр смуги (труби), м;

n – кількість заземлювачів.

$a = 1 \cdot 1 = 1 \cdot 2.5 = 2.5 \text{ (м)};$

$$T_{c,\Pi} = t_0 + \frac{1}{2} \cdot D = 0.7 + \frac{1}{2} \cdot 0.025 = 0.712 \text{ (м)}.$$

					171.6321м.КР.ПЗ.Р4	Арк.
						62
Змн.	Арк.	№ документа	Підпис	Дата		

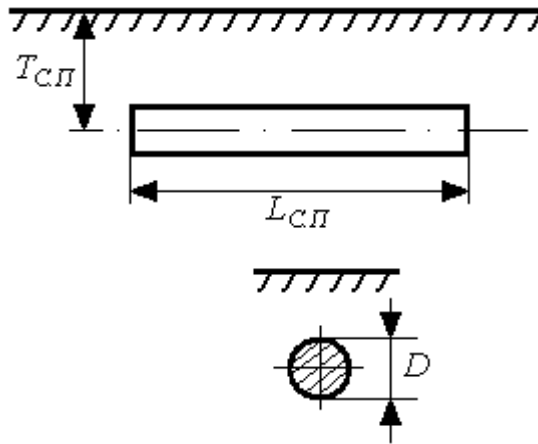


Рисунок 4.2 – Креслення з'єднувального електроду даного типу

$$\text{Отже, } R_{\text{СП}} = \frac{50}{2 \cdot 3.1416 \cdot 10.5} \cdot \ln \frac{10.5^2}{0.025 \cdot 0.712} = 6.617 \text{ (Ом)}.$$

З урахуванням значення коефіцієнту використання металевої смуги (горизонтального електроду) η_r , опір цієї смуги дорівнює:

$$R_{\Pi} = \frac{R_{\text{СП}}}{\eta_r} = \frac{6.617}{0.77} = 8.593 \text{ (Ом)}, \text{ де } \eta_r = 0.77.$$

Фактичний опір заземлювача складається зі значень опорів вертикальних заземлювачів та опору з'єднувальної смуги, і визначається як еквівалентний опір при паралельному з'єднанні цих провідників:

$$R_3 = \frac{R_{\text{в.ф}} \cdot R_{\text{СП}}}{R_{\text{в.ф}} + R_{\text{СП}}} = \frac{5.361 \cdot 8.593}{5.361 + 8.593} = 3.301 \text{ (Ом)}.$$

Отримане значення опору заземлювача задовольняє умові $R_3 \leq R_d$.

Отже, можна зробити висновок, що отримані параметри заземлювача відповідають усім умовам по електробезпеці під час роботи з обладнанням для налагодження програмного забезпечення.

Результати розрахунку:

1. Кількість вертикальних заземлювачів: 4 шт.;
2. Опір розтіканню струму одиночного заземлювача: 16,298 Ом;
3. Довжина з'єднувальної смуги: 10,5 м;
4. Опір розтіканню струму з'єднувальної смуги: 6,617 Ом;
5. Опір з'єднувальної смуги: 8,593 Ом;

					171.6321м.КР.ПЗ.Р4	Арк.
Змн.	Арк.	№ документа	Підпис	Дата		63

6. Фактичний опір ЗП: 3,301 Ом.

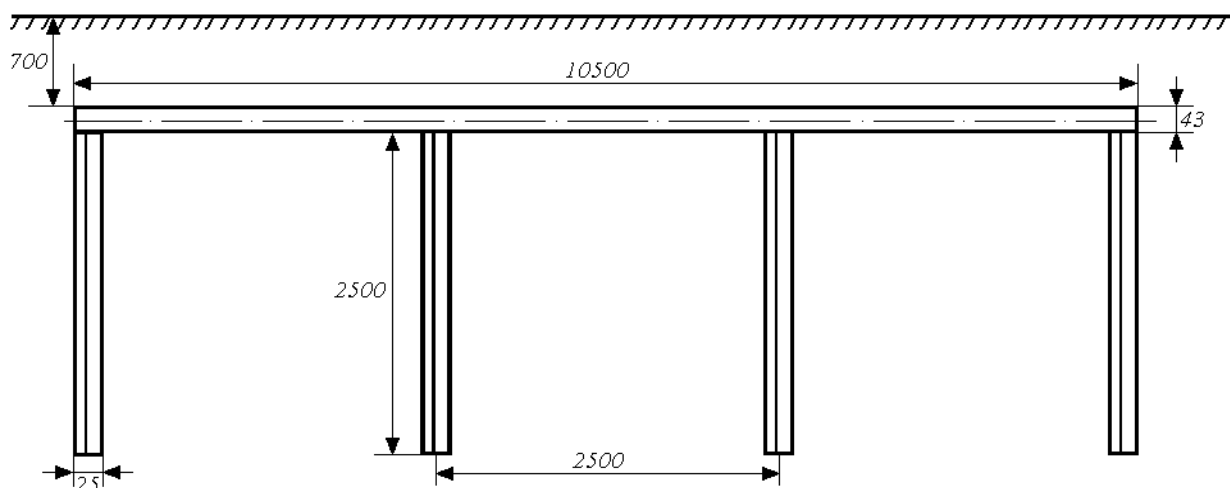


Рисунок 4.3 – Схема з'єднувальної смуги заземлювача

Щоб забезпечити комфорт та безпеку роботи за комп'ютером, необхідно правильно організувати освітлення, мікроклімат і загальне облаштування приміщення.

Освітлення

Оптимальним є комбіноване освітлення, яке поєднує природні та штучні джерела. Вікна доцільно орієнтувати на північ, щоб уникати прямих сонячних променів. Денне світло має надходити збоку та відповідати будівельним нормам. Для зменшення різниці яскравості між монітором і навколишнім середовищем варто застосовувати жалюзі, світловідбивні плівки або фіранки зі середнім коефіцієнтом відбиття.

Дисплеї потрібно розташовувати так, щоб уникати появи відблисків від ламп і вікон. Для штучного освітлення найкраще підходять люмінесцентні світильники, розміщені рівномірно над робочими зонами. Яскравість екрана й оточення повинні співвідноситися не більше ніж 3:1, а пульсація світла має залишатися в межах допустимих норм.

Організація робочого простору

					171.6321м.КР.ПЗ.Р4	Арк.
Змн.	Арк.	№ документу	Підпис	Дата		64

Приміщення слід обладнати системами затемнення, які дають змогу за потреби повністю перекривати денне світло. Одне робоче місце має передбачати не менше 6 м² площі та 20 м³ повітряного об'єму. Робочі місця бажано розділяти перегородками чи ізолювати, забезпечуючи достатню вентиляцію й доступ природного освітлення.

Для оздоблення інтер'єру рекомендовано використовувати матові матеріали, що рівномірно розсіюють світло. Підлога повинна бути стійкою до ковзання, легко очищуваною й такою, що не накопичує статичну електрику. Використання токсичних матеріалів категорично заборонено, а концентрація шкідливих домішок у повітрі не має перевищувати встановлені санітарні норми.

Мікроклімат

Сприятливі параметри середовища є ключовим чинником ефективної роботи персоналу та надійності техніки. Робочий діапазон температур - від 10 до 40, відносна вологість - 40–90 %. Оптимальні значення: температура 19–23 °С, вологість близько 55 %, швидкість руху повітря мінімальна. У спекотний період допускається 23–25 °С, у холодний - 22–24 °С. Для регулювання мікроклімату доцільно встановлювати кондиціонери.

З огляду на зміну іонного складу повітря під час роботи моніторів, варто використовувати вентиляційні системи, іонізатори або захисні екрани. Скупчення людей і паління погіршують якість повітря, тому необхідне регулярне провітрювання, контроль доступу та зонування приміщень.

Пожежна безпека

Наявність електронного обладнання підвищує ризик займання через перегрів, короткі замикання або легкозаймисті речовини. Тому кабінети повинні бути обладнані засобами пожежогасіння: вогнегасниками (не менше двох на кімнату площею до 20 м²), ящиками з піском і азбестовими покривалами.

					171.6321м.КР.ПЗ.Р4	Арк.
						65
Змн.	Арк.	№ документа	Підпис	Дата		

ВИСНОВКИ

У результаті виконання магістерської роботи було проведено дослідження, присвячене проблемам та особливостям створення програмного забезпечення для суднових вбудованих систем керування, збору й обробки даних. Аналіз наукових джерел та практичних розробок показав, що останні роки характеризуються активним впровадженням цифрових технологій на морському транспорті. Встановлено, що використання цифрових алгоритмів для керування збудженням генератора забезпечує стабільність вихідної напруги, високоточну підтримку режимів роботи та зменшення впливу зовнішніх збурень на електричну систему судна.

Розроблено та реалізовано програмне забезпечення для ПІД-регулятора, яке дозволяє обчислювати пропорційну, інтегральну та диференціальну складові керуючого сигналу в реальному часі з урахуванням обмежень вихідної дії та захисту інтегратора від перенасичення. Програмний модуль забезпечує швидкодію та точність регулювання завдяки використанню апаратної підтримки операцій множення у мікроконтролері.

Було розроблено автоматизоване робоче місце оператора, яке дозволяє віддалено контролювати роботу генератора та системи збудження, налаштовувати коефіцієнти ПІД-регулятора, спостерігати за процесами синхронізації та оперативно реагувати на відхилення режимів. Це значно підвищує ефективність експлуатації та мінімізує вплив людського фактора.

Створено алгоритм введення генератора на паралельну роботу з іншими генераторами на загальне навантаження у складі мікропроцесорної системи керування, збору та обробки даних.

Результати дослідження показали, що інтеграція апаратної та програмної частини системи керування судновим синхронним генератором дозволяє досягти високої точності регулювання напруги, стабільності режимів роботи генераторів та ефективної синхронізації з іншими енергоблоками. Розроблене програмне забезпечення може бути використане

					171.6321м.КР.ПЗ.Висновки	Арк.
						66
Змн.	Арк.	№ документу	Підпис	Дата		

для модернізації існуючих систем керування, інтеграції їх у автоматизовані суднові енергетичні системи та підвищення надійності та безпеки суднового електропостачання.

Результати роботи апробовано на IV Міжнародній науково-практичній конференції Таврійського національного університету імені В. І. Вернадського до 107-ї річниці з дня заснування ТНУ, 16 жовтня 2025 р.

Опубліковано тези доповіді: Ushkarenko O.O., Ivanovskii S.A. Challenges in improving the control quality of diesel-generator sets in autonomous electric power systems / Матеріали VI Міжнародної науково-практичної конференції Таврійського національного університету імені В. І. Вернадського, 16 жовтня 2025 р. Київ : ТНУ ім. В.І. Вернадського, 2025.

Таким чином, поставлені в магістерській роботі завдання були виконані, а мета досягнута. Подальші дослідження можуть бути спрямовані на вдосконалення алгоритмів обробки великих масивів даних, інтеграцію елементів штучного інтелекту для прогнозування несправностей та розширення функціональних можливостей системи у напрямі комплексної автоматизації суднових процесів генерування та перетворення електроенергії.

					171.6321м.КР.ПЗ.Висновки	Арк.
						67
Змн.	Арк.	№ документу	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. IEC 61162-1:2024. Maritime navigation and radiocommunication equipment and systems – Digital interfaces – Part 1: Single talker and multiple listeners. Geneva: IEC, 2024.
2. IEC 61162-450:2024. Maritime navigation and radiocommunication equipment and systems – Digital interfaces – Part 450: Multiple talkers and multiple listeners – Ethernet interconnection. Geneva: IEC, 2024.
3. IEC 61162-460:2024. Maritime navigation and radiocommunication equipment and systems – Digital interfaces – Part 460: Multiple talkers and multiple listeners – Ethernet interconnection – Safety and Security. Geneva: IEC, 2024.
4. Michala A. L., Vourganas I., Coraddu A. Vibration edge computing in maritime IoT. ACM Transactions on Internet of Things. 2021. Vol. 3, No. 1. P. DOI: 10.1145/3484717
5. Jurdana I., Perković M., Mrzljak V. Shipboard data compression method for sustainable real-time acquisition and processing. Sustainability. 2021. Vol. 13, No. 15. P. 8264. DOI: 10.3390/su13158264.
6. Spravil J., Hemminghaus C., von Rechenberg M., Elmar Padilla, Jan Bauer. Detecting maritime GPS spoofing attacks based on NMEA sentence integrity monitoring. Journal of Marine Science and Engineering. 2023. Vol. 11, No. 5. P. 928. DOI: 10.3390/jmse11050928
7. Han D., An Y., Li Y. Design and application of a real-time monitoring system for marine environment. Marine Georesources & Geotechnology. 2021. Vol. 39, No. 10. P. 1193–1202.
8. Clavijo-Mesa M. V., Patino-Rodriguez C. E., Guevara Carazas F. J. Cybersecurity at Sea: A Literature Review of Cyber-Attack Taxonomies and Best Practices. Information. 2024. Vol. 15, No. 11:710. DOI: 10.3390/info15110710

					171.6321м.КР.ПЗ.Джерела	Арк.
						68
Змн.	Арк.	№ документа	Підпис	Дата		

9. Hopcraft R., Harish A.V., Tam K., Jones K. Raising the standard of maritime VDR cybersecurity. Journal of Marine Science and Engineering. 2023. Vol. 11, No. 2. P. 267. DOI: <https://doi.org/10.3390/jmse11020267>

10. Kayışoğlu G., Güneş B., Bolat P. ECDIS cyber security dynamics analysis based on the fuzzy-FUCOM method. Transactions on Maritime Science. 2024. Vol. 13, No. 1. DOI: <https://doi.org/10.7225/toms.v13.n01.w09>

11. Правила класифікації та побудови морських суден. Частина X «Електрообладнання». Київ: Регістр судноплавства України, 2020. 284 с.

12. Volynets Taras, Mosiichuk Dmytro. Information systems for observation of marine aquatories and adjacent areas using aerospace technologies. Environmental safety and natural resources. 2025. Vol. 53. P. 146-154. DOI: 10.32347/2411-4049.2025.1.146-154

13. Рудніченко М., Гришин С., Шибасєв Д., Петров І., Носов М. Проект інформаційної системи для збору, обробки та аналізу метеоданих. Інформаційні технології та суспільство. 2021. Вип. 2. С. 34-41. DOI: <https://doi.org/10.32689/maup.it.2021.2.4>

14. С.В. Сагін, Ю.В. Заблоцький. Діагностування технічного стану судових енергетичних установок засобів водного транспорту / Водний транспорт: Збірник наукових праць. Випуск 2(38). 2023. С. 175-187. DOI: doi.org/10.33298/2226-8553.2023.2.38.19

15. Маранов О. В. Прогнозування відмов судових навігаційних систем з використанням методів машинного навчання // Системи управління, навігації та зв'язку. 2023. № 3. С. 37-40. DOI: 10.26906/SUNZ.2023.3.037

16. Xu Lihaotian. Application and Challenge of Automation Technology in Ocean Engineering // Applied and Computational Engineering. 2025. Vol. 151. P. 63-68. DOI: 10.54254/2755-2721/2025.22846

17. Dagkinis Ioannis, Psomas Panagiotis, Platis Agapios. Enhanced Ships Operation using Integrated Automation Systems // Interdisciplinary Journal of Research and Development. 2022. Vol. 9:54. DOI: 10.56345/ijrdv9n4s208

					171.6321м.КР.ПЗ.Джерела	Арк.
						69
Змн.	Арк.	№ документу	Підпис	Дата		

18. Рябенський В.М. Особливості використання автоматного програмування при розробці спеціалізованої SCADA-системи для автономних електроенергетичних установок / В.М. Рябенський, А.О. Ушкаренко, О.И. Дорогань // Технічна електродинаміка. Тем. випуск. Харків, 2012. Ч.3. С.155-159.

19. Kongsberg Gruppen. K-Chief 700 Integrated Automation System [Електронний ресурс]. Режим доступу: <https://www.kongsberg.com/maritime/products/engines-engine-room-and-automation-systems/automation-safety-and-control/vessel-automation-k-chief/integrated-marine-automation-system-k-chief-700/>(<https://www.kongsberg.com/maritime/products/engines-engine-room-and-automation-systems/automation-safety-and-control/vessel-automation-k-chief/integrated-marine-automation-system-k-chief-700/>)

20. Wärtsilä. UNIMACS Automation Systems. 2021. [Електронний ресурс]. Режим доступу: <https://www.wartsila.com/insights/article/next-generation-unic-automation-system-enables-wartsila-31-performance>(<https://www.wartsila.com/insights/article/next-generation-unic-automation-system-enables-wartsila-31-performance>)

21. ABB Marine & Ports. Marine automation and control solutions. 2020. [Електронний ресурс]. Режим доступу: <https://new.abb.com/marine>(<https://new.abb.com/marine>)

22. Рябенський В. М. Автоматизація процесів моніторингу параметрів та керування автономними електроенергетичними системами з дизельгенераторними агрегатами : монографія / В. М. Рябенський, О. І. Дорогань, О. О. Ушкаренко. Миколаїв : НУК, 2017. 240 с.

23. Тимошук О.М., Дакі О.А., Бойко О.А., Карадобрій Т. А. Аналітичний огляд адаптивних систем керування судном та шляхи їх побудови // Водний транспорт: Збірник наукових праць. 2020. Випуск 3(31). С. 120-125.

					171.6321м.КР.ПЗ.Джерела	Арк.
						70
Змн.	Арк.	№ документу	Підпис	Дата		

24. A. Morrall, J. Rainbird, T. Katsoulakas, I. Koliouisis, T. Varelas. e-Maritime for Automating Legacy Shipping Practices. Transportation Research Procedia 2016. Vol. 14. P. 143-152. DOI: 10.1016/j.trpro.2016.05.050

25. Рябенський В.М. Проектування цифрових пристроїв на ПЛІС. Навч. посібник / В.М. Рябенський, О.О. Ушкаренко. К.: «Корнійчук», 2004, 253 с.

26. Пядишев В. Г. Питання вдосконалення кібербезпеки морського транспорту: зарубіжний досвід // Морська безпека та оборона. 2023. № 1. С. 78-86.

27. Лагутін Г. І., Мусаїрова Ю. Д., Грицюк Д. С., Кулинич І. І. Шляхи удосконалення засобів визначення технічного стану дизельних двигунів пересувних електростанцій // Збірник наукових праць Харківського національного університету Повітряних Сил. 2021. С. 122-129. DOI: 10.30748/zhups.2021.68.16.

28. Рябенський В.М. Комп'ютерне керування зовнішніми пристроями через стандартні інтерфейси. Навч. посіб. / В.М. Рябенський, В.Є. Ходаков, О.О. Ушкаренко. Херсон: Олді-плюс, 2008. 380 с.

29. Рябенський В.М. Схемотехніка електронних пристроїв та систем. Том 6. Апаратно-програмні засоби відображення інформації. Підручник / В.М. Рябенський, О.О. Ушкаренко. Миколаїв: Вид-во «Іліон», 2013. 464 с.

30. Рябенський В.М. Програмна реалізація методів розрахунку та аналізу електричних кіл : навч. посібник / В.М. Рябенський, О.О. Ушкаренко, М.Я. Островерхов, М.С. Махмуд Аль-суод. Київ : Вид-во Ліра-К, 2021. 462 с.

31. Abdullah Eial Awwad, Alaa Al-Quteimat, Mahmoud Al-Suod, O.O. Ushkarenko, Atia AlHawamleh. Improving the accuracy of the active power load sharing in paralleled generators in the presence of drive motors shaft speed instability. International Journal of Electronics and Telecommunications. 2021. Vol. 67(3). P. 371–377.

					171.6321м.КР.ПЗ.Джерела	Арк.
						71
Змн.	Арк.	№ документу	Підпис	Дата		

32. Рябенський В.М. Схемотехніка електронних пристроїв та систем : підручник. У 6 т. Т. 3. Мікропроцесорна техніка / В.М. Рябенський, О.О. Ушкаренко, В.С. Буряк. Миколаїв : Іліон, 2012. 445 с.

33. Рябенський В.М. Моделювання електронних, електротехнічних та електроенергетичних систем / Рябенський В.М., Ушкаренко О.О., Махмуд М.С. Аль-Суод. Миколаїв : Іліон, 2024. 424 с.

34. Рябенський В.М. MAX+plus II. Основи проектування цифрових пристроїв на ПЛІС / В.М. Рябенський, О.О. Ушкаренко. К.: «Корнійчук», 2004. 253 с.

35. Ryabenkiy V.M., Ushkarenko A.O. Formalnoe opisanie elementov avtomatizirovannogo rabocheho mesta operatora elektroenergeticheskoy sistemyi / V.M. Ryabenkiy, A.O. Ushkarenko // Vestnik HGMA. 2014. Vol.1. P. 43-50.

36. Рябенський В.М. Verilog. Практика проектування цифрових пристроїв на ПЛІС. : навч. посібник / В.М. Рябенський, О.О. Ушкаренко. Миколаїв : Іліон, 2007. 324 с.

37. Рябенський В.М. Програмовані електронні системи керування, збору та обробки інформації : підручник / В.М. Рябенський, О.О. Ушкаренко. Миколаїв : НУК, 2019. 490 с.

					171.6321м.КР.ПЗ.Джерела	Арк.
						72
Змн.	Арк.	№ документу	Підпис	Дата		

ДОДАТОК А

Програмний код для мікроконтролера у складі вбудованої системи керування судновим синхронним генератором, збору та обробки даних

```
#include <iom16v.h>
#include <macros.h>
#include <math.h>

// === Константи ===
#define MAX_POS_ERROR      250
#define MAX_NEG_ERROR      -250
#define MAX_CORRECTION     310
#define MIN_CORRECTION     -61

#define DELAY_MIN_LIMIT    200
#define DELAY_MAX_LIMIT    8270
#define VOLTAGE_MAX        900
#define VOLTAGE_MIN        70
#define LED_BLINK_LIMIT    50

// === Макроси для керування виводами ===
// Імпульси
#define PULSE1_HIGH()      (PORTC |= 0x02)
#define PULSE1_LOW()       (PORTC &= ~0x02)
#define PULSE2_HIGH()      (PORTC |= 0x01)
#define PULSE2_LOW()       (PORTC &= ~0x01)

// Світлодіоди
#define LED_MAIN_ON()      (PORTB &= ~0x04)
#define LED_MAIN_OFF()     (PORTB |= 0x04)
#define LED_GREEN_TOGGLE() (PORTB ^= 0x20)

#define LED_R1_ON()        (PORTB &= ~0x01)
#define LED_R1_OFF()       (PORTB |= 0x01)
#define LED_R2_ON()        (PORTB &= ~0x02)
#define LED_R2_OFF()       (PORTB |= 0x02)
#define LED_R3_ON()        (PORTB &= ~0x40)
#define LED_R3_OFF()       (PORTB |= 0x40)
#define LED_R4_ON()        (PORTB &= ~0x80)
#define LED_R4_OFF()       (PORTB |= 0x80)

#define LED_Y1_ON()        (PORTB &= ~0x08)
#define LED_Y1_OFF()       (PORTB |= 0x08)
#define LED_Y1_TOGGLE()    (PORTB ^= 0x08)

#define LED_Y2_ON()        (PORTB &= ~0x10)
#define LED_Y2_OFF()       (PORTB |= 0x10)
#define LED_Y2_TOGGLE()    (PORTB ^= 0x10)

// Реле
```

					171.6321м.КР.ПЗ.Додатки	Арк.
						73
Змн.	Арк.	№ документа	Підпис	Дата		

```

#define RELAY1_ON()          (PORTD |= 0x01)
#define RELAY1_OFF()        (PORTD &= ~0x01)
#define RELAY2_ON()          (PORTD |= 0x02)
#define RELAY2_OFF()        (PORTD &= ~0x02)

// Кнопка
#define BUTTON_STATE()      (PORTA & 0x20)
#define BUTTON_PRESSED      0x20

// === Глобальні змінні ===
unsigned int pidKp, pidKi, pidKd;
float pidK1, pidK2, pidK3;

unsigned int adcPrev1, adcPrev2;
signed int pidPrev, prevData;
signed int delayTicks = 6500;
signed long pidIntegral;
char saturationFlag = 0;

char freqMeasureState;
signed int correction;
unsigned int periodCounter;
char syncZero;
char pulseCounter;
char ledBlinkCounter;
char alarmDelay;
int voltageAlarm;

// === Прототипи ===
void init_mcu(void);
unsigned int adc_read(unsigned char channel);
void timer1_start(unsigned int delay_us);
void generate_pulse(void);
void delay_microseconds(int us);
signed int pid_calc(signed int error);

// === Обробники переривань ===
#pragma interrupt_handler int0_isr:2
void int0_isr(void) {
    WDR();
    GIFR &= ~0x40;
    timer1_start(delayTicks);

    if (ledBlinkCounter > LED_BLINK_LIMIT) {
        LED_GREEN_TOGGLE();
        ledBlinkCounter = 0;
    } else {
        ledBlinkCounter++;
    }
}

#pragma interrupt_handler int1_isr:3

```

					171.6321м.КР.ПЗ.Додатки	Арк.
						74
Змн.	Арк.	№ документа	Підпис	Дата		

```

void int1_isr(void) {
    char measuredFreq;

    if (periodCounter == 0) {
        if ((PIND & 0x40) == 0) {
            if (correction < MAX_CORRECTION) {
                correction++;
                LED_Y2_OFF();
            } else {
                correction = MAX_CORRECTION;
                LED_Y2_ON();
            }
            LED_R1_ON();
        }
        if ((PIND & 0x20) == 0) {
            if (correction > MIN_CORRECTION) {
                correction--;
                LED_Y2_OFF();
            } else {
                correction = MIN_CORRECTION;
                LED_Y2_ON();
            }
            LED_R2_ON();
        }
    }

    if (periodCounter < 10) periodCounter++;
    else {
        periodCounter = 0;
        LED_R1_OFF();
        LED_R2_OFF();
    }

    if (freqMeasureState == 0) {
        TCNT2 = 0;
        TCCR2 = 0x07;
        freqMeasureState = 1;
    } else if (freqMeasureState == 1) {
        TCCR2 = 0;
        measuredFreq = TCNT2;
        freqMeasureState = 2;

        if (!(measuredFreq > 130 && measuredFreq < 195)) {
            if (++alarmDelay >= 1) {
                alarmDelay = 0;
                if (measuredFreq <= 130) {
                    LED_R4_ON();
                } else if (measuredFreq >= 195) {
                    LED_R3_ON();
                    RELAY2_ON();
                }
            }
        }
    }
}

```

					171.6321м.КР.ПЗ.Додатки	Арк.
						75
Змн.	Арк.	№ документа	Підпис	Дата		

```

        } else {
            alarmDelay = 0;
            RELAY2_OFF();
            LED_R3_OFF();
            LED_R4_OFF();
        }
    }

}

#pragma interrupt_handler timer1_ovf_isr:9
void timer1_ovf_isr(void) {
    TCCR1B = 0;
    TCNT1H = 0x00;
    TCNT1L = 0x00;
    WDR();
    generate_pulse();
}

// === PID регулятор ===
signed int pid_calc(signed int error) {
    signed int proportional = pidK1 * error;
    signed int derivative   = pidK3 * (error - prevData);
    signed long integralTmp = 0;

    pidKp = adc_read(0);
    pidKi = adc_read(1);
    pidKd = adc_read(2);

    pidK1 = (float)pidKp * 1 / 1023;
    pidK2 = (float)pidKi * 0.9 / 1023;
    pidK3 = (float)(pidKd * 96) / 1023;
    if (!saturationFlag) {
        pidIntegral += error;
    }
    integralTmp = (signed int)pidK2 * pidIntegral / 500;
    signed int result = proportional + derivative +
integralTmp;

    if (result > 3300) {
        saturationFlag = 1;
        result = 3300;
    } else if (result < -3300) {
        saturationFlag = 1;
        result = -3300;
    } else {
        saturationFlag = 0;
    }

    prevData = error;
    return result;
}

```

					171.6321м.КР.ПЗ.Додатки	Арк.
						76
Змн.	Арк.	№ документи	Підпис	Дата		

ДОДАТОК Б

Програмний код системи синхронізації генераторів

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdint.h>
#include <stdbool.h>

/* --- Регістри (глобальні змінні) --- */
/* Відповідають r9..r25 та іншим */
volatile uint8_t freq1 = 0;          // r9
volatile uint8_t freq2 = 0;          // r10
volatile uint8_t freq1H = 0;         // r11
volatile uint8_t freq2H = 0;         // r12

volatile uint8_t timer_presc = 0;    // r15/r16
volatile uint8_t timer1_presc = 0;   // r14
volatile uint8_t empty_reg = 0;      // r13
volatile uint8_t status = 0;         // r20
volatile uint8_t res_freq = 0;       // r19

volatile uint8_t r3 = 0;              // поріг (r3)
volatile uint8_t r4 = 0;              // деяке значення (r4)
volatile uint8_t r7 = 0;              // лічильник передач
volatile uint8_t ZH = 0, ZL = 0;     // ZH:ZL (for transmit)
volatile uint8_t tmp_counter = 0;    // r21
volatile uint8_t temp1 = 0, temp2 = 0; // для затримок

/* 16-bit division operands/results */
volatile uint16_t ddl6u = 0;
volatile uint16_t dres16u = 0;
volatile uint16_t dv16u = 0;

/* Статусні біти */
enum {
    FR1_READY    = (1<<0), // fr1_ready bit index 0
    FR2_READY    = (1<<1), // ...
    INCREASE     = (1<<2),
    FREQ_EQUAL   = (1<<3),
    VOZBUJ_NORMA = (1<<4),
    INC_VOZB     = (1<<5),
    SYNCHRON_BIT = (1<<7)
};

/* Прототипи підпрограм */
void initial_procedure(void);
void send_avt_zamk(void);
void pause_start(void);
void pause_stop(void);
void connect_work(void);
```

					171.6321м.КР.ПЗ.Додатки	Арк.
						77
Змн.	Арк.	№ документа	Підпис	Дата		

```

void work_connect(void);
void check_vozbujdenie(void);
void vozb_inc(void);
void vozb_dec(void);
void div16u(void);
void start_timer1(void);
void stop_timer1(void);

/* Допоміжні затримки */
static inline void delay_2ms(void) { _delay_ms(2.0); }
static inline void delay_35ms(void) { _delay_ms(35.0); }

/* --- Імплементация підпрограм --- */

void div16u(void) {
    // просте ділення 16-bit на 8-bit (повертає 16-bit
результат у dres16u)
    if (dv16u == 0) {
        dres16u = 0xFFFF; // умовно - ділення на 0
    } else {
        dres16u = dd16u / dv16u;
    }
}

/* Ініціалізація */
void initial_procedure(void) {
    cli();
    // АЦ компаратор: налаштування ACSR
    // (в оригіналі:
(0<<ACIS1)+(0<<ACIS0)+(0<<ACD)+(1<<ACIE))
    ACSR = (1<<ACIE);

    timer_presc = 0b00000101; // як в ASM
    timer1_presc = 0b00000011;

    // зовнішні переривання по фронту
    MCUCR = (1<<ISC11)|(1<<ISC10)|(1<<ISC01)|(1<<ISC00);

    // дозвіл INT1, INT0
    GICR = (1<<INT1)|(1<<INT0);

    r7 = 0;
    sei();
}

/* Відправка 0xFF тричі (send_avt_zamk) */
void send_avt_zamk(void) {
    cli();
    delay_35ms();
    // дозвіл передачі, відправка байтів
    UCSRB = (1<<TXEN);
    UDR = 0xFF;
}

```

					171.6321м.КР.ПЗ.Додатки	Арк.
						78
Змн.	Арк.	№ документа	Підпис	Дата		

```

        delay_35ms();
        UDR = 0xFF;
        delay_35ms();
        UDR = 0xFF;
        delay_35ms();
        sei();
    }

    /* Перелік інших підпрограм */
    void pause_start(void) {
        // заборона зовн. переривань
        MCUCR = 0;
        TCNT0 = 0;
        TIMSK = (1<<TOIE0); // дозволити переривання
        переповнення таймера0
        // встановити prescaler для таймера0
        TCCR0 = (1<<CS02) | (0<<CS01) | (1<<CS00);
        r7 = 0;
        // дозвіл UART (TXEN + TXCIE + RXEN + RXCIE)
        UCSRB = (1<<TXEN) | (1<<TXCIE) | (1<<RXEN) | (1<<RXCIE);
        UDR = 1;
        ACSR = (1<<ACIE); // включаємо аналоговий компаратор з
        перериванням
        status |= (1<<6); // еквівалент sbr status, 0b01000000
    }

    void pause_stop(void) {
        TCCR0 = 0;
        TCNT0 = 0;
        // дозволити зовн. переривання INT0/INT1
        GICR = (1<<INT1) | (1<<INT0);
        MCUCR = (1<<ISC11) | (1<<ISC10) | (1<<ISC01) | (1<<ISC00);
        // вимкнути компаратор (ACD=1)
        ACSR = (1<<ACD);
        TCCR1B = 0;
    }

    /* connect_work - імпульс на зменшення обертів */
    void connect_work(void) {
        PORTC |= (1<<7);
        PORTC &= ~(1<<5);
        PORTD |= (1<<6);
        delay_2ms();
        PORTD &= ~(1<<6);
        check_vozbujdenie();
        pause_start();
    }

    /* work_connect - імпульс на збільшення обертів */
    void work_connect(void) {
        PORTC |= (1<<5);
        PORTC &= ~(1<<7);
    }

```

```

        PORTD |= (1<<7);
        delay_2ms();
        PORTD &= ~(1<<7);
        check_vozbujdenie();
        pause_start();
    }

    /* возб_inc / возб_dec: індикація порушення */
    void возб_inc(void) {
        PORTC |= (1<<4);
        PORTD |= (1<<5);
        delay_2ms();
        PORTD &= ~(1<<5);
    }

    void возб_dec(void) {
        PORTC |= (1<<2);
        PORTD |= (1<<4);
        delay_2ms();
        PORTD &= ~(1<<4);
    }

    /* check_vozbujdenie - перевірка різниці напруг */
    void check_vozbujdenie(void) {
        sei();
        PORTC &= ~((1<<4)|(1<<2));
        if ( (PINB & (1<<1)) ) {
            status |= (1<<5);
        } else {
            status &= ~(1<<5);
        }
        if ( (PINB & (1<<0)) ) {
            // jump to возб_ok
            status |= (1<<4);
            PORTC |= (1<<3);
            if (PINB & (1<<1)) PORTC |= (1<<4);
            else PORTC &= ~(1<<4);
            if (!(PINB & (1<<1))) PORTC |= (1<<2);
        } else {
            // not ok
            status &= ~(1<<4);
            PORTC &= ~(1<<3);
            if (status & INC_VOZB) возб_inc();
            else возб_dec();
        }
    }

    /* start_timer1 / stop_timer1 - управління таймером 1 */
    void start_timer1(void) {
        TCNT1L = 0;
        TCNT1H = 0;
        TCCR1B = timer1_presc;
    }

```

					171.6321м.КР.ПЗ.Додатки	Арк.
						80
Змн.	Арк.	№ документа	Підпис	Дата		

```

        PORTC |= (1<<1);
    }

void stop_timer1(void) {
    TCCR1B = 0;
    uint8_t xl = TCNT1L;
    uint8_t xh = TCNT1H;
    PORTC &= ~(1<<1);
}

/* --- Переривання (ISR) --- */

/* Відповідає ext_int1 (зовнішнє переривання) */
ISR(INT1_vect) {
    if ( !(status & FR1_READY) ) {
        // якщо біт не встановлений -> start1
        TCNT0 = 0;
        freq1H = 0;
        TIMSK |= (1<<TOIE0); // дозвіл переповнення таймера0
        timer_presc = 0b00000100;
        TCCR0 = timer_presc;
        status |= FR1_READY;
    } else {
        // зупинити таймер, зчитати TCNT0
        TCCR0 = 0;
        freq1 = TCNT0;
        status &= ~FR1_READY;
        if ( !(status & FR2_READY) ) {
            // виклик freq_compare
            // реалізовано нижче простим викликом
        }
    }
}

/* Відповідає ext_int2 (зовнішнє переривання 2) */
ISR(INT0_vect) {
    if ( !(status & FR2_READY) ) {
        TCNT2 = 0;
        freq2H = 0;
        TIMSK |= (1<<TOIE2); // дозволити переповн. таймера2
        timer_presc = 0b00000110;
        TCCR2 = timer_presc;
        status |= FR2_READY;
    } else {
        TCCR2 = 0;
        freq2 = TCNT2;
        status &= ~FR2_READY;
        if ( !(status & FR1_READY) ) {
            // виклик freq_compare аналогічно вище
        }
    }
}

```

```

/* Timer0 overflow - timer0_ovf */
ISR(TIMER0_OVF_vect) {
    if (status & FR1_READY) {
        // measure
        freq1H++;
        return;
    }
    if (status & FR2_READY) {
        freq2H++;
        return;
    }
    if (tmp_counter == 2) {
        tmp_counter = 0;
        pause_stop();
    } else {
        tmp_counter++;
        check_vozbujdenie();
    }
}

/* Timer2 overflow */
ISR(TIMER2_OVF_vect) {
    freq2H++;
}

/* ADC complete */
ISR(ADC_vect) {
    uint8_t adm = ADMUX;
    if (adm == 1) { // muxer1
        // зчитування ADCL/ADCH
        uint16_t raw = ADC; // ADCL+ADCH
        dd16u = raw;
        dv16u = 110;
        div16u();
        r4 = (uint8_t)dres16u;
        ADCSRA = 0; // out ADCSR,empty_reg
        ADCH = 0; ADCL = 0;
        ADMUX = 2;
        ADCSRA
(1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0) | (1<<ADIE) | (1<<ADEN) | (1<<ADSC);
        if (r4 == 0) r4 = 1;
    } else if (adm == 2) { // muxer2
        uint16_t raw = ADC;
        dd16u = raw;
        ADCSRA = 0; // stop
        dv16u = 65;
        div16u();
        r3 = (uint8_t)dres16u;
        status = 0;
        if (r3 == 0) r3 = 1;
    } else {

```

					171.6321м.КР.ПЗ.Додатки	Арк.
						82
Змн.	Арк.	№ документа	Підпис	Дата		

```

        // інші значення – нічого
    }
}

/* Analog comparator change */
ISR(ANALOG_COMP_vect) {
    if (status & (1<<6)) {
        // після паузи
        status &= ~(1<<6);
        return;
    }
    if (ACSR & (1<<ACO)) {
        // ACO == 1
        stop_timer1();
        // невеликий програмний гистерезис
        for (volatile uint8_t i=0;i<50;i++) { _delay_ms(1);
}
        if (!(ACSR & (1<<ACO))) {
            start_timer1();
        }
    } else {
        // ACO == 0
        stop_timer1();
        // обробка
    }
}

/* USART TX complete (matching tx_complete) */
ISR(USART_TXC_vect) {
    uint8_t temp = r7;
    switch (temp) {
        case 0:
            r7++;
            UDR = freq1;
            break;
        case 1:
            r7++;
            UDR = 2;
            break;
        case 2:
            r7++;
            UDR = freq2;
            break;
        case 3:
            r7++;
            UDR = 3;
            break;
        case 4:
            r7++;
            UDR = ZH;
            break;
        case 5:

```

					171.6321м.КР.ПЗ.Додатки	Арк.
						83
Змн.	Арк.	№ документа	Підпис	Дата		

```

        r7++;
        UDR = ZL;
        break;
    case 6:
        r7++;
        {
            uint8_t tmp = freq1H;
            // операції зі зміщеннями/об'єднанням
            tmp = (tmp << 1) | ((freq2H<<2)>>2);
            uint8_t st = status >> 4;
            tmp |= (st & 0x03);
            UDR = tmp;
        }
        break;
    case 7:
        UCSRB = (1<<RXEN) | (1<<RXCIE);
        r7 = 0;
        break;
    default:
        break;
}

/* USART RX complete (rx_complete) */
ISR(USART_RXC_vect) {
    uint8_t temp = UDR;
    if (temp == 0) return;
    if (temp == 0xEF) {
        if ( (status & SYNCHRON_BIT) == 0 ) {
            // start_sync
            WDTCSR = (1<<WDE) | (1<<WDTOE);
            // (з оригіналу: out WDTCSR,0b00001010)
        }
    } else {
        r4 = temp;
        r3 = temp;
        uint8_t t3 = (r3 >> 4) & 0x0F;
        r4 &= 0x0F;
        r3 = t3;
    }
}

/* --- Головна функція --- */
int main(void) {
    /* Зупинка сторожового таймера */
    WDTCSR = (1<<WDTOE) | (1<<WDE);
    WDTCSR = 0;
    // Очистити регістри/змінні
    empty_reg = 0;
    res_freq = 0;
    tmp_counter = 0;
    status = 0;
}

```

```

r3 = 0; r4 = 0;
freq1H = 0; freq2H = 0;

// Ініціалізація стеку відбувається автоматично при
запуску
// Налаштування портів
DDRC = 0xFF; // всі виводи порту C на вихід (еквівалент
ser r16; out ddrc,r16)
PORTC = 0x00;
DDRD = 0x00; // старші 4 ніжки порту D - вхід (у ASM
було set 0x0 -> вихід?), залишив 0
DDRB = (1<<4); // 5 ніжка порту B як вихід (в ASM
0b00010000)
// чекати декілька разів 35ms
for (int i=0;i<6;i++) delay_35ms();

// Налаштування ADC: вибір каналу 1
ADMUX = 1;
ADCSRA
(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0)|(1<<ADIE)|(1<<ADEN)|(1<<ADSC); =
status |= SYNCHRON_BIT; // sbr status,0b10000000
sei();

// Очікуємо завершення перетворення АЦП
while (status & SYNCHRON_BIT) {
    ; // busy wait як у оригіналі
}

// Після синхронізації: налаштування UART
// UBRR = 3 -> UBRR = 3
UBRR = 3;
UBRRH = 0;
// Увімкнути UART Tx/Rx та Rx interrupt
UCSRB = (1<<TXEN)|(1<<RXEN)|(1<<RXCIE);

// Підготовка передачі: аналіз r4/r3 та передача через
UDR
{
    uint8_t tmp = r4;
    tmp = (tmp << 4) | (r3 & 0x0F);
    UDR = tmp;
}
delay_35ms();

initial_procedure();
sei();

/* Головний цикл */
while (1) {
    if (status & SYNCHRON_BIT) {
        // якщо біт синхронізації встановлений - перейти
на stop_synchron

```

					171.6321м.КР.ПЗ.Додатки	Арк.
						85
Змн.	Арк.	№ документа	Підпис	Дата		

```

// у ASM: sbrc status,7 ; rjmp stop_synchron
// Готові операції при замиканні:
// Заборона зовнішніх переривань
MCUCR = 0;
MCUSR = 0;
TCCR0 = 0;
TCCR1A = 0;
TCCR1B = 0;
TCCR2 = 0;
ACSR = 0;
ADCSRA = 0;
GICR = 0;
ADMUX = 0;
send_avt_zamk();
UCSRB = (1<<RXEN) | (1<<RXCIE);
PORTC &= ~( (1<<5) | (1<<7) | (1<<2) | (1<<4) );
sei();
// бескінецьний цикл
while (1) { asm volatile ("nop"); }
}

return 0;
}

```

					171.6321м.КР.ПЗ.Додатки	Арк.
						86
Змн.	Арк.	№ документу	Підпис	Дата		