

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра комп'ютерних систем та мереж

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр

(назва освітнього ступеня)

на тему: **Методи та програмно-апаратні засоби обміну даними між
мережами Modbus TCP і Modbus RTU**

Виконав: студент(ка) 6 курсу, групи СІм-61
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис) Долгушин Я.В.
(прізвище та ініціали)

Керівник _____
(підпис) Тиш Є.В.
(прізвище та ініціали)

Нормоконтроль _____
(підпис) Луцик Н. С.
(прізвище та ініціали)

Завідувач кафедри _____
(підпис) Осухівська Г.М.
(прізвище та ініціали)

Рецензент _____
(підпис) Литвиненко Я.В.
(прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Осухівська Г.М.
(підпис) (прізвище та ініціали)
«17» листопада 2025 р.

магістр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студенту Долгушину Ярославу Валерійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи та програмно-апаратні засоби обміну даними між мережами Modbus TCP і Modbus RTU

Керівник роботи канд. техн. наук, доцент Тиш Євгенія Володимирівна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 14 » листопада 2025 року № 4/7-987

2. Термін подання студентом завершеної роботи 16 грудня 2025 р.

3. Вихідні дані до роботи програмно-апаратний шлюз, Modbus TCP, Modbus RTU, промислові мережі, мікроконтролер, маршрутизація даних, автоматизовані системи керування.

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ

1 Аналіз технологій та рішень для інтеграції мереж Modbus

2 Теоретичні основи побудови програмно-апаратного шлюзу Modbus

3 Практична реалізація шлюзу Modbus

4 Охорона праці та безпека в надзвичайних ситуаціях

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Актуальність і мета дослідження.

2. Завдання, об'єкт і предмет, наукова новизна і практична цінність дослідження.

3. Архітектура програмно-апаратного шлюзу

4. Апаратна платформа шлюзу

5. Загальна блок-схема

6. Схема перетворення Modbus TCP - Modbus RTU

7. Тестування системи

8. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Охорона праці</i>	<i>Осухівська Г.М., зав.каф. КС</i>	<i>04.12.2025</i>	
<i>Безпека в надзвичайних ситуаціях</i>	<i>Стручок В.С., ст. викладач каф. Обладнання харчових технологій</i>	<i>11.12.2025</i>	

7. Дата видачі завдання 17.11.2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Огляд літературних джерел. Постановка завдання на кваліфікаційну роботу.</i>	<i>17.11.2025р. – 23.11.2025р.</i>	<i>Викон.</i>
2.	<i>Теоретичні основи побудови програмно-апаратного шлюзу Modbus</i>	<i>24.11.2025р. – 02.12.2025р.</i>	<i>Викон.</i>
3.	<i>Практична реалізація шлюзу Modbus</i>	<i>03.12.2025р. – 10.12.2025р.</i>	<i>Викон.</i>
4.	<i>Охорона праці та безпека в надзвичайних ситуаціях</i>	<i>04.12.2025р. – 10.12.2025р.</i>	<i>Викон.</i>
5.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>11.12.2025р. – 19.12.2025р.</i>	<i>Викон.</i>
6.	<i>Попередній захист кваліфікаційної роботи магістра</i>	<i>16.12.2025р.</i>	<i>Викон.</i>
7.	<i>Захист кваліфікаційної роботи магістра</i>	<i>22.12.2025р.</i>	<i>Викон.</i>

Студент

(підпис)*Долгушин Я.В.*

(прізвище та ініціали)

Керівник роботи

(підпис)*Тити Є.В.*

(прізвище та ініціали)

АНОТАЦІЯ

Долгушин Я.В. Методи та програмно-апаратні засоби обміну даними між мережами Modbus TCP і Modbus RTU: робота на здобуття кваліфікаційного ступеня магістра: спец. 123 — комп'ютерна інженерія / наук.кер. Тиш Є.В. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2025.

Ключові слова: програмно-апаратний шлюз, Modbus TCP, Modbus RTU, промислові мережі, мікроконтролер, маршрутизація даних, автоматизовані системи керування.

Кваліфікаційна робота присвячена дослідженню методів і засобів інтеграції промислових мереж Modbus TCP та Modbus RTU на основі програмно-апаратного шлюзу. У роботі виконано аналіз сучасних підходів до побудови промислових комунікаційних систем, розглянуто особливості протоколів Modbus RTU і Modbus TCP, а також проблеми їх спільного використання в автоматизованих системах керування технологічними процесами.

У межах дослідження розроблено програмно-апаратний шлюз, який забезпечує двосторонню конвертацію даних між мережами Ethernet та RS-485. Програмна частина системи реалізована з використанням операційної системи реального часу FreeRTOS і мережевого стеку lwIP, що дозволяє організувати багатозадачну обробку запитів, підтримку кількох TCP-з'єднань та гнучку маршрутизацію Modbus-повідомлень.

Отримані результати можуть бути використані для модернізації існуючих автоматизованих систем керування, інтеграції застарілого обладнання в сучасні мережі та побудови масштабованих промислових комунікаційних рішень.

ANNOTATION

Dolhushyn Y.V. Methods and hardware–software tools for data exchange between Modbus TCP and Modbus RTU networks. Master’s Graduation Thesis: speciality 123 — Computer engineering / supervisor Tysh E.V. Ternopil: Ternopil Ivan Puluj National Technical University, 2025.

Keywords: hardware-software gateway, Modbus TCP, Modbus RTU, industrial networks, microcontroller, data routing, automated control systems.

The Master’s graduation thesis is dedicated to the study of methods and tools for integrating Modbus TCP and Modbus RTU industrial networks based on a hardware-software gateway. The work analyzes modern approaches to the design of industrial communication systems, examines the characteristics of Modbus RTU and Modbus TCP protocols, and identifies the challenges of their joint operation within automated process control systems.

As part of the research, a hardware-software gateway was developed, providing bidirectional data conversion between Ethernet and RS-485 networks. The software architecture is implemented using the FreeRTOS real-time operating system and the lwIP network stack, enabling multitasking request processing, support for multiple simultaneous TCP connections, and flexible routing of Modbus messages.

The obtained results can be applied to the modernization of existing automated control systems, integration of legacy equipment into modern industrial networks, and development of scalable industrial communication solutions.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ТЕХНОЛОГІЙ ТА РІШЕНЬ ДЛЯ ІНТЕГРАЦІЇ МЕРЕЖ MODBUS.....	11
1.1. Актуальність теми.....	11
1.2. Мета і завдання дослідження.....	12
1.3. Об'єкт і предмет дослідження.....	13
1.4. Огляд стандартів і технологій.....	15
1.5. Огляд існуючих рішень і наукових праць.....	20
1.6. Вимоги до шлюзу.....	23
1.7. Підсумки аналітичного огляду.....	25
РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ ПРОГРАМНО-АПАРАТНОГО ШЛЮЗУ MODBUS	27
2.1. Загальна модель архітектури програмно-апаратного шлюзу.....	27
2.2. Розробка внутрішньої логіки конвертації.....	30
2.3. Моделі маршрутизації та правила трансляції.....	32
2.4. Алгоритми обробки запитів і черг.....	34
2.5. Механізм конфігурації через АТ-команди по TCP.....	36
2.6. Проблеми сумісності та рішення.....	38
2.7. Підсумки теоретичних досліджень.....	39
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ШЛЮЗУ MODBUS.....	43
3.1. Опис реалізованого пристрою.....	43
3.2. Схема системи та апаратні рішення.....	46
3.3. Архітектура програмного забезпечення.....	49
3.4. Реалізація перетворення Modbus TCP ↔ Modbus RTU.....	53
3.5. АТ-інтерфейс та конфігурація пристрою.....	56
3.6. Налаштування маршрутизації.....	63

	6
3.7. Тестування системи.....	63
3.8. Практична доцільність застосування та рекомендації щодо удосконалення.....	67
3.9. Підсумки практичної реалізації.....	69
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	70
4.1. Охорона праці.....	70
4.2 Безпека в надзвичайних ситуаціях.....	72
ВИСНОВКИ.....	75
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	77
Додаток А Тези конференцій	
Додаток В Лістинг коду	

ПЕРЕЛІК СКОРОЧЕНЬ

ADU (англ. Application Data Unit) блок даних прикладного рівня.

CRC (англ. Cyclic Redundancy Check) циклічна надлишкова перевірка.

MBAP (англ. MODBUS Application Protocol) прикладний протокол MODBUS.

PDU (англ. Protocol Data Unit) блок даних протокового рівня.

RTOS (англ. Real-Time Operating System) операційна система реального часу.

RTU (англ. Remote Terminal Unit) віддалений термінальний пристрій.

SCADA (англ. Supervisory Control And Data Acquisition) система диспетчерського керування та збору даних.

TCP (англ. Transmission Control Protocol) протокол керування передаванням даних.

UART (англ. Universal Asynchronous Receiver/Transmitter) універсальний асинхронний приймач-передавач.

АСКТП Автоматизована система керування технологічними процесами.

ВСТУП

Актуальність роботи. У сучасних автоматизованих системах керування технологічними процесами (АСКТП) спостерігається одночасна експлуатація обладнання різних поколінь, що суттєво відрізняються за принципами організації обміну даними. Значна частина промислових пристроїв продовжує використовувати інтерфейс протокол Modbus RTU, тоді як сучасні SCADA-системи, контролери та сервери функціонують у мережах Ethernet, застосовуючи протокол Modbus TCP. Така різnorodність комунікаційних середовищ породжує проблему сумісності та потребу у створенні універсальних засобів інтеграції, здатних забезпечити коректну взаємодію між пристроями без зміни їх апаратної або програмної частини.

Критичний аналіз наявних рішень свідчить, що існуючі промислові шлюзи часто мають низку обмежень — недостатню гнучкість конфігурації, неможливість масштабування, обмеження на кількість одночасних підключень, закриту архітектуру та низьку адаптивність до складних топологій реальних виробничих систем. Зростання потреб у цифровізації підприємств, модернізації застарілих об'єктів, інтеграції систем моніторингу та впровадженні гнучких технологічних платформ робить проблему інтеграції мереж Modbus TCP і Modbus RTU актуальною та такою, що потребує інженерного вирішення.

Метою кваліфікаційної роботи є розробка програмно-апаратного шлюзу, здатного забезпечити надійну двосторонню конвертацію між протоколами Modbus TCP і Modbus RTU, підтримувати гнучку маршрутизацію та стабільну роботу.

Завдання кваліфікаційної роботи:

- розробити апаратну платформу шлюзу;
- побудувати архітектуру програмного забезпечення з урахуванням специфіки обох протоколів;
- реалізувати механізми перетворення PDU/ADU;

- створити гнучкий механізм конфігурації маршрутизації.

Об'єкт дослідження — процес комунікації між різномірними промисловими мережами, що використовують протоколи Modbus TCP та Modbus RTU.

Предмет дослідження — методи та засоби конвертації, маршрутизації і узгодження параметрів взаємодії між протоколами Modbus TCP та Modbus RTU в умовах ресурсних обмежень програмно-апаратної платформи.

Методи дослідження :

- методи системного аналізу для визначення вимог до архітектури шлюзу;
- протокольний аналіз для моделювання та узгодження форматів кадрів Modbus TCP і Modbus RTU;
- методи аналізу часових характеристик для оцінки реакції системи під навантаженням;
- експериментальні методи для тестування роботи шлюзу у різних сценаріях;
- інженерні методи проектування вбудованого програмного забезпечення та організації паралельних задач у FreeRTOS.

Наукова новизна дослідження. У роботі вперше запропоновано комплексну архітектуру шлюзу, яка поєднує багатопоточну модель обробки запитів, гнучку таблицю маршрутизації Modbus-адрес та механізм динамічного конфігурування пристрою. Удосконалено методи організації обміну даними, а також запропоновано рішення щодо оптимального та безконфліктного доступу до ресурсів послідовного порта з декількох паралельних задач.

Практичне значення результатів кваліфікаційної роботи. Створений шлюз може бути використаний у промислових системах автоматизації, енергетиці, машинобудуванні, системах моніторингу та в інших сферах, де необхідно інтегрувати обладнання з інтерфейсом RS-485 у мережеву інфраструктуру Ethernet. Запропонована архітектура є масштабованою і може бути розширена для підтримки більшої кількості інтерфейсів, інших промислових протоколів або додаткових механізмів безпеки. Розроблений

прототип може слугувати основою для комерційного продукту або подальших наукових досліджень.

Публікації. Результати дослідження апробовано на XIV міжнародній науково-технічній конференції молодих учнів та студентів «Актуальні задачі сучасних технологій», XIII науково-технічній конференції «Інформаційні моделі, системи та технології» у вигляді тез конференцій.

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка містить вступ, 4 розділи, висновки, список використаних джерел та додатків.

РОЗДІЛ 1

АНАЛІЗ ТЕХНОЛОГІЙ ТА РІШЕНЬ ДЛЯ ІНТЕГРАЦІЇ МЕРЕЖ MODBUS

1.1. Актуальність теми

Індустріальна автоматизація зазнала суттєвої трансформації від автономних контролерів і простих послідовних шин до комплексних розподілених систем на базі Ethernet та IP-технологій [16]. Одночасно значна кількість польових пристроїв, що експлуатуються на критичних об'єктах, досі використовують традиційні протоколи й інтерфейси — зокрема Modbus RTU [24]. Повна заміна такого обладнання є економічно нераціональною, часто технічно неможливою через специфіку процесів або обмеження експлуатації. Отже, актуальною залишається задача забезпечення сумісності та інтеграції між усталеними послідовними мережами й сучасними IP-мережами без заміни кінцевого обладнання.

Поширення Modbus TCP як простого й відкритого рішення для Ethernet-зв'язку створює додаткові вимоги щодо організації взаємодії між численними TCP-клієнтами і вузько концептуалізованими RTU шинами, де доступ до середовища обмежений і визначається суворими таймінгами. Інтеграція цих двох підходів накладає набір інженерних та наукових завдань: розробка механізмів маршрутизації запитів, запобігання колізіям у RTU-шинах, збереження детермінованості часу відгуку, забезпечення надійності та стійкості в умовах електромагнітних завад і нестабільних ліній передачі [2].

Крім того, сучасні тенденції вимагають від шлюзових рішень не лише простого «перекладу» протоколів, а й розумного маршрутизування, масштабованості, можливостей для моніторингу і віддаленого керування, а також підтримки централізованого управління конфігураціями. Для наукової та практичної спільноти залишається важливим створення універсальних методів і

засобів, які дозволяли б поєднувати високу конфігурованість з детермінованістю роботи й мінімальними витратами на впровадження.

Таким чином, розробка та дослідження методів і засобів інтеграції мереж Modbus TCP і Modbus RTU на базі програмно-апаратного шлюзу мають як прикладну, так і наукову цінність: поєднання практичної потреби в модернізації реальних об'єктів з необхідністю вирішення низки нетривіальних інженерних задач визначає актуальність теми для магістерської кваліфікаційної роботи.

1.2. Мета і завдання дослідження

Метою даного дослідження є розроблення методів та програмно-апаратних засобів інтеграції мереж Modbus TCP і Modbus RTU з використанням універсального шлюзу, здатного забезпечувати сумісність між сучасними Ethernet-орієнтованими системами та обладнанням, що працює через послідовний інтерфейс RS-485. Досягнення цієї мети передбачає створення такого рішення, яке дозволяє прозоро та надійно транслювати запити між різними фізичними та протокольними середовищами, а також забезпечувати коректну маршрутизацію даних у складних промислових мережах.

Завдання дослідження полягає у формуванні теоретичних передумов та практичних принципів побудови шлюзу, здатного виконувати двосторонню конвертацію кадрів Modbus TCP ↔ Modbus RTU без втрати достовірності інформації. У межах дослідження необхідно проаналізувати особливості промислових протоколів, виявити ключові відмінності між їх структурою та логікою обміну, визначити обмеження щодо швидкодії та продуктивності, сформулювати підхід до організації черги запитів та вибору механізмів синхронізації між процесами обробки TCP-трафіку та доступу до RTU-шини. Важливою частиною завдань також є встановлення принципів маршрутизації, що дозволяють спрямовувати запити окремих клієнтів до конкретних пристроїв

у мережі RTU, запобігати конфліктам у доступі та забезпечувати ізоляцію окремих областей адресного простору.

Окрім цього, завдання дослідження охоплюють створення засобів конфігурації та управління параметрами шлюзу, які дозволяють адаптувати його до різноманітних топологій та забезпечують зручність експлуатації у промислових умовах. Важливо досягти стійкості та надійності роботи системи, що включає обробку помилкових пакетів, відновлення роботи після збоїв, а також забезпечення стабільної роботи у багатозадачному середовищі операційної системи реального часу. У підсумку дослідження має привести до створення методологічно обґрунтованого та практично підтвердженого підходу до побудови універсального шлюзу конвертації Modbus TCP ↔ Modbus RTU, який здатний функціонувати у широкому спектрі промислових застосувань та відповідати сучасним вимогам інтеграції різнорідних мереж [24].

1.3. Об'єкт і предмет дослідження

Об'єктом дослідження в даній роботі є інформаційні процеси, пов'язані з обміном даними в автоматизованих системах керування технологічними процесами, що включають комунікацію між пристроями польового рівня та мережевими системами верхнього рівня. Йдеться про широкий клас технічних засобів, до складу якого входять датчики, модулі збору даних, виконавчі механізми, контролери, сервери SCADA та інші компоненти промислової інфраструктури, що взаємодіють між собою через інтерфейси RS-485, Ethernet та протоколи обміну Modbus RTU і Modbus TCP [23]. У межах такого об'єкта особливу роль відіграють механізми передачі команд і телеметрії, способи організації доступу до фізичного середовища, взаємодія між вузлами з різними функціональними характеристиками, а також питання синхронізації та надійності комунікації у режимах реального часу.

Предметом дослідження є методи та програмно-апаратні засоби інтеграції різнорідних сегментів промислових мереж, що функціонують на основі протоколів Modbus TCP і Modbus RTU. Предмет дослідження охоплює структурні й алгоритмічні аспекти побудови шлюзів, здатних забезпечити двосторонню конвертацію запитів та відповідей без порушення логіки протоколів і фізичних обмежень середовищ передачі.

З огляду на це, предмет дослідження охоплює аналіз алгоритмів формування та обробки черг запитів, синхронізації між задачами, що працюють у різних контекстах протоколів, взаємодії програмних компонентів у середовищі операційних систем реального часу, а також підходи до маршрутизації запитів, коли кілька клієнтів мережі TCP звертаються до різних або спільних вузлів у мережі RTU. У такому контексті важливо враховувати механізми прив'язки Modbus-адрес до певних фізичних портів RS-485, способи логічної сегрегації доступу до окремих груп обладнання.

Предмет охоплює також проблематику конфігурування шлюзу, методи дистанційного управління параметрами, питання збереження налаштувань, організацію інтерфейсів для взаємодії з оператором або зовнішніми системами. У сучасних умовах, коли промислові об'єкти поступово переходять до цифровізованих систем керування, можливість централізованого або віддаленого управління конфігурацією шлюзу стає критично важливою, оскільки дозволяє гнучко адаптувати пристрій до різних топологій мережі, змінювати правила маршрутизації та контролювати поведінку шлюзу без необхідності фізичного доступу до обладнання.

Загалом об'єкт і предмет дослідження у цій роботі охоплюють комплексну технічну проблему, що лежить на перетині промислової автоматизації, теорії телекомунікацій, систем реального часу та інформаційних технологій. Предмет має прикладний характер, оскільки спрямований на забезпечення сумісності протоколів та різних фізичних середовищ, але водночас передбачає глибокий теоретичний аналіз процесів маршрутизації, трансляції та синхронізації

інформаційних потоків. Це надає дослідженню як інженерного, так і наукового значення, оскільки дозволяє формувати узагальнені методи побудови шлюзів промислових протоколів, що можуть бути використані у широкому спектрі систем АСКТП.

1.4. Огляд стандартів і технологій

Modbus RTU є одним із найбільш розповсюджених протоколів промислової автоматизації, який використовується для обміну даними між контролерами, датчиками та периферійним обладнанням у системах нижнього рівня. Його популярність пояснюється простою структурою кадрів, відкритою специфікацією та мінімальними вимогами до апаратних ресурсів, що робить його доступним для впровадження в широкому спектрі пристроїв. Початково розроблений для роботи через фізичні інтерфейси RS-232 і RS-485, протокол Modbus RTU зберіг свою актуальність навіть після появи більш сучасних мережевих технологій.

Ключовою особливістю Modbus RTU є його чутливість до часових характеристик лінії. Формат кадру в цьому протоколі не містить спеціальних маркерів початку або кінця пакета; межі кадру визначаються саме паузами між передаванням байтів [22]. Ця властивість забезпечує простоту реалізації, але одночасно створює додаткові вимоги до таймінгів, що накладає відповідальність на програмну й апаратну частину пристроїв. Невірно організовані паузи можуть призвести до некоректного зчитування даних або до пропуску кадрів, що особливо критично в умовах великої кількості запитів або шумних промислових середовищ.

У Modbus RTU використовується модель взаємодії типу “ведучий—ведений”, де запити ініціюються виключно ведучим пристроєм, який контролює цикл опитування мережі. Така модель гарантує відсутність колізій на фізичному рівні, що є суттєвою перевагою для шинних інтерфейсів RS-485 [19].

Кожен запит адресований конкретному пристрою за допомогою унікального ідентифікатора (Slave ID), що дає можливість будувати на одній шині до 247 підлеглих пристроїв. Разом з тим, обмеження у вигляді лише одного ведучого робить протокол менш гнучким у масштабованих та динамічних системах, де кількість клієнтів може змінюватися. На рис. 1.1 наведено типову структуру мережі Modbus RTU з використанням шинної архітектури.



Рис. 1.1. Типова структура мережі Modbus RTU

Структура кадру Modbus RTU є відносно простою. Кадр містить байт адреси, код функції, область даних та контрольну суму CRC16. Відсутність складних заголовків дозволяє швидко обробляти кадри навіть на мікроконтролерах із невеликим обсягом пам'яті, що є однією з причин, чому Modbus RTU так широко використовується у системах виробничої автоматизації. Стандарт охоплює як базові функції читання та запису регістрів, так і розширені можливості, включно з діагностикою, протоколюванням та передаванням великих блоків даних.

Проте, незважаючи на простоту реалізації, Modbus RTU має низку обмежень, які необхідно враховувати при інтеграції з сучасними Ethernet-мережами. Серед них варто відзначити обмежену пропускну здатність, що залежить від швидкості передавання на RS-485; значну чутливість до

промислових перешкод; необхідність дотримання суворих таймінгів; відсутність уніфікованих механізмів ускладнення протоколу, таких як фрагментація або асинхронний обмін. Це створює низку технічних викликів під час побудови шлюзу між Modbus RTU та Modbus TCP, де логіка взаємодії суттєво відрізняється. Наприклад, необхідно забезпечити сумісність одночасних TCP-з'єднань з єдиною послідовною шиною, синхронізувати таймінги, уникати перенавантаження RTU-боку запитами, що надходять із мережі Ethernet, та гарантувати коректне формування і розпізнавання меж кадру.

Таким чином, протокол Modbus RTU залишається ключовою складовою сучасних АСКТП, але його ефективне використання в умовах інтеграції з мережами Ethernet потребує застосування спеціалізованих шлюзів, здатних коректно виконувати трансляцію протоколів, враховуючи обмеження серійного обміну та вимоги до детермінованого доступу.

На відміну від послідовного протоколу Modbus RTU, який працює в умовах обмеженої пропускну здатності та вимагає суворого дотримання таймінгів, Modbus TCP орієнтований на сучасні мережеві інфраструктури та забезпечує значно ширші можливості для масштабування, паралельної роботи та інтеграції у складні промислові системи [21]. Його поява стала логічним етапом розвитку протоколу Modbus після переходу багатьох галузей на мережі Ethernet і IP, які поступово витіснили традиційні польові шини як у локальних, так і у розподілених системах керування.

Modbus TCP використовує стек протоколів TCP/IP, а отже, працює поверх мережевих та транспортних рівнів, які відповідають за маршрутизацію, доставку сегментів без втрат, забезпечення цілісності даних та можливість багатозадачної взаємодії. Завдяки цьому протокол повністю позбавлений характерних для RTU обмежень, пов'язаних з довжиною лінії, електромагнітними завадами, синхронізацією та кодуванням кадрів. Доступ до середовища передачі визначається правилами комутації Ethernet, а не логікою ведучого пристрою, як це відбувається в RTU. Це дає можливість організувати

одночасну роботу багатьох клієнтів, що звертаються до одного або кількох Modbus-серверів.

Одним із ключових елементів Modbus TCP є спеціальний заголовок MBAP (Modbus Application Protocol Header), який передує даним прикладного протоколу. На відміну від RTU-формату, де межі кадру визначаються паузами, у Modbus TCP кадр має чітко структурований формат зі службовими полями для ідентифікації транзакцій та визначення довжини переданих даних. Такий підхід підвищує гнучкість протоколу, дозволяє підтримувати паралельні транзакції від одного клієнта та зменшує ризики помилкового інтерпретування кадру в умовах інтенсивного мережевого трафіку.

Modbus TCP не потребує механізму CRC-контролю, оскільки TCP/IP-стек уже забезпечує перевірку цілісності пакетів. Проте така перевага не є абсолютною, адже за відсутності окремого поля CRC старші рівні TCP/IP повністю відповідають за виявлення помилок. Це означає, що правильність реалізації протоколу на рівні застосунку не менш важлива, ніж у RTU, особливо тоді, коли шлюз має обробляти десятки одночасних підключень і забезпечувати їх коректне маршрутизування до спільного серійного каналу.

Завдяки використанню TCP протокол набуває можливості встановлювати численні паралельні з'єднання, які можуть надходити як від незалежних ПК, SCADA-систем або верхнього рівня керування, так і від окремих програмних компонентів. У реальних умовах це означає, що один Modbus-сервер може одночасно обслуговувати кілька клієнтів, які виконують різні операції та звертаються до різних регістрів. У той час як RTU-доступ до шини суворо послідовний, Modbus TCP підтримує модель, близьку до багатопоточності, що створює нові виклики для побудови шлюзів, які повинні ефективно узгоджувати паралельність у TCP-мережі із суворою послідовністю в RTU-середовищі.

Ще однією важливою характеристикою Modbus TCP є значно ширший діапазон швидкодії. Пропускна здатність мережі Ethernet дозволяє передавати великі масиви даних у дуже короткий час, що робить протокол придатним для

високопродуктивних систем. Саме тому Modbus TCP часто використовується у сучасних SCADA-системах, промислових контролерах, модулях збору даних та інтелектуальних пристроях, де необхідна швидка реакція на події та можливість централізованого опитування.

Однією з унікальних особливостей Modbus TCP, важливою з погляду побудови шлюзу, є механізм використання поля Unit ID. Це поле слугує індикатором логічного адресата в системах, де Modbus TCP використовується для доступу до RTU-мереж. Саме через Unit ID визначається, який підлеглий пристрій у RTU-шині має отримати конкретний запит. Відповідно, шлюз повинен виконувати коректну інтерпретацію цього поля, перетворювати MBAP-заголовок у RTU-адресу та правильно синхронізувати обробку запитів між Ethernet та RS-485. На рис. 1.2 представлено спрощену структуру стандартного Modbus TCP кадру із зазначенням його логічних полів.



Рис. 1.2. Структура кадру Modbus TCP з MBAP-заголовком

З точки зору інженерії програмних систем, Modbus TCP також визначає певні вимоги до реалізації програмного забезпечення шлюзів. Оскільки TCP є орієнтованим на з'єднання, шлюз повинен підтримувати повноцінний стек мережевої взаємодії, зокрема логіку створення, обслуговування та завершення з'єднань, механізми очищення ресурсів у разі аварійного розриву сесії, контроль тайм-аутів і управління буферами. У системах реального часу ці задачі повинні узгоджуватися з пріоритетами інших процесів, наприклад, обробкою

UART-портів або таймерів RTU-протоколу, що вимагає ретельного підходу до побудови архітектури ПЗ.

Таким чином, Modbus TCP є сучасним та потужним мережевим протоколом, що забезпечує масштабованість, гнучкість та швидкодію, але водночас потребує ретельно продуманих механізмів трансляції й синхронізації при інтеграції з мережами Modbus RTU. Саме ця різниця у принципах роботи двох протоколів зумовлює складність побудови ефективних програмно-апаратних шлюзів, здатних забезпечувати коректну обробку запитів у реальному часі.

1.5. Огляд існуючих рішень і наукових праць

На сучасному ринку промислових рішень існує широка лінійка апаратних і програмних засобів, призначених для забезпечення взаємодії між мережами Modbus TCP і Modbus RTU. Потреба у таких засобах виникає як у випадках модернізації існуючих систем, так і при побудові нових комплексів автоматизації, де одночасно застосовуються як застарілі, так і сучасні комунікаційні технології. Огляд існуючих рішень дозволяє сформулювати цілісне уявлення про поточний стан галузі та виявити недоліки й обмеження, які потребують подолання при розробці нових шлюзових засобів.

Серед провідних виробників апаратних шлюзів можна відзначити компанії Моха, Wago, ICP DAS, HMS та інші, що пропонують пристрої різної продуктивності та функціональності. Типовою рисою таких рішень є орієнтація на надійність та стабільну роботу в промислових умовах, де від пристрою очікується безперервне функціонування протягом тривалих періодів без ручного втручання. У більшості випадків комерційні шлюзи побудовані на спеціалізованих апаратних платформах, що включають кілька послідовних портів RS-485 та один або кілька інтерфейсів Ethernet. Архітектура таких пристроїв здебільшого передбачає мінімальні можливості конфігурації

маршрутизації: запити від TCP-клієнтів направляються на відповідні серійні порти відповідно до налаштувань конфігураційних меню, але глибока прив'язка між окремими підключеннями або адресними областями зазвичай не підтримується. У деяких моделях реалізовано фіксовані таблиці маршрутизації, однак їх гнучкість обмежена виробником і не дозволяє виконувати складні сценарії, пов'язані з доступом різних клієнтів до різних вузлів у RTU-мережі.

Під час аналізу комерційних пристроїв помітним є також те, що більшість із них використовує універсальні операційні системи або власні прошивки, які не завжди оптимізовані для високого паралельного навантаження. У разі підключення великої кількості TCP-клієнтів такі шлюзи можуть знижувати продуктивність або демонструвати непередбачувану поведінку, пов'язану з тим, що реалізація обробки запитів у багатьох випадках є лінійною і не враховує можливість одночасного доступу. Окрім цього, типовим обмеженням є невелика кількість одночасних TCP-з'єднань, що рідко перевищує 3–5, а в бюджетних моделях — навіть менше.

Ще одним напрямком інтеграції є програмні шлюзи, які реалізуються у вигляді модулів SCADA-систем, серверних застосунків або спеціалізованих програмних компонентів для Linux та Windows. Такі рішення часто використовуються при переході на IP-орієнтовані системи управління або в умовах, коли існує потреба в централізованому керуванні великою кількістю пристроїв. На відміну від апаратних шлюзів, програмні мають значно вищу гнучкість, але при цьому позбавлені апаратних засобів захисту від промислових завад, що робить їх менш придатними до використання на польовому рівні.

У наукових публікаціях і технічній літературі розглядається низка підходів до побудови промислових шлюзів, які відрізняються алгоритмами синхронізації запитів, методами кешування даних або оптимізованими способами обробки трафіку для зменшення затримок [1]. Багато дослідників приділяють увагу впливу мережевих параметрів на продуктивність шлюзів, аналізуючи, як збільшення кількості TCP-клієнтів або зростання частоти

опитування впливає на час проходження команд через RTU-лінію. Окремим напрямком наукових робіт є питання забезпечення безпеки промислових шлюзів, де розглядаються механізми фільтрації та виявлення атипових запитів, а також методи ізоляції технологічних сегментів від зовнішніх мереж.

Попри значну кількість існуючих рішень, загальною проблемою для багатьох шлюзів залишається обмежена масштабованість. Більшість рішень погано працює у сценаріях, коли кількість клієнтів або інтенсивність обміну суттєво збільшується. Це пов'язано з тим, що архітектури програмних моделей часто не враховують принципи пріоретизації, адаптивного планування чи багаторівневої маршрутизації. Водночас промислові системи дедалі частіше вимагають не лише стабільної передачі даних, але й можливості гнучко розмежовувати доступ до окремих ділянок мережі, організовувати різні режими роботи шлюзу та забезпечувати відмовостійкість під час роботи у складних умовах.

Ще одним аспектом, що потребує розгляду, є вплив апаратної платформи на ефективність роботи шлюзів. Пристрої, які побудовані на простих мікроконтролерах без операційної системи або зі спрощеними циклами обробки, часто демонструють обмежені можливості у паралельній обробці TCP-трафіку. З іншого боку, рішення, які використовують повноцінні операційні системи Linux на потужних процесорах, здебільшого є дорожчими та споживають більше ресурсів, ніж це потребує типове застосування на польовому рівні, де головну роль відіграють компактність, невелике енергоспоживання та простота інтеграції з існуючими мережами.

Таким чином, огляд існуючих апаратних, програмних і наукових рішень показує, що хоча проблема інтеграції мереж Modbus TCP і Modbus RTU активно досліджується та має широке практичне застосування, значна кількість рішень має обмеження, пов'язані з недостатньою гнучкістю, низькою масштабованістю, відсутністю ефективних механізмів маршрутизації та невисокою стійкістю до великих навантажень. Саме ці недоліки зумовлюють

необхідність створення нових підходів до побудови шлюзів, орієнтованих на сучасні вимоги промислових систем, включно з підтримкою багатьох одночасних клієнтів, динамічним налаштуванням та можливістю адаптації до різних топологій мереж.

1.6. Вимоги до шлюзу

Під час розроблення програмно-апаратного шлюзу для інтеграції мереж Modbus TCP і Modbus RTU постає завдання визначення таких вимог, які б забезпечили його стабільну роботу, сумісність з широким спектром промислового обладнання та здатність функціонувати в умовах реального виробничого навантаження. Вимоги до шлюзу охоплюють функціональні можливості, які визначають його основну поведінку в системі, а також нефункціональні характеристики, що впливають на якість, продуктивність та надійність роботи. Обидві групи вимог є взаємопов'язаними і мають бути враховані під час формування архітектури пристрою та вибору технічних засобів його реалізації.

З позиції функціональності шлюз повинен забезпечувати повноцінну двосторонню конвертацію між протоколами Modbus TCP і Modbus RTU, зберігаючи семантичну відповідність команд, форматів кадрів та логіки обміну даними. Він має коректно інтерпретувати заголовки МВАР у TCP-пакетах та відповідно транслювати їх у формат RTU-кадрів. Важливо, щоб шлюз міг обробляти різні типи команд Modbus, включно з читанням регістрів, записом окремих значень, читанням блоків даних і операціями діагностики. Поведінка шлюзу повинна забезпечувати точну відповідність між структурами запитів і відповідей у залежності від вимог протоколів, не допускаючи втрати полів або некоректного формування контрольних сум CRC, що можуть призвести до збоїв у роботі підлеглих пристроїв.

Крім того, функціональна частина шлюзу має включати механізми маршрутизації. Вони визначають, який TCP-клієнт може отримати доступ до конкретного пристрою у RTU-мережі, і як саме здійснюється прив'язка Unit ID або IP-адреси до відповідного фізичного порту. Адекватна маршрутизація дозволяє уникнути конфліктів між клієнтами, запобігти ситуаціям, коли кілька з'єднань одночасно намагаються звертатися до одного й того ж пристрою, та забезпечити керованість інформаційних потоків. У реальних умовах шлюз повинен забезпечувати стабільність роботи навіть тоді, коли кількість клієнтів динамічно змінюється або в системі присутні пристрої з різними часовими характеристиками опитування.

З огляду на характер промислових застосувань важливе значення мають нефункціональні вимоги до продуктивності шлюзу. Пристрій повинен забезпечувати достатню пропускну здатність для обробки інтенсивних потоків даних, особливо в умовах, коли мережа Ethernet містить кілька SCADA-клієнтів або програмних компонентів, що регулярно опитують пристрої RTU [22]. Продуктивність визначається не лише швидкістю процесора або пропускну здатністю Ethernet-інтерфейсу, а й ефективністю алгоритмів обробки запитів, логікою побудови черг, здатністю системи коректно встановлювати часові паузи між RTU-кадрами та уникати перевантаження UART-портів. Системи, що працюють у реальному часі, вимагають мінімізованої затримки між отриманням пакета, його перетворенням і пересиланням до цільового пристрою. У разі недотримання таких вимог можуть виникати ситуації, коли цикл опитування перевищує допустимі межі або знижується частота оновлення телеметрії.

Не менш важливою нефункціональною характеристикою шлюзу є надійність. У промислових умовах пристрій повинен працювати безперервно протягом тривалих періодів, бути стійким до збоїв живлення, електромагнітних перешкод та особливостей навколишнього середовища [20]. Це вимагає наявності механізмів відновлення після помилок, зокрема очищення та повторної ініціалізації TCP-з'єднань, обробки завислих запитів, контролю

тайм-аутів та можливості автоматичного перезапуску окремих функціональних компонентів. Важливим аспектом є також стійкість до пошкоджених або неправильно сформованих кадрів, які можуть бути результатом зовнішніх впливів або помилок клієнтського обладнання.

Ще одним важливим компонентом є масштабованість. Сучасні підприємства часто розвиваються швидкими темпами, що вимагає можливості адаптувати конфігурацію шлюзу до різних топологій мереж і до змін у кількості клієнтів. Шлюз повинен коректно обробляти як поодинокі з'єднання, так і значні обсяги запитів, забезпечуючи стабільну роботу в умовах різної інтенсивності трафіку.

Узагальнюючи, вимоги до шлюзу Modbus TCP ↔ Modbus RTU охоплюють цілий спектр параметрів, починаючи від коректності трансляції протоколів і закінчуючи стійкістю до навантаження, можливостями масштабування та безпекою роботи в умовах реального промислового середовища. Відповідність цим вимогам формує основу для розроблення ефективного програмно-апаратного рішення, яке здатне забезпечити надійну та безпечну інтеграцію різнорідних мереж у системах автоматизації.

1.7. Підсумки аналітичного огляду

У ході аналітичного огляду, представленого в цьому розділі, було сформовано цілісне уявлення про сучасний стан технологій, пов'язаних з інтеграцією мереж Modbus TCP і Modbus RTU, а також визначено основні передумови та вимоги до створення програмно-апаратного шлюзу. Розгляд історичних передумов, особливостей протоколів і фізичних інтерфейсів показав, що попри активний розвиток мережевих Ethernet-технологій, значна кількість промислових систем продовжує функціонувати на основі RS-485 і Modbus RTU, що зумовлює необхідність у гнучких та надійних засобах їх інтеграції з сучасними IP-орієнтованими системами. Актуальність проблеми

визначається потребою збереження існуючого обладнання, забезпечення сумісності поколінь технологій, підвищення рівня керованості та створення масштабованих архітектур промислових мереж.

Аналіз наукових праць і комерційних рішень показав, що хоча на ринку існує значна кількість шлюзів для конвертації Modbus-протоколів, більшість із них має низку обмежень, пов'язаних із недостатньою масштабованістю, низькою гнучкістю маршрутизації або недостатнім рівнем адаптивності до особливостей конкретних об'єктів автоматизації. У наукових дослідженнях значну увагу приділяють питанням синхронізації протоколів, зменшенню затримок, побудові алгоритмів доступу до загальної RTU-шини та оптимізації взаємодії між численними TCP-клієнтами. Проте на практиці багато рішень стикаються з труднощами, пов'язаними з високою інтенсивністю трафіку або з обробкою нетипових сценаріїв роботи, що робить проблему побудови ефективного шлюзу актуальною та науково значущою.

Важливим результатом проведеного аналізу стало визначення вимог, яким повинен відповідати сучасний шлюз для інтеграції Modbus TCP і Modbus RTU. До них належать як функціональні аспекти, пов'язані з реалізацією механізмів маршрутизації та можливістю адаптації до різних топологій мереж, так і нефункціональні параметри, що визначають якість роботи шлюзу: продуктивність у багатокористувацьких сценаріях, надійність при великих навантаженнях, а також здатність працювати в умовах безперервного функціонування.

Таким чином, аналітичний розділ заклав теоретичну основу для подальшої роботи, визначивши ключові технології, способи їх взаємодії та проблемні аспекти інтеграції. Сформовані висновки дозволяють перейти до теоретичної частини дослідження, де буде детальніше розглянуто методи побудови архітектури шлюзу, принципи синхронізації протоколів, способи організації багатозадачності й алгоритми обробки запитів, що забезпечують ефективне узгодження мереж Modbus TCP і Modbus RTU.

РОЗДІЛ 2

ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ ПРОГРАМНО-АПАРАТНОГО ШЛЮЗУ MODBUS

2.1. Загальна модель архітектури програмно-апаратного шлюзу

Архітектура програмно-апаратного шлюзу, що забезпечує інтеграцію мереж Modbus TCP і Modbus RTU, ґрунтується на поєднанні двох рівнів: апаратного, який відповідає за фізичні інтерфейси та низькорівневі механізми передачі даних, та програмного, який реалізує логіку конвертації протоколів, маршрутизацію запитів і управління інформаційними потоками між різними середовищами. Загальна модель побудови шлюзу передбачає створення єдиного комунікаційного ядра, яке синхронізує роботу мережкових і послідовних інтерфейсів, забезпечує узгодженість протоколів і виконує необхідні перетворення форматів даних відповідно до вимог протоколів Modbus.

У типовому випадку апаратна основа шлюзу складається з мікроконтролера з достатньою обчислювальною потужністю, окремого апаратного Ethernet-контролера або PHY-мікросхеми, а також одного чи кількох інтерфейсів RS-485, які забезпечують роботу з послідовною шиною. У досліджуваному проєкті використано платформу на основі STM32F407ZGT6, яка поєднує високу тактову частоту, апаратне прискорення для обробки операцій з пам'яттю, наявність апаратних таймерів та низку периферійних модулів, що робить її оптимальною для розв'язання задач реального часу. Для забезпечення коректної роботи мережевого інтерфейсу застосовано PHY DP83848, який забезпечує сумісність з Ethernet 10/100 Мбіт/с та гарантує стабільний зв'язок у промислових умовах. Послідовні інтерфейси RS-485 реалізовано через гальванічно розв'язані трансивери, що дозволяє захистити мікроконтролер від потенційних завад та різниці потенціалів у польових мережах.

Програмна архітектура шлюзу побудована на основі операційної системи реального часу FreeRTOS, що дає змогу організувати багатозадачну модель обробки подій. В окремих задачах функціонують TCP-сервери, менеджери підключень, обробники UART-інтерфейсів, а також центральний маршрутизатор Modbus-кадрів. Такий підхід дозволяє виділити логічно незалежні компоненти, оптимізувати час реакції, мінімізувати блокування ресурсів та гарантувати детермінованість обробки запитів у критичних сценаріях. Завдяки FreeRTOS реалізовано можливість гнучкого керування пріоритетами задач, що дозволяє, наприклад, надавати вищий пріоритет обробці кадрів RTU, уникаючи зривів таймінгів або втрати кадрів через інтенсивне Ethernet-навантаження. На рис. 2.1 наведено узагальнену структурну схему архітектури програмно-апаратного шлюзу, що демонструє взаємодію основних підсистем.

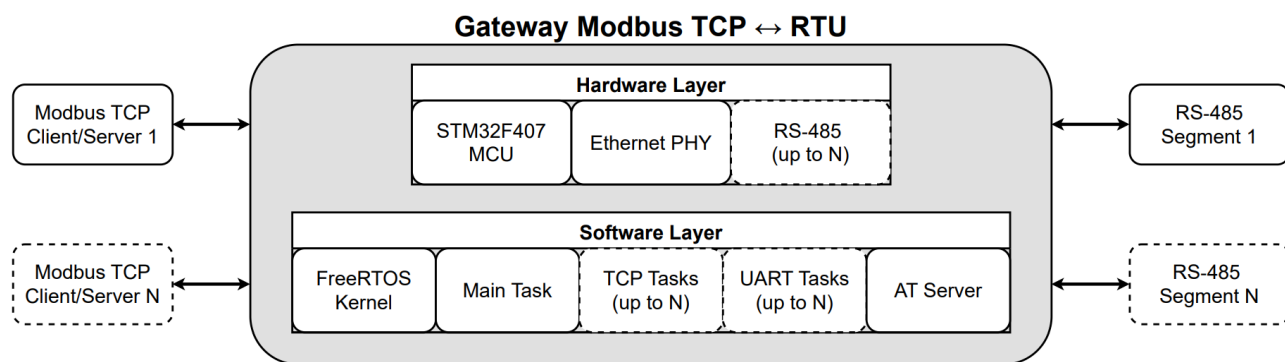


Рис. 2.1. Загальна архітектура програмно-апаратного шлюзу
Modbus TCP↔ Modbus RTU

Центральним елементом програмної архітектури шлюзу є основна системна задача, яка ініціалізує всі допоміжні процеси та здійснює моніторинг їх працездатності протягом усього часу функціонування пристрою. Під час старту пристрою вона відповідає за ініціалізацію апаратних модулів мікроконтролера, конфігурацію периферії, запуск стеку lwIP, створення об'єктів синхронізації FreeRTOS (черги, семафори, mutex), а також послідовний запуск функціональних задач: серверів TCP, UART-обробників, задачі

АТ-сервера та діагностичних служб. Усім задачам призначаються фіксовані пріоритети, що відповідають їх ролі та часовій критичності, після чого основна задача переходить у режим циклічного контролю.

Однією з ключових функцій основної задачі є перевірка активності та стану кожної задачі за допомогою внутрішнього механізму «живих маркерів». Кожна задача періодично оновлює власний статус у загальній структурі моніторингу, підтверджуючи, що вона виконана без зависань, переповнення черг або порушення таймінгів. Основна задача періодично опитує ці маркери та аналізує індикатори виконання. Якщо для певної задачі не оновився «живий маркер» протягом дозволеного інтервалу, система визначає її як таку, що «вийшла з робочого стану».

Завдяки власному циклу моніторингу основна задача також виконує функцію програмного сторожа (soft-watchdog), доповнюючи апаратний watchdog мікроконтролера. Вона скидає апаратний WWDG/IWDG лише тоді, коли переконана, що всі задачі працюють коректно. Таким чином забезпечується гарантований контроль над усім програмним середовищем: якщо одна з критичних задач зависає у некерованому стані, основна задача перестає скидати watchdog, і пристрій автоматично перезавантажується, відновлюючи працездатність у промисловому середовищі.

Комунікаційна модель шлюзу базується на концепції модуля конвертації Modbus, який приймає кадри з мережі TCP, інтерпретує MBAP-заголовки та трансформує їх у RTU-формат, забезпечуючи коректне формування CRC, встановлення адреси підлеглого пристрою. Зворотній напрямок — передача RTU-відповіді назад у TCP-сесію — виконується з урахуванням логіки маршрутизації, що дозволяє визначити, який клієнтський запит відповідає отриманому RTU-кадру. Такий двоспрямований процес формує модуль функціональності шлюзу та визначає більшість вимог до узгодженості його роботи.

Модуль конвертації встановлює відповідність між транзакціями TCP і адресними просторами RTU-мережі. Він може враховувати параметри, отримані

з конфігураційного інтерфейсу, такі як Unit ID, IP-адреса клієнта та вибраний TCP-порт. У розробленому пристрої маршрутизація є гнучкою і визначається набором параметрів, у яких можна вказати режим роботи, прив'язку до конкретного серійного порту, адресу пристрою Modbus, таймаути та правила доступу. Центральна логіка модуля забезпечує, щоб кожен запит TCP отримував коректного адресата у мережі RTU, а конфліктні ситуації, пов'язані з одночасним доступом до одного пристрою, оброблялися коректно та без порушення роботи шини.

Загальна архітектура шлюзу, таким чином, являє собою багаторівневу систему, де апаратні компоненти забезпечують фізичний доступ до мереж, а програмні — реалізують логіку перетворення та синхронізації. Такий підхід дозволяє створити гнучку, масштабовану та надійну систему, придатну для використання в реальних промислових умовах.

2.2. Розробка внутрішньої логіки конвертації

Внутрішня логіка конвертації між протоколами Modbus TCP та Modbus RTU є ключовим компонентом програмної архітектури шлюзу і визначає спосіб взаємодії між мережевими та послідовними інтерфейсами. Оскільки обидва протоколи використовують однакову модель прикладного рівня (Modbus Application Protocol), але відрізняються структурою транспортних кадрів, фізичними характеристиками каналу та механізмами синхронізації, процес конвертації потребує точного дотримання специфікацій форматів PDU та ADU для кожного середовища.

Специфікація Modbus визначає PDU (Protocol Data Unit) як універсальну структуру даних, що складається з коду функції та блоку параметрів. Ця структура повністю ідентична для TCP та RTU, що значно спрощує процес конвертації. Різниця полягає у структурі ADU (Application Data Unit), який включає у себе PDU та службові поля, специфічні для транспортного середовища. У Modbus RTU ADU містить адресу підлеглого пристрою та

CRC-контрольну суму, тоді як у Modbus TCP формат ADU будується навколо MBAP-заголовка, що включає ідентифікатор транзакції, ідентифікатор протоколу, довжину повідомлення та поле Unit ID.

У межах програмної моделі шлюзу конвертація здійснюється у два етапи: спершу відокремлюється PDU, яке є спільною частиною обох протоколів, а після цього виконується формування нового ADU відповідно до специфікації цільового середовища. При цьому логіка обробки значною мірою залежить від того, як саме організовано роботу задач FreeRTOS та комунікацію між різними інтерфейсами.

Через те, що Ethernet-стек у пристрої ґрунтується на lwIP, кожне активне TCP-з'єднання обслуговується окремою задачею, яка відповідає за приймання та передачу даних, підтримку сесії та обробку можливих винятків, пов'язаних із мережею. Ця модель дозволяє рівномірно розподілити навантаження між задачами, але водночас породжує ситуацію, коли багато клієнтів можуть одночасно створювати запити. Саме тому необхідно забезпечити коректне узгодження доступу до загального ресурсу — RTU-каналу. На боці UART інтерфейс реалізовано через HAL UART у режимах DMA та interrupt, що дає змогу уникати простоїв процесора під час передавання та приймання даних. UART-драйвер побудований у модульній формі, що дозволяє перенести його у середовище Linux або Windows без значних змін у програмному коді, що є важливим аспектом масштабованості й супроводу.

Усі запити, отримані від TCP-клієнтів, транслюються у єдину чергу повідомлень, доступ до якої контролюється через м'ютекс. Такий підхід дозволяє синхронізувати обробку даних між задачами, що працюють паралельно, і гарантує, що лише одна задача має право доступу до UART у певний момент часу. На відміну від систем, де маршрутизатор повністю контролює модель “ведучий—підлеглий”, у даному шлюзі логіка протоколу не змінюється: шлюз не бере на себе роль ведучого, а лише передає запити від TCP та очікує відповідей від RTU-вузлів, повністю зберігаючи семантику стандарту Modbus.

Під час конвертації з TCP до RTU особливе значення має обробка поля Unit ID, оскільки саме воно визначає адресу цільового пристрою у послідовній мережі. На етапі трансформації шлюз вилучає МВАР-заголовок, зберігаючи ідентифікатор транзакції у внутрішній структурі, та формує RTU-кадр відповідно до вимог протоколу, включно з обчисленням CRC16. У зворотному напрямку — з RTU до TCP — шлюз відновлює МВАР-заголовок та виконує доставку відповіді у відповідну TCP-сесію, використовуючи збережений ідентифікатор транзакції. Такий механізм дозволяє підтримувати паралельну роботу багатьох TCP-клієнтів і гарантує правильне зіставлення відповідей з відповідними запитами.

Таким чином, внутрішня логіка конвертації у шлюзі має багаторівневу структуру, де поєднуються апаратні механізми UART із DMA, програмні механізми lwIP та FreeRTOS, а також алгоритмічні механізми формування Modbus-кадрів. Такий підхід забезпечує гнучкість, масштабованість і коректність обробки запитів у режимі реального часу, що є критично важливим для промислових систем.

2.3. Моделі маршрутизації та правила трансляції

Механізм маршрутизації у програмно-апаратному шлюзі відіграє ключову роль у забезпеченні коректної та передбачуваної взаємодії між мережами Modbus TCP та Modbus RTU. Оскільки моделі комунікації цих протоколів істотно відрізняються — зокрема можливістю багатопотокової взаємодії у TCP та строгою послідовністю в RTU — система маршрутизації повинна компенсувати ці відмінності та гарантувати узгоджений потік даних у обох напрямках.

У запропонованій архітектурі маршрутизація реалізується через набір параметрів, де кожен порт може бути налаштований у режимі TCP-сервера або TCP-клієнта та пов'язаний із певним фізичним інтерфейсом або маршрутом у мережі RTU. При цьому він не є жорстко прив'язаним до одного конкретного

пристрою Modbus, що істотно розширює можливості системи. Один TCP-клієнт, може здійснювати доступ до кількох пристроїв у RTU-мережі, а поле Unit ID визначає кінцевого адресата конкретного запиту. Такий підхід відповідає логіці протоколу Modbus TCP, де Unit ID використовується як засіб вказування адреси підлеглого пристрою у разі роботи через шлюз.

Гнучкість системи маршрутизації обмежується не фіксованою кількістю пристроїв, а лише доступною пам'яттю мікроконтролера. Маршрутизатор працює за принципом логічного зв'язування контекстів TCP і RTU. Коли TCP-клієнт надсилає пакет Modbus TCP, шлюз аналізує відповідний МВАР-заголовок, визначає поле Unit ID та зіставляє його з конфігураційними параметрами, що зберігаються у налаштуваннях. Саме цей механізм дозволяє визначити, чи має даний клієнт право доступу до запитуваної адреси, і який саме серійний порт RS-485 повинен бути використаний для передачі кадру. Така модель є масштабованою і дозволяє одному клієнту працювати з дуже різними пристроями у межах тієї самої RTU-шини, а також забезпечує ізоляцію між різними TCP-з'єднаннями, що можуть працювати незалежно один від одного.

У зворотному напрямку, тобто під час передачі відповідей з RTU до TCP, шлюз використовує збережений контекст транзакції, де ID транзакції служить ключовим елементом для правильного відновлення логічного зв'язку між відправленим запитом і відповіддю. Після отримання відповіді від RTU-пристрою відповідний МВАР-заголовок відтворюється, і пакет повертається у TCP-сесію саме того клієнта, який ініціював запит. Такий підхід дозволяє гарантувати коректність роботи навіть тоді, коли кілька клієнтів звертаються до різних пристроїв у RTU-мережі паралельно, а час опитування суттєво варіюється залежно від особливостей кожного пристрою.

Особливістю реалізованої архітектури є те, що маршрутизація не втручається у модель master/slave протоколу Modbus і не змінює логіку взаємодії у RTU-сегменті. Шлюз не виконує роль майстра для RTU-мережі у традиційному сенсі. Натомість він передає запити у тому вигляді, у якому вони надходять із мережі TCP, що дозволяє уникнути формування власних циклів

опитування або зміни стану пристроїв. Це робить реалізацію максимально прозорою для обох сторін і забезпечує повну сумісність з існуючим обладнанням [26].

У складних сценаріях, коли у системі присутні кілька фізичних або віртуальних RTU-шин, маршрутизатор здатний розмежовувати їх між собою за допомогою параметрів у налаштуваннях. Модель маршрутизації дозволяє будувати багаторівневі схеми доступу, де одна частина TCP-клієнтів працює з певною групою пристроїв, тоді як інші клієнти обслуговують інші адресні сегменти. Це робить систему гнучкою і здатною масштабуватися відповідно до вимог конкретної промислової інфраструктури.

Таким чином, у реалізованому шлюзі модель маршрутизації базується на гнучкій таблиці відповідностей, де TCP-сесії, Unit ID, конфігураційні параметри та фізичні серійні порти взаємодіють у єдиній структурі. Такий підхід забезпечує надійне узгодження пакетів у обох напрямках, дозволяє обслуговувати багато клієнтів без порушення логіки RTU-шини, а також забезпечує можливість масштабування системи без необхідності суттєвого перепроєктування архітектури.

2.4. Алгоритми обробки запитів і черг

Алгоритми обробки запитів і організація черг у програмно-апаратному шлюзі відіграють визначальну роль у забезпеченні стабільної роботи системи в умовах багатопотокової взаємодії та різнорідного навантаження, притаманного одночасному використанню Modbus TCP і Modbus RTU. Оскільки обидва протоколи функціонують у середовищах із різною природою синхронізації — система повинна правильно узгоджувати ці відмінності, забезпечуючи коректність передачі.

У реалізованому шлюзі обробка запитів базується на багатозадачній моделі FreeRTOS, де кожен TCP-порт має власну задачу, що відповідає за прийом та обробку запитів, підтримку TCP-сесій і передачу кадрів у внутрішню

систему маршрутизації. Подібним чином кожен послідовний порт має власну задачу, що обробляє операції прийому та передачі через UART, працює зі спеціалізованим DMA-буфером і виконує формування RTU-кадрів. Завдяки такому розподілу система отримує можливість паралельно обслуговувати кілька незалежних потоків, що істотно підвищує загальну продуктивність та дозволяє використовувати переваги апаратної багатозадачності мікроконтролера.

Алгоритм планування у системі базується на пріоритетах задач FreeRTOS, де задачі UART-сегмента мають підвищений пріоритет порівняно із задачами TCP-сегмента. Забезпечення пріоритетного доступу задач UART до процесора дозволяє мінімізувати затримки в RTU-обміні та гарантує коректність надсилання і приймання кадрів. Навпаки, задачі TCP обслуговують клієнтів у режимі, де часова критичність значно нижча, а мережа сама компенсує можливі варіації затримки, властиві IP-трафіку.

Комунікація між TCP-задачами і задачами UART організована через механізм черг повідомлень та mutex-захисту. Коли TCP-клієнт надсилає запит, задача, що його обслуговує, витягує PDU із MBAP-заголовка та формує внутрішнє повідомлення, яке містить всю інформацію, необхідну для маршрутизації — контекст TCP-сеансу, ID транзакції, Unit ID та адресу RTU-порту. Це повідомлення заноситься у чергу відповідного порту, яка може містити обмежену кількість запитів, залежно від того, як конфігуровано систему. Перед доступом до UART-порту задача отримує mutex, тим самим гарантується, що лише одна задача в системі має право працювати із фізичним інтерфейсом у конкретний момент. Такий механізм унеможливорює конфлікти доступу під час паралельного надходження запитів від різних TCP-клієнтів.

Важливою частиною алгоритму є механізм планування, який реалізується на рівні UART-задачі. Після надсилання кадру у RTU-мережу задача переходить у режим очікування відповіді з використанням апаратного DMA-прийому або interrupt-механізму. У випадку успішного прийому відповідь переноситься у внутрішній буфер, перевіряється CRC і надалі передається у TCP-сесію, яка пов'язана з оригінальним запитом. Якщо ж відповідь не надходить протягом

визначеного тайм-ауту, застосовується retry-логіка або залежно від налаштувань задачі формує виключення і повертає відповідь із помилкою TCP-клієнту. При цьому обробка наступних запитів не блокується, оскільки система продовжує працювати зі структурою черг [25].

Пріоритезація задач також охоплює спеціальні службові потоки, серед яких важливе місце займає задача watchdog. Цей компонент контролює стан усіх основних задач, відстежуючи випадки зависання, аномально довгого блокування або низького рівня активності. Такий механізм дозволяє виявляти й автоматично коригувати несправності, що є критично важливим у промислових системах, де тривалі простої неприпустимі. Додаткові задачі, такі як AT-сервер або echo-сервер, працюють на нижчих пріоритетах і не впливають на критичні процеси обміну, але забезпечують важливу функціональність — конфігурацію та діагностику роботи шлюзу.

Загалом алгоритмічна модель обробки запитів у шлюзі побудована на поєднанні пріоритезації задач, асинхронної взаємодії між TCP і RTU, захищених черг повідомлень та механізмів DMA/interrupt, які дозволяють реалізувати обробку у реальному часі без значного навантаження на процесор. Така архітектура гарантує стабільну роботу системи навіть у сценаріях із великою кількістю одночасних клієнтів, складними маршрутами доступу та високою частотою опитування RTU-пристроїв.

2.5. Механізм конфігурації через AT-команди по TCP

Механізм конфігурації шлюзу Modbus TCP ↔ Modbus RTU є одним з ключових компонентів системи, оскільки саме він визначає гнучкість, масштабованість та зручність інтеграції пристрою в різноманітні промислові топології. Замість статичної конфігурації або складних веб-інтерфейсів у досліджуваному шлюзі використано механізм AT-команд, адаптований для роботи через TCP-з'єднання. Такий підхід поєднує компактність текстового

протоколу, низькі вимоги до ресурсів і сумісність із широким спектром клієнтських систем — від SCADA до власних утиліт конфігурації.

Механізм АТ-команд реалізований як окрема задача FreeRTOS, яка функціонує незалежно від основних задач маршрутизації та обробки Modbus. Це дозволяє уникнути блокувань або впливу конфігураційних операцій на роботу основних комунікаційних каналів. АТ-сервер працює поверх lwIP TCP-стека і використовує окремий порт, що дозволяє клієнтам встановлювати з'єднання без втручання у роботу інших служб. При підключенні клієнта сервер очікує текстові рядки, які закінчуються символом `\n`, і обробляє їх відповідно до синтаксичної структури АТ-команд.

Синтаксис команд у системі стандартизовано та містить три основні режими: запит значення параметра (АТ?ПАРАМЕТР), зміна параметра (АТ+ПАРАМЕТР=значення), та виконання дії (АТ~ПАРАМЕТР). Такий формат є інтуїтивно зрозумілим і дозволяє користувачеві легко керувати широким набором налаштувань, включно з параметрами мережевого інтерфейсу, конфігурацією UART, призначенням TCP-портів, налаштуваннями маршрутизації та системними параметрами. Завдяки текстовому формату команда одночасно виконує функцію документації, оскільки відображає логічну структуру параметрів.

Особливу роль відіграє блок налаштувань, який визначає спосіб взаємодії між TCP-сесіями та RTU-портами. Параметри містять режим роботи (Server/Client), дозволені Modbus-адреси, відповідний номер серійного порту, номер TCP-порту для підключень та тайм-аут неактивності. На основі цих параметрів реалізується повноцінна модель маршрутизації TCP → RTU, при цьому один порт може обслуговувати кілька Modbus-адрес, що забезпечує значно більшу гнучкість у порівнянні з комерційними шлюзами. Кількість допустимих портів визначається лише обсягом доступної пам'яті, що дозволяє масштабувати систему на етапі компіляції.

Уся конфігурація зберігається у Flash-пам'яті мікроконтролера, що забезпечує збереження параметрів під час перезавантаження. Для цього

використовується спеціальна команда AT~SAVE, яка ініціює запис поточних налаштувань. Команда AT~REBOOT дозволяє застосувати зміни, які вимагають повторної ініціалізації мережевого або послідовного інтерфейсу. Окремо реалізована команда AT~RESTORE, що виконує скидання параметрів до заводських значень. Такий підхід дозволяє гнучко керувати конфігурацією пристрою без потреби в апаратному доступі.

Функціонування AT-сервера також інтегровано з системою діагностики, де користувач може отримати інформацію про RTOS-завдання, продуктивність, інформацію про Flash, MAC-адресу, параметри мережі й інші службові дані. Це дозволяє оперативно відстежувати стан пристрою, виявляти можливі несправності та здійснювати технічне обслуговування. Такий підхід значно підвищує зручність роботи зі шлюзом та забезпечує його придатність для використання у різних індустріальних сценаріях.

Таким чином, механізм конфігурації через AT-команди у запропонованому шлюзі представляє собою універсальний, легкий та розширюваний протокол керування. Він не потребує складних графічних інтерфейсів, працює у будь-якому мережевому середовищі та дозволяє гнучко керувати всіма аспектами роботи шлюзу — від базових параметрів мережі до багаторівневої маршрутизації Modbus-запитів. Завдяки використанню окремої задачі FreeRTOS, протокол AT не впливає на продуктивність основної логіки обміну Modbus, а інтеграція зі службовими механізмами контролю забезпечує надійність і прогнозованість роботи системи.

2.6. Проблеми сумісності та рішення

Проблеми сумісності при інтеграції мереж Modbus TCP і Modbus RTU є одним із найважливіших аспектів, що визначає надійність і універсальність програмно-апаратного шлюзу. У реальних промислових системах доводиться працювати з обладнанням різного покоління, різних виробників, з різними реалізаціями протоколу Modbus та різними фізичними параметрами каналу.

Кожен з цих факторів може створювати бар'єри для коректної взаємодії, і тому завдання шлюзу полягає не лише в простому перетворенні форматів даних, а й у компенсації недоліків, відмінностей і неточностей у реалізаціях протоколів.

Однією з найпоширеніших проблем є різні параметри серійного інтерфейсу, зокрема швидкість UART, довжина кадру, кількість стоп-бітів та паритет. У промислових системах можуть одночасно працювати пристрої зі швидкостями від 1200 до 115200 бод, і шлюз повинен забезпечувати їх підтримку без порушення загальної логіки обміну. У досліджуваній системі ця проблема вирішена завдяки можливості конфігурування UART через AT-команди, що дозволяє адаптувати роботу послідовного порту під конкретний пристрій. Застосування апаратного DMA та interrupt-моделі гарантує коректність обробки байтів навіть на високих швидкостях, а мультиплатформений модуль HAL забезпечує можливість адаптації під різні середовища (Linux/Windows) без зміни логіки роботи верхнього рівня.

Наступною проблемою є різні реалізації Unit ID у протоколі Modbus TCP. І хоча специфікація визначає його як адресу підлеглого пристрою у разі роботи через шлюз, на практиці деякі програми і пристрої використовують Unit ID як службове поле або завжди передають його рівним 0xFF. Такі ситуації можуть призвести до некоректного визначення адреси RTU-пристрою або конфлікту між TCP-сесіями. Завдяки гнучкій структурі параметрів, шлюз може зіставляти запитувану адресу із дозволеними у конфігурації, а додаткові механізми маршрутизації дають змогу уникнути некоректних запитів або перенаправити їх відповідно до правил трансляції.

Проблеми виникають також через наявність у RTU-мережі пристроїв, які мають нестандартну поведінку або не дотримуються вимог протоколу щодо інтервалів між байтами. Такі пристрої можуть генерувати кадри з неточними паузами, що ускладнює визначення меж кадру. У досліджуваній системі ця проблема компенсується через використання апаратних таймерів STM та DMA-режиму приймання, який дозволяє уникати помилок, пов'язаних з нерівномірністю потоку байтів. Таймер завершує прийом у випадку виявлення

паузи, а алгоритм перевірки CRC гарантує, що прийнятий кадр дійсно належить коректному пристрою.

Ще одна група проблем пов'язана з різними моделями доступу до шини. У RTU-мережі зазвичай усі пристрої під'єднані до однієї фізичної лінії, і будь-яка спроба одночасної передачі призводить до колізій. Оскільки шлюз не змінює модель master/slave і не виконує роль незалежного майстра, він повинен правильно синхронізувати доступ TCP-клієнтів до спільної RTU-шини. У досліджуваній системі це питання вирішене через mutex-захист, що забезпечує монопольний доступ задачі UART до передавання у конкретний момент часу. Механізм черг FreeRTOS дозволяє контролювати потік запитів таким чином, щоб жодна TCP-сесія не могла зруйнувати обмін, надсилаючи запити у невідповідний момент або із недотриманням тайм-аутів [31].

Проблеми сумісності можуть також торкатися TCP-стека, особливо в умовах, коли клієнтські програми використовують некоректні або нестандартні реалізації Modbus TCP. Наприклад, можуть надходити неправильно сформовані МВАР-заголовки, неузгоджені поля довжини або запити з порушеною структурою кадру. У шлюзі реалізовано механізм попередньої валідації TCP-пакетів, що дозволяє відфільтровувати помилкові запити, а також автоматичну реконфігурацію сесій у випадку некоректної поведінки клієнта. Такий підхід підвищує надійність системи та захищає RTU-пристрої від некоректних команд.

Таким чином, проблеми сумісності у шлюзі Modbus TCP ↔ Modbus RTU виникають у широкому спектрі: від різних параметрів фізичного рівня та нестандартної поведінки пристроїв до суттєвих відмінностей у часах реакції та структурі кадрів. Розроблена архітектура забезпечує комплексне вирішення цих проблем за рахунок гнучких конфігураційних механізмів, захисту доступу до шини, апаратної стійкості до перешкод та інтегрованих алгоритмів перевірки коректності кадрів. Завдяки цьому шлюз здатний працювати з різними пристроями, забезпечуючи стабільний та передбачуваний обмін у реальних промислових умовах.

2.7. Підсумки теоретичних досліджень

Проведені у цьому розділі теоретичні дослідження дозволили сформуванати цілісну модель побудови програмно-апаратного шлюзу для інтеграції мереж Modbus TCP і Modbus RTU, а також визначити ключові принципи його функціонування. На основі аналізу внутрішньої логіки протоколів, часових параметрів, особливостей роботи фізичного та транспортного рівнів було сформовано архітектуру, яка забезпечує коректну трансляцію кадрів, узгодження протоколів та синхронізацію паралельних потоків даних.

Розгляд загальної архітектури шлюзу показав, що поєднання FreeRTOS, lwIP та DMA/interrupt-механізмів UART створює ефективне підґрунтя для забезпечення мультипоточності, стабільності й детермінованості виконання задач у режимі реального часу. Виділення окремих задач для TCP-підключень, UART-інтерфейсів, АТ-сервера та службових модулів дозволяє збалансувати навантаження між різними частинами системи та забезпечує масштабованість при збільшенні кількості клієнтів або портів.

Важливе значення має аналіз внутрішньої логіки конвертації між форматами ADU та PDU, що визначає коректність роботи шлюзу у різних мережесхемних сценаріях. Було показано, що модуль конвертації має враховувати специфіку обох протоколів, зокрема структуру МВАР-заголовка, формат RTU-кадрів і правила формування CRC. Окремо розглянуто механізм контекстного зв'язку між TCP- і RTU-транзакціями, що забезпечує правильне прив'язування відповідей до відповідних клієнтів.

Детальний аналіз моделей маршрутизації продемонстрував, що використання конфігураційних параметрів забезпечує гнучке та динамічне відображення TCP-клієнтів на фізичні та логічні адреси RTU-пристроїв. Важливо, що один TCP-клієнт може обслуговувати кілька Modbus-вузлів, а максимальна кількість таких вузлів визначається доступною пам'яттю та може

масштабуватися під час білду проєкту. Це виділяє систему серед типових комерційних шлюзів, які часто мають жорстко обмежені таблиці маршрутизації.

У контексті багатозадачності було розглянуто алгоритми планувальника, блокування доступу до UART через mutex, роботу черг повідомлень, а також механізми пріоритезації задач. Було показано, що правильне виділення пріоритетів — зокрема підвищення пріоритету UART-задач — дозволяє уникнути порушення таймінгів у RTU-протоколі та забезпечити стабільність передачі навіть у умовах інтенсивного TCP-трафіку.

Розгляд механізму конфігурації через AT-команди продемонстрував, що застосування текстового протоколу налаштування поверх TCP дозволяє забезпечити гнучкість, простоту керування та дистанційний доступ до всіх параметрів шлюзу без складних інтерфейсів. Розділення AT-сервера в окрему задачу FreeRTOS зберігає стабільність основних процесів обміну даними.

Також у межах теоретичного аналізу були детально розглянуті проблеми сумісності. Було показано, що відмінності в швидкостях UART, реалізаціях RTU, поведінці TCP-клієнтів або різних таймінгах можуть призводити до помилок, і саме тому архітектура шлюзу містить адаптивні механізми маршрутизації, таймінгові компенсатори, CRC-контроль, апаратні засоби захисту RS-485 та алгоритми автоматичного відновлення.

Таким чином, теоретична частина дослідження підтверджує обґрунтованість вибраної архітектури, правильність її структурних рішень та здатність системи забезпечувати надійну інтеграцію мереж Modbus TCP та Modbus RTU. Результати цього розділу є фундаментом для переходу до практичної частини, де буде детально описано реалізацію шлюзу, експериментальні дослідження його роботи та аналіз отриманих результатів.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ШЛЮЗУ MODBUS

3.1. Опис реалізованого пристрою

Реалізований у межах даної роботи програмно-апаратний шлюз Modbus TCP ↔ Modbus RTU представляє собою компактний високопродуктивний пристрій, призначений для інтеграції Ethernet-мереж та послідовних ліній RS-485 у промислових системах керування. Апаратна платформа разом із програмною архітектурою формує комплексне рішення, що поєднує низьке енергоспоживання, стійкість до завад, масштабованість і підтримку гнучкої маршрутизації протоколу Modbus.

Основою апаратної частини шлюзу є мікроконтролер STM32F407ZGT6 — представник 32-бітного сімейства ARM Cortex-M4 [27], який вирізняється високою продуктивністю, наявністю апаратного плаваючого множника (FPU) та широким набором периферійних модулів. Обсяг вбудованої Flash-пам'яті становить 1 МБ, що дозволяє розміщувати як мережевий стек, так і програмний модуль конвертації без необхідності зовнішніх накопичувачів. Значний розмір оперативної пам'яті забезпечує можливість одночасної обробки кількох TCP-сесій, мережевих буферів та внутрішніх структур FreeRTOS.

Для роботи з Ethernet у пристрої використовується зовнішній РНУ-трансивер DP83848, який забезпечує підтримку інтерфейсу 10/100 Мбіт/с та відповідає вимогам промислових застосувань щодо стійкості до шумів і температурних змін. РНУ підключений до мікроконтролера за допомогою інтерфейсу RMIІ, що дозволяє зменшити кількість ліній та забезпечити стабільність передачі. Таке апаратне рішення робить можливим використання повного стеку ІwІР без зниження продуктивності та дозволяє обслуговувати до шістнадцяти одночасних клієнтських з'єднань. На рис. 3.1 зображено апаратну платформу шлюзу.

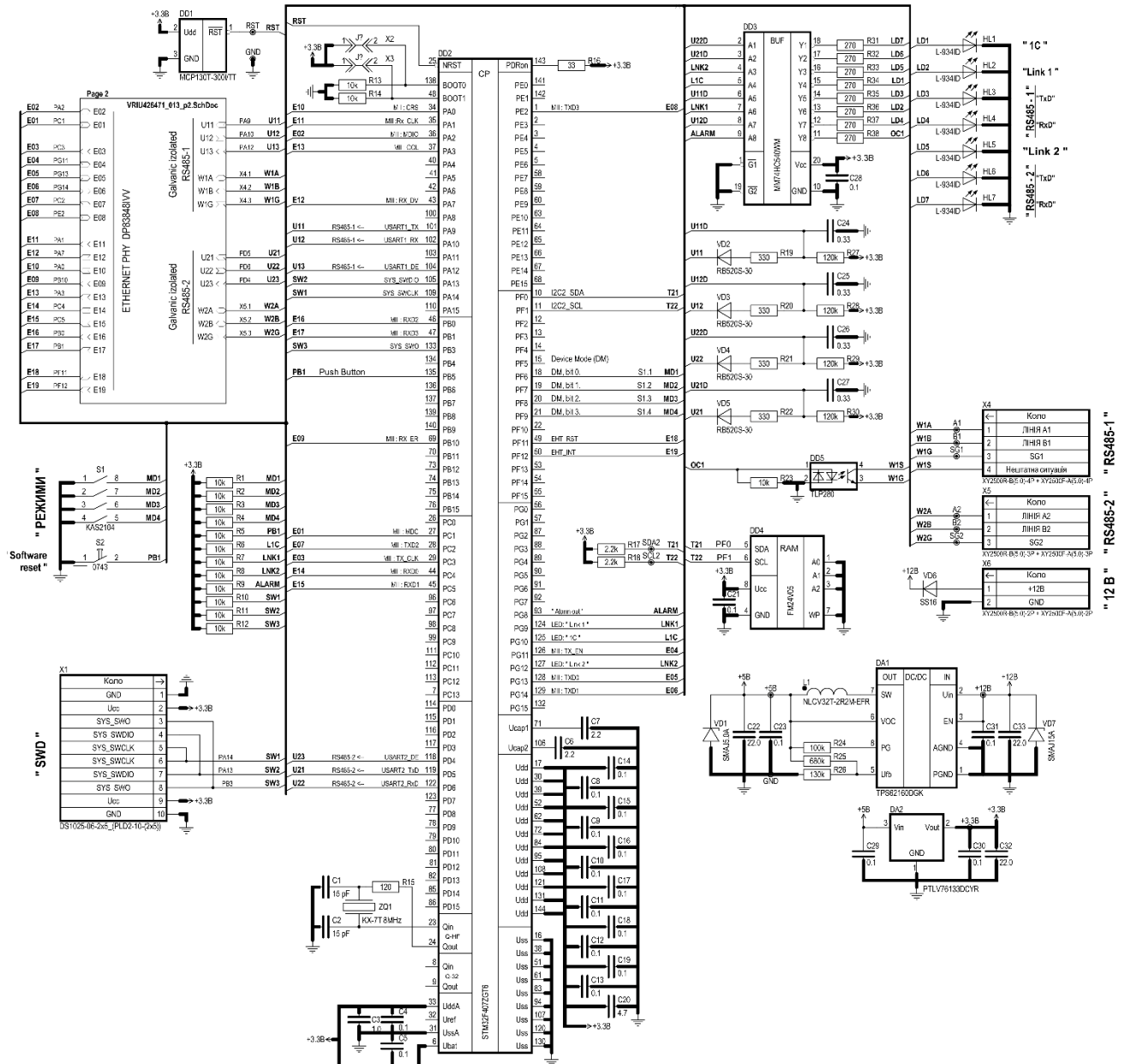


Рис. 3.1. Апаратна платформа шлюзу

Однією з ключових особливостей пристрою є наявність двох інтерфейсів RS-485, які реалізовано на базі промислових трансиверів зі вбудованими захисними схемами. Гальванічна розв'язка забезпечує відсутність паразитних струмів між системами, захист від різниці потенціалів, а також дозволяє підключати пристрій до довгих ліній у промислових приміщеннях. Обидва RS-485-інтерфейси працюють під керуванням UART-модулів мікроконтролера у режимах DMA та interrupt, що дозволяє зменшити навантаження на процесор та забезпечити точність часових параметрів RTU-протоколу.

Одним із важливих конструктивних рішень є модульна організація HAL-рівня, що дозволяє відокремити апаратну логіку UART, Ethernet та GPIO від програмної частини, відповідальної за конвертацію протоколів. Такий підхід дозволяє портовати програмну частину шлюзу у середовище Linux або Windows, зберігаючи структуру задач, алгоритми маршрутизації та механізми TCP/RTU-конвертації. Фактично, нижній рівень пристрою є універсальним апаратно-програмним модулем, який легко адаптується до різних платформ.

Апаратна архітектура шлюзу розроблена з урахуванням можливості масштабування кількості UART-портів. На рівні програмного забезпечення система підтримує динамічну конфігурацію кількості послідовних інтерфейсів залежно від параметрів, визначених під час збирання проєкту. Це дозволяє адаптувати шлюз до різних задач: від компактних двопортових рішень до багатопортових систем для розширених промислових застосувань. Кількість підтримуваних TCP-підключень, таблиці маршрутизації та буфери також масштабуються відповідно до доступної оперативної пам'яті.

У конструктивній частині пристрою передбачені також елементи для захисту живлення та сигналів, включаючи фільтри, стабілізатори та ESD-захист. Живлення системи здійснюється через промисловий DC-вхід з діапазоном, відповідним вимогам до обладнання, яке може працювати поблизу електричних шаф та силових ліній. Продумане розміщення компонентів на друкованій платі мінімізує перешкоди між Ethernet та RS-485, а гальванічна розв'язка запобігає передаванню збурень від польового рівня до цифрової частини.

Загалом реалізований пристрій є повноцінною апаратною платформою, що поєднує потужний мікроконтролер, два незалежні промислові інтерфейси RS-485, сучасний Ethernet-трансивер та спеціалізовані електричні захисні рішення. У поєднанні з багатозадачним програмним забезпеченням він формує основу для надійної, стабільної та масштабованої інтеграції мереж Modbus TCP та Modbus RTU.

3.2. Схема системи та апаратні рішення

Апаратна частина програмно-апаратного шлюзу є ключовим елементом його функціональної моделі, адже від коректності схемотехнічних рішень залежить стабільність обміну даними, надійність роботи в умовах промислових перешкод і здатність забезпечувати гарантовані часові параметри протоколу Modbus RTU. Схема системи розроблена з урахуванням вимог до промислових комунікаційних пристроїв, а також із акцентом на гнучкість конфігурації, можливість масштабування та забезпечення електричної безпеки.

Центральним елементом схеми є мікроконтролер STM32F407ZGT6, який виконує роль обчислювального ядра і координатора всіх периферійних модулів. Його функціонування забезпечується стабільним тактуванням від кварцового резонатора, який визначає частоту роботи ядра та периферії. Додатково передбачено джерело резервного тактування для RTC-модуля, що дозволяє уникнути порушення внутрішніх таймерів у випадку короткочасних збоїв живлення. Ретельно спроектована система живлення містить стабілізатори напруги для цифрової частини та окремі гілки живлення для інтерфейсів, чутливих до шумів.

Ethernet-підсистема реалізована на основі зовнішнього PHY DP83848, який з'єднаний із мікроконтролером за допомогою інтерфейсу RMII. У схемі передбачено всі необхідні компоненти узгодження сигналів — резистори термінування, трансформаторна розв'язка Ethernet-лінії та RC-фільтри живлення PHY-чипа. Таке рішення забезпечує коректну роботу навіть при великих довжинах мережевих кабелів та дозволяє досягати стабільної швидкості 100 Мбіт/с. На схемі також передбачено індикатори стану Ethernet — світлодіоди активності та зв'язку, інтегровані або у роз'єм RJ-45 залежно від конкретного виконання плати. На рис. 3.2 зображено схему підключення DP83848.

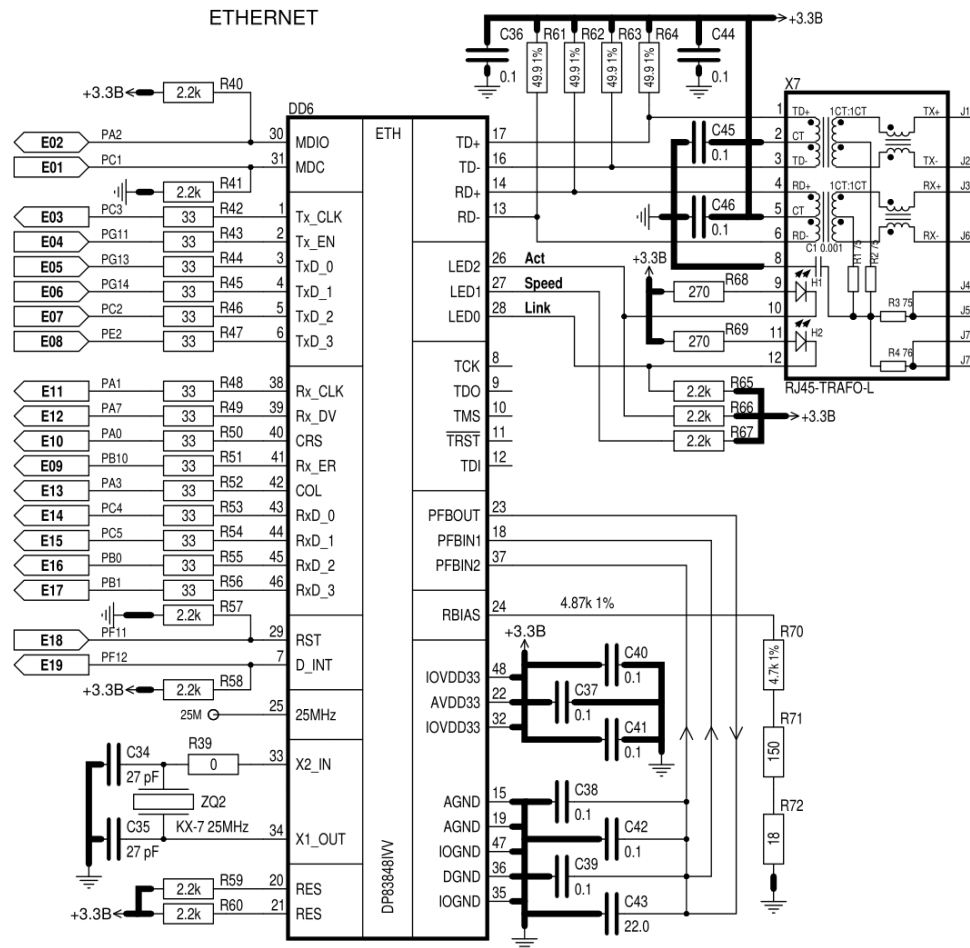


Рис. 3.2. Схема підключення DP83848

Для підключення до польових пристроїв у схемі використано два незалежні послідовні інтерфейси RS-485, кожен з яких має власний трансивер з ізоляцією. Гальванічна розв'язка реалізована із застосуванням спеціалізованих цифрових ізоляторів, що дозволяє повністю відокремити високовольтне польове середовище від цифрової частини пристрою. Трансивери RS-485 живляться від окремих ізольованих DC/DC-перетворювачів, які формують стабільовану напругу та додатково захищають пристрій від завад. Наявність таких рішень дозволяє гарантувати коректний обмін даними навіть при значних електромагнітних впливах та різниці потенціалів між сегментами мережі.

У складі схеми також реалізовано апаратний контроль за напрямком передачі на RS-485, що забезпечує автоматичне перемикавання між режимами прийому та передачі без суттєвих затримок. Завдяки можливості керування

лінією DE/RE мікроконтролер може точно синхронізувати момент початку передачі Modbus RTU-кадру, що є надзвичайно важливим для забезпечення правильного формування міжкадрових інтервалів. На рис. 3.3 зображено схему підключення RS-485.

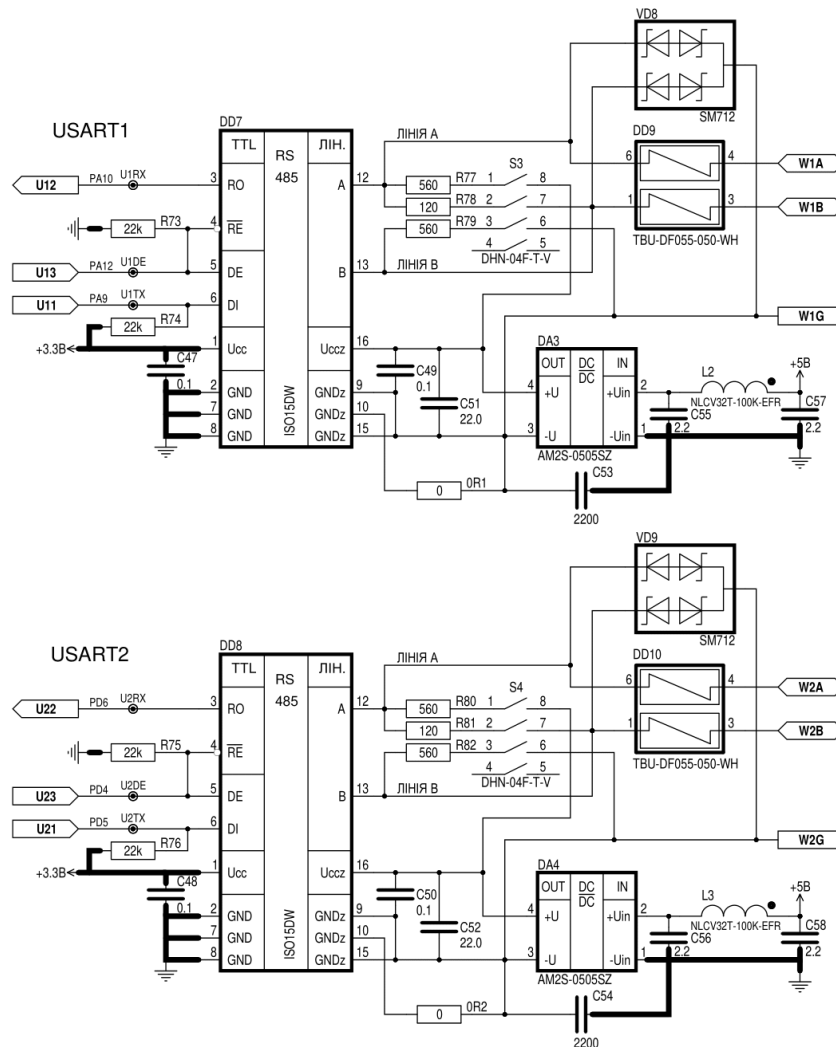


Рис. 3.3. Схема підключення RS-485

Пристрій містить також допоміжні ланцюги, зокрема схеми захисту від електростатичного розряду (ESD) на Ethernet-лініях і ланцюгах UART/RS-485. Додані діодні захисні елементи та TVS-діоди допомагають запобігти пошкодженню мікросхем через імпульсні перенапруги, що є поширеним явищем у промислових умовах.

Окрему увагу під час проєктування було приділено трасуванню друкованої плати. Лінії RМІІ між мікроконтролером та РНУ мінімізованої довжини та узгоджені за імпедансом, щоб уникнути спотворень сигналу. Аналогічним чином здійснено трасування ліній RS-485, де особливо важливою є симетрія диференційної пари. Розв'язка цифрової землі та землі інтерфейсів RS-485 виконана таким чином, щоб запобігти виникненню паразитних струмів та коливань потенціалу.

На схемі також присутні модулі індикації стану системи, серед яких світлодіод живлення, індикатори активності UART, індикатори роботи Ethernet, та службові лінії Reset, Boot і SWD. Вони забезпечують можливість відлагодження пристрою та спрощують технічне обслуговування.

Таким чином, реалізоване схемотехнічне рішення поєднує в собі високу стійкість до зовнішніх факторів, модульність, гнучкість конфігурації та відповідність вимогам до роботи в промислових умовах. Воно створює надійну основу для функціонування всієї системи та гарантує стабільний обмін даними як через Ethernet, так і по RS-485.

3.3. Архітектура програмного забезпечення

Програмне забезпечення реалізованого шлюзу Modbus TCP ↔ Modbus RTU побудоване за модульним принципом і ґрунтується на багатозадачній моделі, яку забезпечує операційна система реального часу FreeRTOS. Архітектура ПЗ розроблена таким чином, щоб забезпечити одночасне опрацювання декількох незалежних потоків даних, гарантувати узгодження TCP- та RTU-протоколів, а також забезпечити надійність і стійкість системи в режимі тривалої безперервної роботи.

Структурно програмне забезпечення складається з декількох логічних рівнів: апаратно-абстрактного рівня (HAL), транспортного рівня, рівня логіки протоколів, рівня маршрутизації, конфігураційного рівня та службових модулів.

HAL-рівень [26] відповідає за взаємодію з фізичними інтерфейсами — UART, Ethernet, GPIO, таймерами та механізмами DMA. Він реалізований таким чином, щоб не бути прив'язаним до конкретної апаратної платформи. Це дає змогу переносити більшість кодової бази на інші середовища (Linux або Windows) без зміни логіки роботи протоколів.

Над HAL-рівнем розташовані модулі транспортного рівня, зокрема стек lwIP, який забезпечує обробку Ethernet-кадрів, TCP/UDP-сесій та IP-маршрутизації. Стек працює у режимі безопераційного ядра (no-OS), але інтегрований із FreeRTOS завдяки системі callback-функцій, таймерів і задач, які відповідають за обробку пакетів. Із lwIP взаємодіють задачі TCP-серверів, кожна з яких обслуговує окремий встановлений порт або набір портів.

На рівні логіки протоколів реалізовано модулі Modbus TCP і Modbus RTU, які відповідають за виділення PDU, перевірку коректності MBAP-заголовків, формування RTU-кадрів, перевірку CRC та перетворення ADU відповідно до правил, зазначених у специфікації Modbus. Пакет обробки Modbus реалізує уніфіковану структуру запитів і відповідей, що дозволяє спростити маршрутизацію та мінімізувати дублювання коду.

Однією з найважливіших частин архітектури є рівень маршрутизації, який здійснює зіставлення TCP-клієнтів із фізичними портами RS-485 та відповідними Modbus-адресами. Керування цим процесом виконується відповідно до конфігураційних параметрів, що зберігаються у Flash-пам'яті. Завдяки такому підходу маршрутизація стає гнучким механізмом, який може бути адаптований для будь-якого числа Modbus-пристроїв, обмеженого лише обсягом доступної оперативної пам'яті.

Кожен TCP-порт обслуговується окремою задачею FreeRTOS. Задача відповідає за прийом пакетів, виділення PDU, передачу їх у чергу маршрутизатора, очікування відповіді та відправлення даних клієнту. Аналогічним чином кожен UART-порт обслуговується власною задачею, яка здійснює запис даних у DMA-буфер, перемикає напрямку RS-485, роботу з

інтервалами RTU-протоколу та приймання відповіді. Важливо, що кожна UART-задача має підвищений пріоритет відносно TCP-задач, оскільки Modbus RTU є протоколом із жорсткими часовими обмеженнями.

Зв'язок між TCP-задачами та UART-задачами реалізований через механізм черг FreeRTOS. Черги дозволяють організувати буферизацію запитів, уникнути втрати даних та забезпечити впорядкованість обробки. Для синхронізації доступу до UART використовується mutex, що гарантує монопольний доступ у момент обробки кадру, запобігаючи передчасному або паралельному надсиланню даних.

Окремою задачею реалізовано AT-сервер, який відповідає за прийом і обробку команд конфігурації через TCP-з'єднання. Інтерфейс AT-команд виконується у власному контексті, не впливаючи на маршрутизацію Modbus. Він має доступ до системи збереження параметрів у Flash, а також містить функції читання діагностичної інформації про RTOS, стекові ресурси, стан задач і конфігурацію інтерфейсів.

Додаткові службові задачі включають watchdog-задачу, яка здійснює контроль за життєвим циклом інших задач, визначаючи можливі зависання або порушення у роботі. Echoserver, що може бути активований у налаштуваннях, використовується для діагностики мережевої взаємодії. Окрема задача зберігає системні лічильники, реалізує логику системного моніторингу та слугує опорною точкою для планувальника.

Побудова ПЗ за таким модульним принципом дозволяє забезпечити надійну взаємодію між окремими компонентами системи, а також підвищує стійкість до помилок та відмов. Завдяки багатозадачності FreeRTOS, чіткій структурі черг і mutex-механізмам, а також чітко визначеним інтерфейсам між модулями, система здатна працювати у багатоканальному режимі, обслуговуючи широкий спектр клієнтів і RTU-пристроїв. На рис. 3.4 зображено загальну блок-схему роботи пристрою.

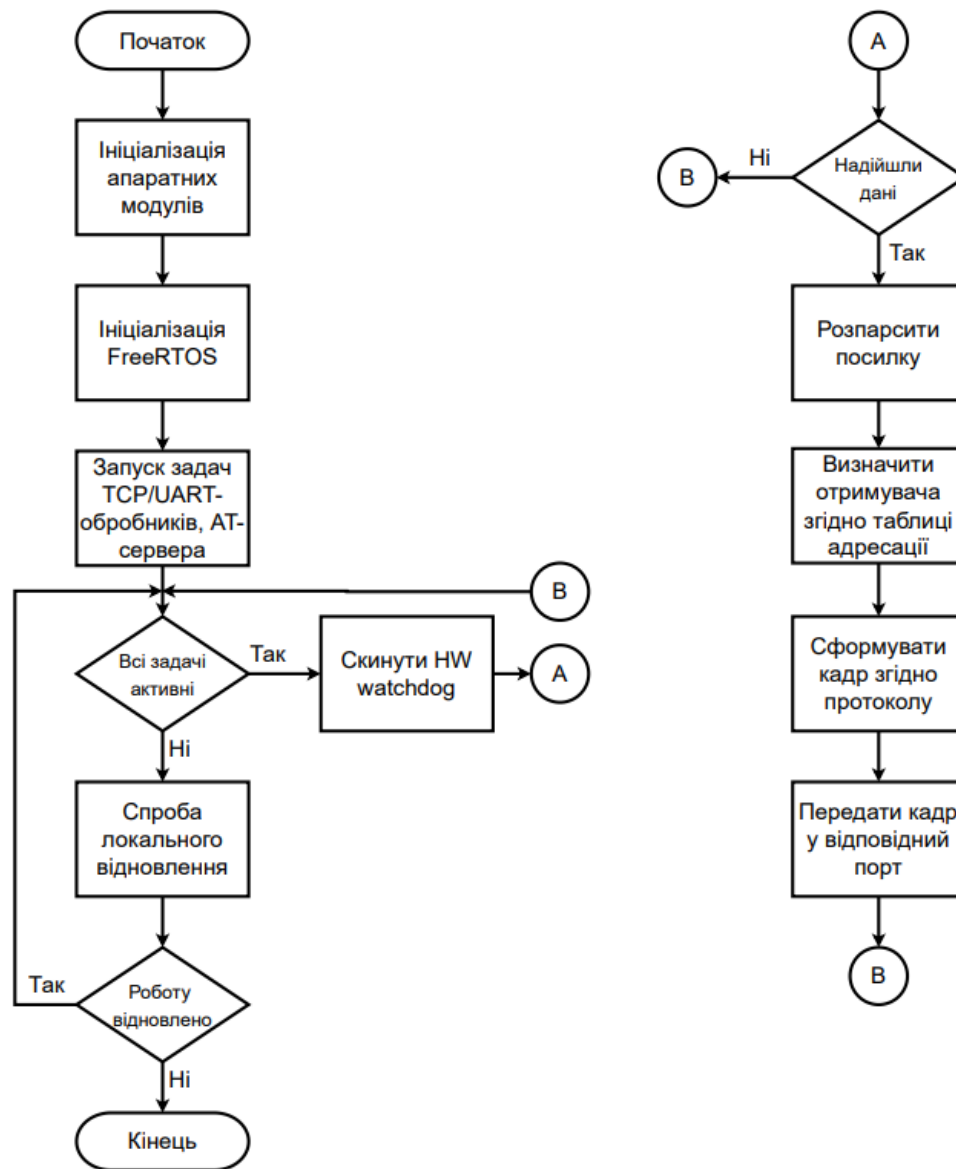


Рис. 3.4. Загальна блок-схема

Блок-схема відображає загальний робочий цикл шлюзу Modbus TCP ↔ Modbus RTU. Вона починається з ініціалізації апаратних модулів мікроконтролера та запуску середовища FreeRTOS, після чого створюються й активуються усі функціональні задачі: обробники TCP-підключень, UART-процеси та AT-сервер конфігурації. Основна системна задача постійно контролює їхню активність за допомогою механізму «живих маркерів», і в разі підтвердження нормальної роботи скидає апаратний watchdog для підтримання стабільності системи.

У разі виявлення неактивної задачі система здійснює локальну спробу її відновлення; якщо відновлення неможливе, пристрій завершує роботу та переходить у режим перезапуску через watchdog. Паралельно в окремому сегменті блок-схеми відображено алгоритм обробки вхідних даних: кожен отриманий Modbus-кадр проходить парсинг, маршрутизацію відповідно до таблиці адрес, перетворення у відповідний формат протоколу та передачу у відповідний порт TCP або RS-485.

Таким чином, схема демонструє поєднання двох ключових механізмів — циклічного контролю працездатності задач та процедури конвертації і маршрутизації Modbus-кадрів — що забезпечує надійну, відмовостійку та передбачувану роботу шлюзу у промислових умовах.

3.4. Реалізація перетворення Modbus TCP ↔ Modbus RTU

Реалізація механізму перетворення між протоколами Modbus TCP і Modbus RTU є центральним елементом роботи програмно-апаратного шлюзу. Саме цей модуль забезпечує коректну взаємодію між мережами Ethernet та RS-485 та гарантує правильність структури кадрів на обох напрямках комунікації. Побудова системи перетворення ґрунтується на розділенні логіки TCP, RTU та маршрутизації, що дозволяє підтримувати мультипоточність і забезпечувати обробку великої кількості одночасних транзакцій.

Одним із ключових принципів реалізації є розмежування рівнів ADU (Application Data Unit) та PDU (Protocol Data Unit). У протоколі Modbus TCP ADU містить MBAP-заголовок, що включає ідентифікатор транзакції, довжину повідомлення та поле Unit ID. У Modbus RTU ADU складається з PDU та поля CRC, яке є обов'язковим для перевірки цілісності даних. Оскільки PDU залишається однаковим у обох випадках, реалізація перетворення полягає у заміні транспортної оболонки, збереженні PDU та відновленні коректних контрольних полів.

При надходженні запиту від TCP-клієнта задача TCP-сервера виділяє PDU, перевіряє довжину пакету та коректність МВАР-заголовка, а також зчитує Unit ID, який у системі використовується для визначення RTU-адреси. Подальший етап включає передавання PDU у чергу маршрутизатора разом із метаданими: номером TCP-сесії, номером порту та очікуваним часом відповіді. Модуль маршрутизації на основі таблиці визначає, який RS-485-порт має обробити запит, а також формує RTU-кадр, додаючи CRC, розраховану відповідно до стандартного алгоритму Modbus.

Передавання кадру на RTU-шину виконує UART-задача відповідного порту. Вона отримує PDU, перетворює його в ADU для Modbus RTU, перемикає трансивер у режим передавання, ініціює DMA-транзакцію та забезпечує дотримання міжкадрових інтервалів, визначених специфікацією (переважно 3.5 символи тиші). Після завершення передачі UART-задача перемикає лінію RS-485 у режим прийому і розпочинає приймання відповіді, використовуючи апаратні таймери для визначення завершення кадру за міжбайтовими паузами. Приймання завершується при настанні паузи або заповнення DMA-буфера.

По завершенні прийому кадру виконується перевірка CRC. Якщо відповідь є коректною, вона надходить до модуля маршрутизації, який зіставляє її з TCP-клієнтом, що ініціював запит. Далі формується ADU Modbus TCP: додається МВАР-заголовок із початковим ідентифікатором транзакції, і кадр передається у задачу відповідного TCP-порту. Це дозволяє уникнути змішування відповідей та гарантує коректне зіставлення у випадку одночасної роботи кількох TCP-клієнтів.

Особливу роль в архітектурі відіграє механізм буферизації та черг. Кожен TCP-запит передається в маршрутизатор через окрему повідомлення-чертгу, що дозволяє уникати втрати даних та забезпечує впорядкованість обробки. Обробка виконується за принципом FIFO, що відповідає природі протоколу Modbus і гарантує цілісність транзакцій у випадку великої кількості клієнтів. Для синхронізації доступу до UART використовується mutex, який гарантує

монопольний доступ до фізичної шини RS-485, що є критично важливим для протоколу RTU, який не допускає паралельного передавання.

У напрямку RTU → TCP логіка відповідає зворотному перетворенню: після коректного приймання RTU-кадру система виділяє PDU, формує MBAP-заголовок і повертає дані клієнту через відповідну TCP-задачу. Важливо, що у Modbus TCP відсутній CRC, тому структура кадру визначається лише полем Length у MBAP. Система формує це поле динамічно, виходячи з довжини PDU, що забезпечує відповідність стандарту та сумісність з різними клієнтськими реалізаціями. На рис. 3.5 зображено схему перетворення даних Modbus TCP ↔ Modbus RTU.

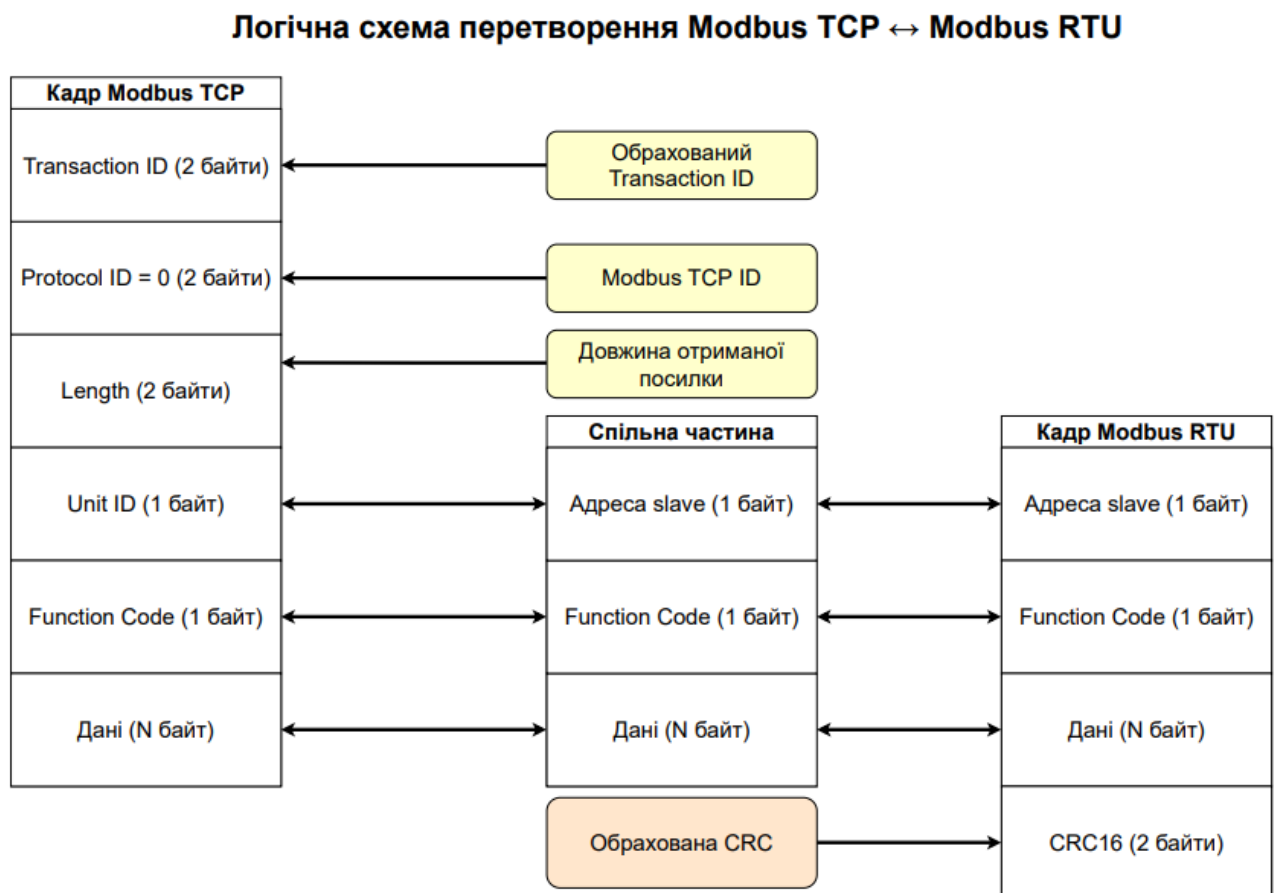


Рис. 3.5. Схема перетворення даних Modbus TCP ↔ Modbus RTU

У разі надходження некоректних TCP-пакетів, зокрема із неправильним МВАР-заголовком або некоректною довжиною, система виконує захисне відкидання пакету. Подібна валідація застосовується і до RTU-кадрів: у випадку виявлення CRC-помилки дані не потрапляють у TCP-канал, що захищає клієнта від некоректних відповідей і підвищує надійність системи.

Архітектура перетворення також враховує можливість обслуговування кількох RTU-адрес одним TCP-клієнтом. Завдяки гнучкості параметрів клієнт може взаємодіяти з багатьма RTU-пристроями через одну TCP-сесію, що значно розширює функціональність шлюзу та дає змогу уникати зайвої кількості клієнтських підключень.

Загалом структура реалізації перетворення Modbus TCP ↔ Modbus RTU є багаторівневою і комплексною, поєднуючи транспортний рівень, рівень протоколів, логіку маршрутизації та часові механізми. Результатом є система, здатна працювати у багатосесійному середовищі з високим навантаженням, забезпечуючи коректну трансляцію, дотримання тайм-аутів, підтримку промислових режимів роботи та стабільність протягом тривалого часу.

3.5. АТ-інтерфейс та конфігурація пристрою

Конфігураційний інтерфейс, реалізований у вигляді набору АТ-команд, є ключовим механізмом управління роботою шлюзу Modbus TCP ↔ Modbus RTU. На відміну від графічних або web-інтерфейсів, текстові АТ-команди забезпечують легку інтеграцію з будь-яким клієнтським програмним забезпеченням, компактність реалізації, а також мінімальні вимоги до апаратних ресурсів мікроконтролера. Інтерфейс працює поверх TCP, обробляється окремою задачею FreeRTOS і повністю ізольований від процесів маршрутизації Modbus, що гарантує надійність та передбачуваність роботи пристрою.

Комунікація з АТ-сервером здійснюється через спеціально виділений TCP-порт, який встановлюється у конфігурації пристрою. При підключенні до

цього порту користувач взаємодіє з пристроєм за допомогою текстових команд, кожна з яких завершується символом нового рядка \n. Синтаксис команд стандартизований і складається з трьох основних типів:

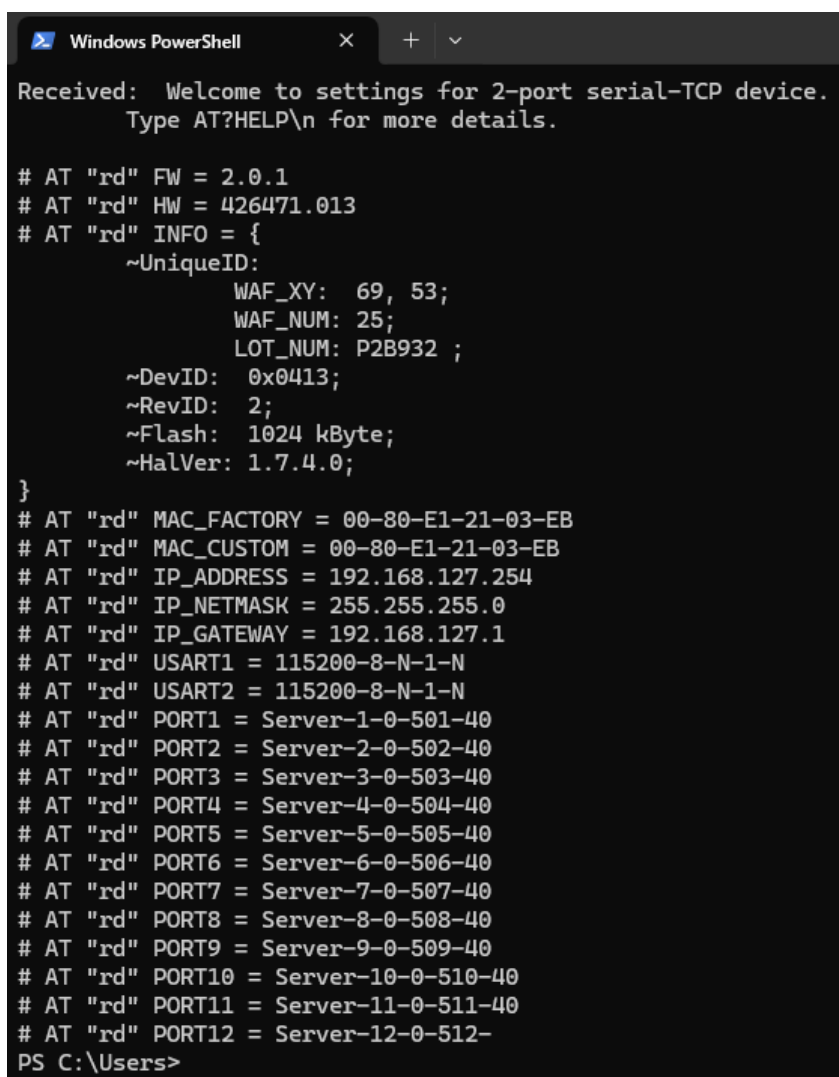
- AT? — читання параметра. Використовується для перегляду конфігураційних значень. Наприклад: AT?IP_ADDRESS\n.
- AT+ — запис або зміна параметра. Використовується для конфігурації інтерфейсів, портів, мережі. Наприклад: AT+USART1=19200-8-N-1\n.
- AT~ — виконання дії. Застосовується для ініціалізації служби, збереження параметрів або перезавантаження. Наприклад: AT~SAVE\n, AT~REBOOT\n.

Ці типи команд створюють логічно цілісну систему управління, де кожна операція має чітко визначений формат і призначення. Усі параметри згруповані таким чином, що відображають функціональну структуру шлюзу: налаштування мережі, конфігурація UART, параметри TCP-портів, системна інформація та службові функції.

Одним із ключових наборів параметрів є PORTx, який визначає правила маршрутизації протоколу Modbus. Структура параметра містить режим роботи порту (Server або Client), дозволені Modbus-адреси, номер UART-інтерфейсу, номер TCP-порту та тайм-аут неактивності. Така гнучкість дозволяє організовувати будь-які схеми маршрутизації: один TCP-клієнт може взаємодіяти з кількома RTU-адресами, різні клієнти можуть працювати через незалежні логічні порти, а керування доступом виконується на рівні конфігурації PORTx. Максимальна кількість таких портів визначається доступною оперативною пам'яттю та встановлюється на етапі збірки прошивки.

Налаштування UART також виконуються через AT-команди. Параметри передавання включають швидкість обміну, кількість біт даних, біт парності,

кількість стоп-бітів та опціональні сигнали керування потоком. Завдяки цьому інтерфейс RS-485 можна адаптувати під практично будь-який промисловий пристрій, незалежно від того, використовує він стандартні або нетипові комбінації параметрів. На рис. 3.6 зображена взаємодія з АТ-сервером через TCP-канал.



```

Windows PowerShell
Received: Welcome to settings for 2-port serial-TCP device.
Type AT?HELP\n for more details.

# AT "rd" FW = 2.0.1
# AT "rd" HW = 426471.013
# AT "rd" INFO = {
    ~UniqueID:
        WAF_XY: 69, 53;
        WAF_NUM: 25;
        LOT_NUM: P2B932 ;
    ~DevID: 0x0413;
    ~RevID: 2;
    ~Flash: 1024 kByte;
    ~HalVer: 1.7.4.0;
}
# AT "rd" MAC_FACTORY = 00-80-E1-21-03-EB
# AT "rd" MAC_CUSTOM = 00-80-E1-21-03-EB
# AT "rd" IP_ADDRESS = 192.168.127.254
# AT "rd" IP_NETMASK = 255.255.255.0
# AT "rd" IP_GATEWAY = 192.168.127.1
# AT "rd" USART1 = 115200-8-N-1-N
# AT "rd" USART2 = 115200-8-N-1-N
# AT "rd" PORT1 = Server-1-0-501-40
# AT "rd" PORT2 = Server-2-0-502-40
# AT "rd" PORT3 = Server-3-0-503-40
# AT "rd" PORT4 = Server-4-0-504-40
# AT "rd" PORT5 = Server-5-0-505-40
# AT "rd" PORT6 = Server-6-0-506-40
# AT "rd" PORT7 = Server-7-0-507-40
# AT "rd" PORT8 = Server-8-0-508-40
# AT "rd" PORT9 = Server-9-0-509-40
# AT "rd" PORT10 = Server-10-0-510-40
# AT "rd" PORT11 = Server-11-0-511-40
# AT "rd" PORT12 = Server-12-0-512-40
PS C:\Users>

```

Рис. 3.6. Взаємодія з АТ-сервером через TCP-канал

Критично важливим механізмом АТ-інтерфейсу є збереження конфігурації у Flash-пам'яті за допомогою команди АТ~SAVE. Після виконання цієї команди всі змінені параметри записуються у постійну пам'ять і використовуються після кожного перезавантаження пристрою. Окрема команда АТ~REBOOT дозволяє

застосувати зміни, які вимагають повторної ініціалізації інтерфейсів або стеку lwIP. Для повернення до заводських параметрів використовується команда AT~RESTORE, що є корисною у випадках некоректної конфігурації або помилкових налаштувань.

AT-інтерфейс підтримує також розширені команди діагностики, які дозволяють переглядати версію прошивки, апаратну ревізію, стан мережевих параметрів, список задач FreeRTOS, статистику часу виконання задач і службові дані. Це значно полегшує відлагодження пристрою, аналіз продуктивності та моніторинг роботи в реальному часі. Зокрема, команди AT?OS_LIST і AT?OS_RTS дають змогу контролювати рівень використання стеку задач, що є важливим для виявлення можливих переповнень або збоїв у системі.

У межах AT-інтерфейсу кожна команда проходить процедуру валідації: перевіряються числові параметри, формат введення, коректність структури, допустимі діапазони та внутрішня сумісність параметрів. У разі введення некоректного значення пристрій повертає повідомлення про помилку без зміни конфігурації. Це забезпечує захист від випадкових неправильних налаштувань, які можуть порушити роботу шлюзу.

Важливо підкреслити, що AT-сервер реалізований як окрема задача FreeRTOS з власним TCP-сокетом, тому його робота не впливає на маршрутизацію Modbus. Навіть при високій інтенсивності конфігураційних запитів шлюз продовжує обслуговувати TCP-клієнтів і обмінюватися даними по RTU без погіршення продуктивності.

Приклади використання AT-інтерфейсу демонструють простоту та інтуїтивність управління пристроєм. Наприклад, для зміни IP-адреси достатньо виконати команду:

- AT+IP_ADDRESS = 192.168.1.100\n
- AT~SAVE\n

- AT~REBOOT\n

Налаштування нового Modbus TCP-порту зіставленого з другим UART-інтерфейсом може виконуватися так:

- AT+PORT2 = Server-15-2-502-40\n

- AT~SAVE\n

Такий підхід дозволяє оперативно змінювати топологію мережі, додавати нові пристрої або адаптувати шлюз під інші промислові сценарії без необхідності фізичного доступу або перепрошивки.

Отже, AT-інтерфейс шлюзу є зручним, компактним та функціонально наповненим інструментом керування, який охоплює всі аспекти конфігурації пристрою — від мережевих параметрів до складних правил маршрутизації. Він забезпечує високу гнучкість, зручність експлуатації, швидкість налаштування та повну інтеграцію з внутрішньою архітектурою пристрою, що робить його важливою складовою практичної реалізації.

3.6. Налаштування маршрутизації

Маршрутизація в програмно-апаратному шлюзі Modbus TCP ↔ Modbus RTU є одним із ключових функціональних механізмів, який визначає спосіб, за допомогою якого TCP-клієнти отримують доступ до фізичних пристроїв, підключених до RS-485. На відміну від простих шлюзів, де кожен TCP-порт жорстко прив'язаний до одного UART-інтерфейсу, реалізований у межах цієї роботи шлюз має гнучку, масштабовану та динамічну модель маршрутизації, засновану на конфігураційних параметрах. Така модель дозволяє одній TCP-сесії обслуговувати декілька RTU-адрес, розподіляти клієнтів між різними фізичними інтерфейсами та керувати доступом до окремих пристроїв.

Основою маршрутизації є концепція логічних портів, кількість яких задається на етапі компіляції прошивки і визначається доступною оперативною пам'яттю. Кожен порт описується структурою параметрів, які містять режим роботи (Server/Client), список дозволених Modbus-адрес, номер UART-порту, номер TCP-порту та тайм-аут бездіяльності. Формат команд конфігурації дозволяє користувачу швидко змінювати правила маршрутизації, адаптуючи шлюз до специфічних топологій промислової мережі.

Однією з найважливіших можливостей системи є підтримка багатьох логічних адрес у межах одного TCP-порту. Це означає, що один клієнт може звертатися до багатьох RTU-пристроїв без необхідності відкриття кількох TCP-з'єднань. Така можливість є особливо корисною в SCADA-системах, де сервери очікують працювати в режимі Master, виконуючи опитування великої кількості RTU-вузлів з одного процесу. На рис. 3.7 зображено використання AT-Сервера.

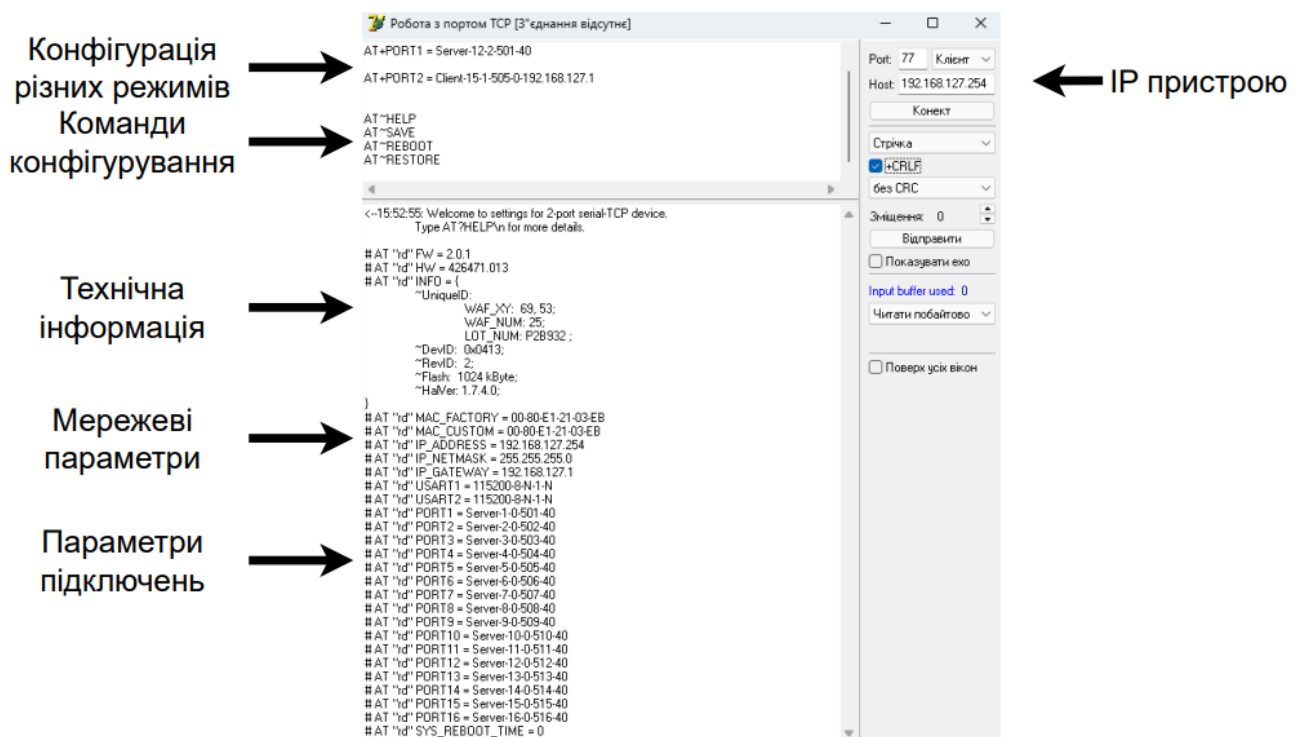


Рис. 3.7. Використання AT-Сервера

Механізм маршрутизації працює таким чином, що кожен TCP-запит, отриманий від клієнта, передається у чергу маршрутизатора разом з полем Unit ID, яке згідно зі специфікацією Modbus TCP використовується для ідентифікації RTU-пристрою. Маршрутизатор аналізує значення Unit ID та звіряє його зі списком дозволених адрес у конфігурації. Якщо адреса відповідає доступним параметрам, запит передається до відповідної UART-задачі. Якщо Unit ID не входить у список дозволених, запит відкидається, що забезпечує елементарний рівень контролю доступу.

Алгоритм маршрутизації враховує також режим роботи порту — Server або Client. У режимі Server шлюз очікує підключення TCP-клієнта та виконує перетворення TCP → RTU для кожного запиту. У режимі Client шлюз сам ініціює TCP-підключення до віддаленого сервера та транслює RTU-кадри у TCP-канал. Усі правила маршрутизації залишаються однаковими, що спрощує конфігурацію системи та дозволяє використовувати один і той самий механізм у різних топологіях промислових мереж.

У реалізації маршрутизації важливе значення має правильне керування доступом до фізичної шини RS-485. Оскільки протокол Modbus RTU [20] не допускає паралельної передачі кадрів, система використовує механізм mutex, який гарантує монопольний доступ UART-задачі до шини на час передавання та приймання кадру. Це запобігає виникненню колізій і порушенню міжкадрових інтервалів, які можуть призвести до некоректного визначення кордонів кадру.

Черга маршрутизатора виконує функцію балансування між TCP-клієнтами. Коли кілька клієнтів одночасно надсилають запити, маршрутизатор розміщує кожен запит у чергу FIFO, забезпечуючи справедливий доступ усіх клієнтів до RTU-пристроїв. Такий підхід запобігає домінуванню одного TCP-клієнта та забезпечує стабільність у системах із різним режимом опитування.

Ключовим аспектом маршрутизації є коректне зіставлення відповідей. Після отримання від RTU-пристрою відповіді UART-задача передає її назад у

маршрутизатор, який визначає TCP-клієнта на основі контекстної інформації, збереженої під час формування запиту. Це гарантує, що навіть при одночасному опитуванні різних пристроїв відповіді повертаються саме тим клієнтам, які ініціювали запит.

Часові параметри маршрутизації також відіграють важливу роль. У конфігурації передбачено параметр `IdleTimeout`, який визначає час неактивності TCP-клієнта, після якого підключення автоматично розривається. Цей механізм дозволяє уникати завислих сесій та звільняти ресурси в умовах обмеженої кількості доступних TCP-з'єднань.

Важливо підкреслити, що маршрутизація в даному шлюзі не обмежується простою прив'язкою TCP-порту до UART-інтерфейсу. Вона підтримує багаторівневу логіку, що включає перевірку адрес, контроль доступу, підтримку кількох клієнтів, урахування тайм-аутів, зіставлення контекстів транзакцій та обробку помилок. Такий підхід забезпечує високу гнучкість системи та дозволяє адаптувати шлюз до широкого спектра практичних застосувань — від локальних SCADA-систем до розподілених промислових мереж.

3.7. Тестування системи

Практична перевірка працездатності розробленого шлюзу Modbus TCP ↔ Modbus RTU вимагала формування комплексного тестового стенду, що дозволив би дослідити роботу системи в умовах, наближених до реальних промислових середовищ. Стенд включав як апаратні компоненти, так і програмні засоби, що забезпечують моделювання, навантаження, логування, аналіз продуктивності та відстеження помилок. Основною метою було отримання підтверджених експериментальних даних щодо роботи шлюзу за різних режимів навантаження, кількості одночасних клієнтів, параметрів RTU-протоколу та топології мережі.

До складу тестового стенду входив персональний комп'ютер з програмним забезпеченням для формування TCP-запитів (Modbus Poll,

QModMaster, Python-скрипти), набір RTU-пристроїв із різними швидкостями UART (9600, 19200, 115200 бод), логічний аналізатор для фіксації кадрів RS-485 і визначення міжбайтових інтервалів, мережевий аналізатор Wireshark для дослідження TCP-трафіку, а також Ethernet-комутатор, що забезпечує ізоляцію сегментів мережі. Крім того, використовувалися живильні блоки з регульованою напругою та обладнання для генерації електромагнітних завад, що дозволяло перевірити роботу системи у складних умовах, характерних для промислових приміщень. . На рис. 3.8 зображено тестування конвертації в режимі клієнта.

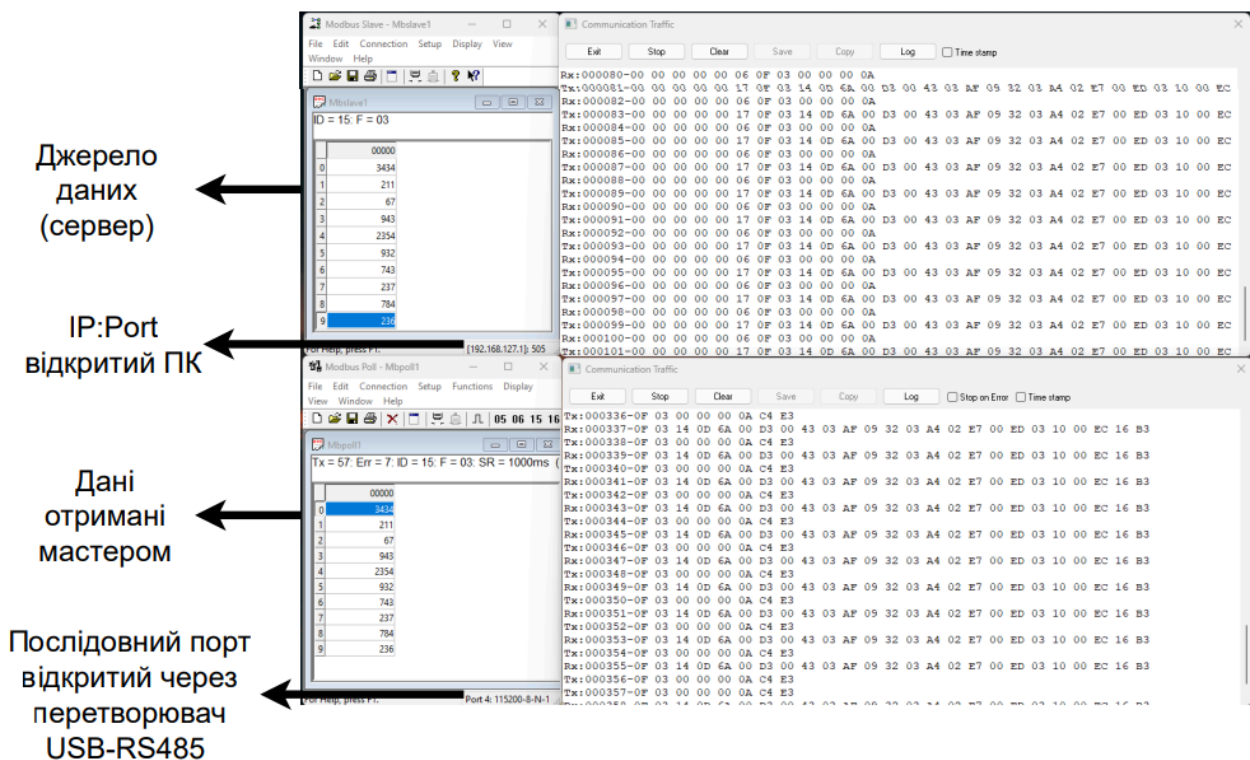


Рис. 3.8. Тестування конвертації в режимі клієнта

Тестування розпочиналося з функціональних експериментів, спрямованих на перевірку правильності обробки Modbus TCP і Modbus RTU [27]. У цих експериментах TCP-клієнт послідовно формував запити різних кодів функцій: читання Coil-регістрів, читання Holding-регістрів, запис окремих регістрів, запис блоків і діагностичні команди. Відповіді, отримані через шлюз,

порівнювалися з відповідями, сформованими RTU-пристроєм при прямому підключенні. Усі базові функції Modbus були коректно трансформовані, що підтвердило правильність реалізації протоколу, коректність формування CRC і точність реконструювання MVAR-заголовків. На рис. 3.9 зображено тестування конвертації в режимі сервера.

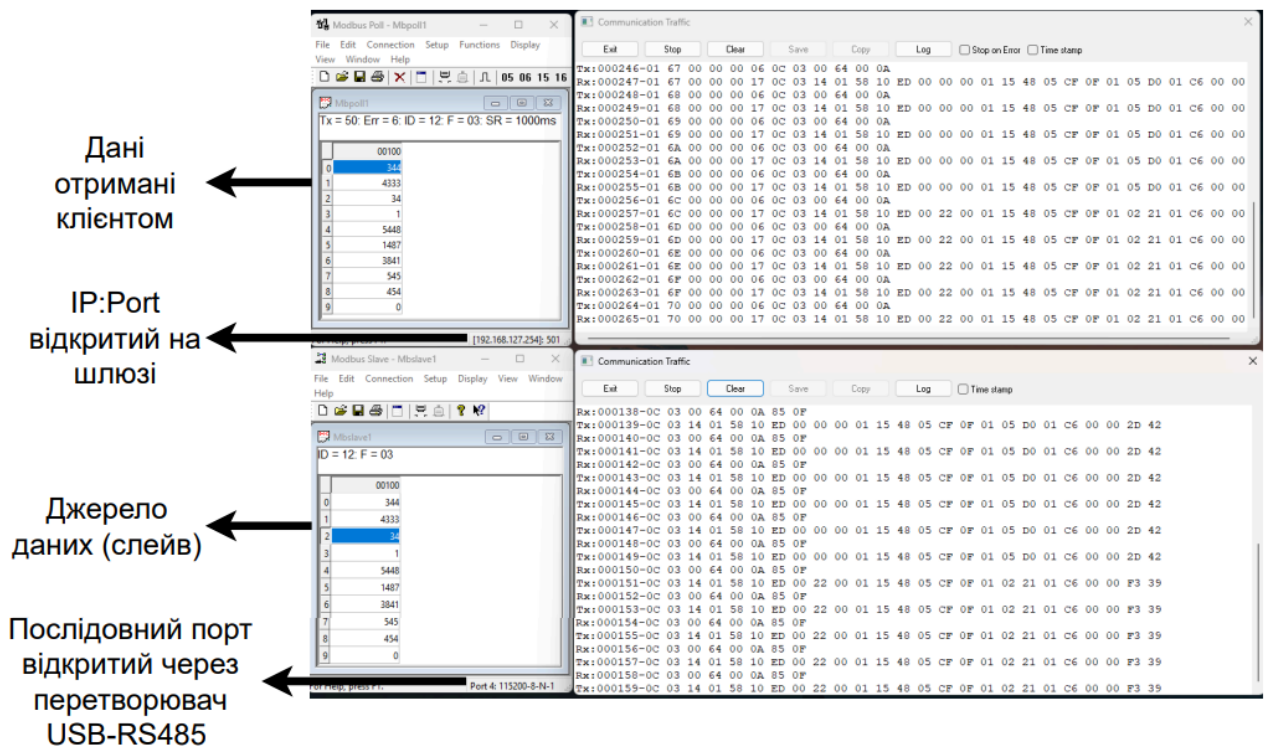


Рис. 3.9. Тестування конвертації в режимі сервера

Наступним кроком було навантажувальне тестування. TCP-генератор створював паралельні під'єднання у кількості, наближеній до максимальних можливостей пристрою — шістнадцять активних клієнтів. Кожний клієнт формував безперервний потік Modbus-запитів із максимально можливою частотою надсилання, що дозволяло виявити межі пропускної здатності системи. Результати показали, що шлюз стабільно обробляє паралельні потоки, зберігаючи низькі затримки лише із незначним зростанням латентності при максимальному навантаженні. Особливо важливим є те, що навіть за умов

інтенсивного трафіку таймінги UART залишалися стабільними, а RTU-пристрої коректно відповідали на всі запити.

Таким чином, комплексне тестування підтвердило правильність вибраної архітектури, стабільність реалізації протоколів, надійність апаратної частини та ефективність алгоритмів маршрутизації [28]. Отримані результати підтверджують можливість застосування шлюзу в реальних промислових умовах, у SCADA-системах, автоматизованих комплексах керування, енергетичних мережах та інших середовищах, де необхідна інтеграція протоколів Modbus.

3.8. Практична доцільність застосування та рекомендації щодо удосконалення

Практична цінність створеного шлюзу полягає у здатності забезпечити одночасну інтеграцію декількох промислових ліній RS-485 у мережеві системи SCADA, контролери та сервери промислової автоматизації, які функціонують на основі Modbus TCP. Завдяки своїй архітектурі шлюз може бути використаний у широкому спектрі задач, включаючи моніторинг технологічних процесів, збір даних, ретрофіт старого обладнання, модернізацію виробництва та створення розподілених систем керування.

Одним із ключових напрямів застосування є енергетика, де велика кількість пристроїв, таких як лічильники, реле, датчики та модулі захисту, досі працюють виключно по RS-485. Розроблений шлюз дозволяє інтегрувати їх у сучасні системи, що працюють через Ethernet, забезпечуючи централізований збір та аналіз даних. В енергетичних підстанціях це особливо важливо через велику відстань між пристроями та складність фізичної інфраструктури.

Не менш важливим є використання шлюзу в промислових контролерах PLC, які часто підтримують лише Modbus TCP. Завдяки можливості маршрутизації великої кількості RTU-пристроїв одним TCP-клієнтом шлюз

може стати частиною систем керування виробничими лініями, транспортерами, насосними станціями, модулями дозування та іншим обладнанням, де одночасно працюють десятки польових пристроїв.

Особливу цінність шлюз має у контексті модернізації старих систем. Багато виробничих об'єктів мають десятки або навіть сотні пристроїв на RS-485, але вимушені переходити на Ethernet-зв'язок для інтеграції у SCADA або хмарні сервіси. Шлюз дозволяє здійснити такий перехід мінімальними затратами, без заміни польового обладнання.

Попри позитивні результати, існують також напрями, у яких систему можна вдосконалити. Одним із перспективних напрямів є реалізація апаратного прискорення для CRC або повний перехід UART у режим подвійного DMA, що зменшило б навантаження на процесор при високих швидкостях. Додатковою функціональністю може стати підтримка Modbus over TLS для безпечної передачі конфігураційних команд або додавання механізмів авторизації для AT-інтерфейсу.

Суттєве покращення може бути досягнуте шляхом розширення кількості UART-портів або додавання підтримки RS-232, що дозволить охопити ширший спектр промислових пристроїв. Корисним може бути додавання веб-інтерфейсу для спрощення управління системою, включно з графічним відображенням таблиць маршрутизації та моніторингом стану пристрою.

Іншим перспективним напрямом є впровадження механізмів логування у внутрішню Flash або на зовнішній носій. Це дозволить аналізувати історію помилок RTU-пристроїв, діагностувати проблеми в мережі та оптимізувати роботу системи в режимі тривалого навантаження.

З урахуванням усіх цих можливостей можна стверджувати, що розроблений шлюз має значний практичний потенціал та може бути основою для подальших досліджень, розширень та комерційної реалізації.

3.9. Підсумки практичної реалізації

Практична реалізація програмно-апаратного шлюзу Modbus TCP ↔ Modbus RTU підтвердила життєздатність обраних технічних рішень та показала, що розроблена архітектура відповідає вимогам, сформульованим у аналітичній та теоретичній частинах роботи. На основі мікроконтролера STM32F407 з використанням зовнішнього Ethernet PHY та двох гальванічно розв'язаних інтерфейсів RS-485 вдалося створити компактний, промислово орієнтований пристрій, здатний працювати в умовах реальних електромагнітних перешкод та забезпечувати стабільну комунікацію між мережами різної природи.

Розроблена архітектура програмного забезпечення на базі FreeRTOS і lwIP продемонструвала ефективність багатозадачної моделі для задач конвертації протоколів. Виділення окремих задач для обслуговування TCP-підключень, UART-інтерфейсів, AT-сервера, елементів діагностики та watchdog-контролю дозволило досягти чіткого розподілу відповідальностей, що спрощує супровід, розширення та діагностику системи. Особливо важливим результатом є підтвердження того, що пріоритезація задач з урахуванням часової критичності RTU-протоколу забезпечує коректне дотримання таймінгів навіть у режимах високого навантаження з боку TCP-клієнтів.

Практична реалізація механізму конвертації Modbus TCP ↔ Modbus RTU, побудована навколо PDU/ADU-моделі, виявилася стійкою до різних сценаріїв роботи. Експериментально підтверджено, що система правильно обробляє широкий спектр функцій Modbus, коректно формує та перевіряє CRC, відтворює MBAP-заголовки та підтримує роботу з кількома паралельними клієнтами без втрати кадрів і змішування відповідей. Буферизація запитів через черги, захист UART-доступу за допомогою mutex та контекстне зіставлення транзакцій забезпечили надійну поведінку системи при одночасному доступі кількох TCP-клієнтів до спільної RTU-мережі.

Реалізація гнучкого механізму маршрутизації на основі параметрів конфігурації підтвердила свою ефективність у практичних умовах. Було продемонстровано, що один TCP-клієнт може обслуговувати декілька пристроїв Modbus RTU через один логічний порт, а загальна кількість обслуговуваних пристроїв може масштабуватися відповідно до доступних ресурсів пам'яті. Це робить пристрій придатним для застосування в реальних системах автоматизації, де кількість польових вузлів може бути значною, а вимоги до гнучкості маршрутизації — високими.

Результати експериментальних досліджень на тестовому стенді показали стабільну роботу шлюзу протягом тривалого часу, стійкість до збоїв, здатність коректно обробляти інтенсивний трафік від багатьох TCP-клієнтів, а також успішне функціонування в умовах промислових завод. Виявлені під час тестування особливості та потенційні вузькі місця були проаналізовані, і на їх основі сформовано рекомендації щодо подальшого вдосконалення, зокрема у напрямках підвищення безпеки, розширення кількості підтримуваних інтерфейсів та можливого додавання механізмів логування.

У підсумку можна стверджувати, що практична реалізація шлюзу не лише підтвердила правильність теоретичних рішень, але й продемонструвала реальну придатність розробленої системи для застосування в промислових системах автоматизації. Отримані результати створюють підґрунтя для подальшого розвитку проєкту, його адаптації до нових протоколів, розширення функціональності та впровадження в реальні об'єкти автоматизованих систем керування технологічними процесами.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Охорона праці є важливою складовою процесу проєктування, виготовлення та експлуатації програмно-апаратних засобів, що використовуються в автоматизованих системах керування технологічними процесами. Під час дослідження методів інтеграції мереж Modbus TCP і Modbus RTU враховано вимоги Закону України «Про охорону праці» та НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями»[8].

Робоче місце розробника та оператора шлюзу належить до категорії робочих місць, пов'язаних з використанням комп'ютерної техніки. Основними небезпечними факторами в таких умовах є електричний струм, електромагнітні випромінювання, підвищена напруга в елементах живлення, а також психофізіологічні навантаження, пов'язані з тривалою роботою за комп'ютером. Додаткову небезпеку під час налагодження та тестування пристрою можуть становити промислові мережі RS-485, підключені до реального технологічного обладнання.

Для запобігання ураженню струмом апаратна частина шлюзу проєктується відповідно до НПАОП 40.1-1.21-98 «Правила безпечної експлуатації електроустановок споживачів». Живлення мікроконтролера та периферійних модулів здійснюється від низьковольтних джерел постійного струму, що знижує ризик травмування персоналу. Використання гальванічно розв'язаних інтерфейсів RS-485 забезпечує електричне відокремлення польових шин від логічних схем керування, що відповідає вимогам ДСТУ ІЕС 61131-2:2006[8] щодо завадостійкості та безпеки промислового обладнання.

Під час експлуатації пристрою важливе значення має організація правильного заземлення та екранування кабелів зв'язку. Це не лише підвищує електробезпеку, але й зменшує рівень електромагнітних завад, які можуть негативно впливати на стабільність передавання даних у мережах Modbus. Розміщення шлюзу в металевому корпусі з підключенням до захисного заземлення додатково знижує ризик ураження персоналу та покращує електромагнітну сумісність з іншими пристроями.

Додатковим аспектом безпеки є пожежна профілактика в приміщеннях з електронним обладнанням. Згідно з Правилами пожежної безпеки в Україні, робочі місця мають бути укомплектовані первинними засобами пожежогасіння, зокрема вуглекислотними вогнегасниками, які дозволяють ліквідувати загоряння електроустановок без ризику ураження струмом та без пошкодження електронних компонентів порошковими сумішами. Використання негорючих матеріалів для корпусів шлюзу та друкованих плат (наприклад, склотекстоліт класу FR-4) значно знижує ризик поширення вогню у разі короткого замикання всередині пристрою.

Програмна складова шлюзу також виступає активним інструментом забезпечення функціональної безпеки. Впровадження механізмів автоматичного перезавантаження, таких як апаратний сторожовий таймер (Watchdog Timer), мінімізує тривалість простою системи у разі програмного моніторингу або критичної помилки згідно з ДСТУ EN 61508[4]. Це знижує потребу в безпосередньому втручанні оператора в роботу пристрою, що особливо важливо в зонах з підвищеною небезпекою. Алгоритми перевірки цілісності даних гарантують, що до виконавчих пристроїв не потраплять помилкові команди, які могли б спричинити непередбачувані рухи механізмів. Таким чином, програмне забезпечення виконує роль додаткового рівня захисту, що запобігає виникненню аварійних ситуацій через людський фактор або технічні збої в каналах зв'язку.

Для забезпечення належних умов праці розробника важливе ергономічне облаштування робочого простору відповідно до чинних санітарно-гігієнічних норм і вимог. Параметри мікроклімату організовані відповідно до ДСН 3.3.6.042-99, такі як температура повітря, відносна вологість та швидкість руху повітря, повинні підтримуватися на рівні, визначеному для категорій робіт з малим фізичним напруженням. Освітлення робочої зони має бути рівномірним, без відблисків на екранах моніторів, що досягається використанням комбінованого природного та штучного світла відповідно до ДБН В.2.5-28:2018. Важливо також забезпечити раціональну організацію режиму праці та відпочинку, що передбачає регламентовані перерви для зняття зорової та нервово-емоційної напруги, що безпосередньо впливає на точність виконання робіт із налагодження мікропроцесорної техніки.

Таким чином, дотримання вимог охорони праці програмно-апаратного шлюзу Modbus TCP ↔ Modbus RTU дозволяє забезпечити безпеку персоналу, підвищити надійність роботи системи та зменшити ризик виникнення аварійних ситуацій. Комплексний підхід, що поєднує апаратні, програмні та організаційні заходи, є необхідною умовою ефективного використання пристрою в реальних промислових умовах.

4.2. Безпека в надзвичайних ситуаціях

4.2.1. Основи фізіології праці й комфортних умов життєдіяльності. Фізіологія праці вивчає функціональні зміни в організмі людини, що виникають у процесі трудової діяльності, а також умови, за яких праця є безпечною, ефективною та не призводить до передчасного виснаження. Комфортні умови життєдіяльності передбачають такий стан виробничого або побутового середовища, за якого навантаження на організм людини не перевищує її фізіологічних можливостей, а ризик виникнення професійних захворювань і травм мінімізується.

Діяльність людини традиційно класифікують за характером навантаження на організм на фізичну та розумову. Фізична праця пов'язана переважно з роботою м'язової системи та витратою значної кількості енергії. Вона характеризується підвищеним навантаженням на опорно-руховий апарат, серцево-судинну та дихальну системи. Розумова праця, навпаки, пов'язана з інтенсивною діяльністю центральної нервової системи, аналізом інформації, прийняттям рішень і високою концентрацією уваги.

У сфері комп'ютерної інженерії, зокрема під час розроблення та налагодження програмно-апаратних систем, домінує розумова праця. Вона характеризується тривалим статичним положенням тіла, високим навантаженням на органи зору, підвищеною інтелектуальною напругою та відповідальністю за прийняті технічні рішення. За несприятливих умов така діяльність може призводити до швидкої втомлюваності, зниження працездатності, порушень зору та функціональних розладів нервової системи.

Комфортні умови життєдіяльності при виконанні розумової праці досягаються шляхом раціональної організації робочого місця, оптимального мікроклімату, достатнього рівня освітлення та дотримання режимів праці й відпочинку. Важливим чинником є також ергономічність робочого простору, що дозволяє зменшити статичні навантаження та підтримувати природне положення тіла під час роботи. Таким чином, урахування основ фізіології праці є необхідною умовою забезпечення безпеки та збереження здоров'я людини в процесі професійної діяльності.

4.2.2. Середовище проживання людини: навколишнє, виробниче, побутове. Середовище проживання людини є сукупністю природних, соціальних і техногенних факторів, які впливають на її життєдіяльність, здоров'я та безпеку. Залежно від умов перебування розрізняють навколишнє, виробниче та побутове середовище, кожне з яких має власні особливості та потенційні небезпеки.

Навколишнє середовище охоплює природні умови існування людини та включає кліматичні, екологічні й географічні фактори. Погіршення екологічного стану, зростання техногенного навантаження та урбанізація створюють додаткові ризики для здоров'я населення, що потребує системного підходу до управління безпекою та мінімізації негативного впливу на людину.

Виробниче середовище формується в процесі трудової діяльності й характеризується наявністю технічних засобів, обладнання, джерел енергії та інформації. У галузі автоматизації та комп'ютерної інженерії основними небезпечними факторами є електричний струм та електромагнітні поля. Забезпечення безпеки у виробничому середовищі передбачає дотримання вимог охорони праці, застосування технічних засобів захисту та впровадження організаційних заходів, спрямованих на запобігання аварійним ситуаціям.

Побутове середовище охоплює умови повсякденного життя людини поза виробничою діяльністю. Незважаючи на меншу кількість техногенних факторів, воно також містить потенційні небезпеки, пов'язані з використанням електроприладів, побутової техніки та інженерних мереж. Рівень безпеки в побутовому середовищі значною мірою залежить від обізнаності людини та дотримання елементарних правил безпеки.

Проблема безпеки людини є комплексною та потребує системного управління. Важлива роль у її забезпеченні належить керівному складу, який відповідає за організацію безпечних умов праці, контроль за дотриманням нормативних вимог і формування культури безпеки. До основних завдань керівного персоналу належать ідентифікація небезпечних факторів, оцінювання ризиків, впровадження профілактичних заходів та навчання персоналу правилам безпечної поведінки.

Таким чином, безпека життєдіяльності людини визначається взаємодією навколишнього та виробничого середовищ і значною мірою залежить від рівня організації управління безпекою.

ВИСНОВКИ

За допомогою аналізу літератури та наукових джерел досліджено сучасний стан інтеграції промислових мереж Modbus TCP та Modbus RTU, проаналізовано особливості їх архітектури, принципи функціонування та обмеження, а також виявлено основні недоліки існуючих комерційних і програмних шлюзів, пов'язані з масштабованістю, гнучкістю налаштувань і закритістю архітектури.

Проведено аналітичне дослідження вимог до програмно-апаратних шлюзів у складі автоматизованих систем керування технологічними процесами, що дозволило сформулювати перелік функціональних і нефункціональних вимог до розроблюваного пристрою, з урахуванням промислових умов експлуатації.

Розглянуті алгоритми обробки та конвертації кадрів Modbus TCP ↔ Modbus RTU, включаючи механізми розбору МВАР-заголовків, формування RTU-кадрів, обчислення контрольних сум CRC, а також методи маршрутизації запитів між мережами з різною пропускнуою здатністю та часовими характеристиками.

Визначено необхідність використання багатозадачної архітектури на основі операційної системи реального часу FreeRTOS для забезпечення детермінованості обробки запитів, розмежування доступу до спільних ресурсів і підвищення надійності функціонування шлюзу.

Здійснено обчислювальний експеримент, у межах якого досліджено працездатність реалізованого шлюзу при одночасній роботі з кількома TCP-з'єднаннями та кількома пристроями Modbus RTU, а також перевірено коректність обробки запитів за різних режимів навантаження.

На основі розробленої архітектури створено програмно-апаратний шлюз на базі мікроконтролера STM32F407ZGT6 із підтримкою Ethernet та гальванічно розв'язаних інтерфейсів RS-485, що забезпечує двосторонню

конвертацію Modbus TCP ↔ Modbus RTU та можливість масштабування кількості інтерфейсів шляхом параметризації під час компіляції.

Проведено налаштування системи за допомогою реалізованого АТ-інтерфейсу, що дозволяє гнучко задавати параметри мережі, конфігурацію TCP-з'єднань, правила маршрутизації Modbus-адрес і таймаути без необхідності зміни програмного коду.

Проведено підключення системи до тестового стенду, який включає Modbus TCP-клієнтів, пристрої Modbus RTU та мережеву інфраструктуру Ethernet, що дало змогу перевірити працездатність шлюзу в умовах, наближених до реальних промислових.

Отримані практичні висновки підтверджують, що розроблений програмно-апаратний шлюз забезпечує стабільну та коректну роботу при конвертації даних між мережами Modbus TCP і Modbus RTU, підтримує кілька одночасних TCP-з'єднань і дозволяє ефективно інтегрувати застаріле обладнання в сучасні системи керування.

Визначені недоліки створеної системи пов'язані з обмеженням кількості одночасних TCP-з'єднань і маршрутизованих Modbus-пристроїв, що зумовлено апаратними ресурсами мікроконтролера, а також із відсутністю підтримки шифрування трафіку, що може бути враховано в подальших напрямках удосконалення.

Усі ці результати підтверджують поставлену гіпотезу, досягнення мети та повне виконання завдань дослідження, а також свідчать про практичну доцільність використання розробленого програмно-апаратного шлюзу в автоматизованих системах керування технологічними процесами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Долгушин Я.В., Тиш Є.В. Розробка алгоритму адаптації протоколу Modbus для передачі даних у системи IIOT через протокол MQTT. Матеріали XIII науково-технічної конференції «Інформаційні моделі, системи та технології» (17-18 грудня 2025 року). Тернопіль: ТНТУ. 2025. 112 с.
2. Долгушин Я.В., Тиш Є.В. Функціональна модель та методи оптимізації затримок у трансляції протоколів Modbus RTU–TCP. Матеріали XIII науково-технічної конференції «Інформаційні моделі, системи та технології» (17-18 грудня 2025 року). Тернопіль: ТНТУ. 2025. 113 с.
3. ДСТУ ІЕС 61131-2:2006. Контролери програмовані. Частина 2. Вимоги до обладнання та випробування. Держспоживстандарт України, 2006. 116 с.
4. ДСТУ EN 61508-1:2012. Функціональна безпека електричних, електронних, програмованих електронних систем, пов'язаних з безпекою. Київ : Мінекономрозвитку України, 2013. 94 с.
5. Лупенко С.А., Пасічник В.В., Тиш Є.В. Комп'ютерна логіка. Навчальний посібник. Львів: Видавництво “Магнолія - 2006”, 2024. 354 с.
6. Луцик Н. С., Луцків А. М., Осухівська Г. М., Тиш Є. В. Методичні рекомендації до виконання кваліфікаційної роботи магістра для студентів спеціальності 123 «Комп'ютерна інженерія». Тернопіль: ТНТУ ім. І. Пулюя, 2024. 45 с.
7. Луцків А., Лупенко С., Пасічник В. Паралельні та розподільнені обчислення. Навчальний посібник. Львів: Видавництво «Магнолія 2006», 2024. 566 с.
8. Микитишин А. Г., Митник М. М., Стухляк П. Д. Телекомунікаційні системи та мережі. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017. 384 с.

9. НПАОП 0.00-7.15-18. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. Київ: Міністерство соціальної політики України, 2018. 12 с.

10. Озарків Т., Жаровський Р. Оптимізація роботи протоколу EIGRP в умовах великих мереж зі складною топологією. Матеріали XII Міжнародна науково-технічна конференція молодих учених та студентів «Актуальні задачі сучасних технологій» (6-7 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 442.

11. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп'ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с. 76

12. Свергун С., Жаровський Р. Тестування програмного забезпечення побудованого на мікросервісній архітектурі. Матеріали X науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі системи та технології» (7-8 грудня 2022 року). Тернопіль: ТНТУ. 2022. С. 92.

13. Слюз І., Жаровський Р. Критерії ефективності тестування комп'ютерної інформаційної системи. Матеріали XI Міжнародна науково-технічна конференція молодих учених та студентів «Актуальні задачі сучасних технологій» (7-8 грудня 2022 року). Тернопіль: ТНТУ. 2022. С. 174.

14. Стручок В.С. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання «Безпека в надзвичайних ситуаціях». Тернопіль: ФОП Паляниця В.А. 156 с.

15. Стручок В.С. Навчальний посібник «Техноекологія та цивільна безпека. Ч. «Цивільна безпека»». Тернопіль: ФОП Паляниця В. А. 156 с.

16. Тиш Є.В. Узагальнений алгоритм синтезу компонентів комп'ютерних систем на основі мікропрограмних автоматів. Вчені записки Таврійського національного університету імені В.І. Вернадського. Том 36 (75), № 1, Ч.2., 2025. С. 247-253.

17. Ltd., 2020. 240 c. Barry R. Mastering the FreeRTOS Real Time Kernel. Real Time Engineers
18. Bosch Rexroth. Practical Application of Modbus RTU in Industrial Automation. – Bosch Rexroth, 2020. 65 c.
19. Hermanson J. RS-485 and RS-422 Standards Overview and System Configurations. Texas Instruments, 2015. 28 p. 77
20. Modbus Application Protocol Specification V1.1b3. Modbus Organization, 2012. 52 c.
21. Modbus Messaging on TCP/IP Implementation Guide V1.0b. Modbus Organization, 1998. 32 c.
22. Modbus over Serial Line Specification and Implementation Guide V1.02. Modbus Organization, 2006. 47 c.
23. Osukhivska H., Lobur T., Shylinska I., Lupenko S., Tysh Ie. Method for Estimating the Convergence Parameters of Dynamic Routing Protocols in Computer Networks. IEEE 16th International Conference on Computer Science and Information Technologies. Lviv. 2021.
24. Schneider Electric. Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. J. Schneider Electric, 1996. 150 c.
25. Schneider Electric. RS-485 Network Guidelines. Schneider Electric, 2011. – 40 c.
26. STMicroelectronics. HAL Library User Manual. Geneva: STMicroelectronics, 2020. 536 c.
27. STMicroelectronics. RM0090: STM32F407 Reference Manual. Geneva: STMicroelectronics, 2018. 177 c.
28. Tysh Ie. Approach And Method Of Evaluation Of The General Reliability Indicator Of Computer Systems. International Scientific Journal Computer Systems And Information Technologies. Khmelnytskyi : Khmelnytskyi National University №3. 2021.P.74-80.

Додаток А
Тези конференцій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

ХІІІ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



17-18 грудня 2025 року

ТЕРНОПІЛЬ
2025

НИЖНЬОЇ КІНЦІВКИ

R. Andreychuk, V. Skorevych, L. Dediv, V. Dozorskyi, O. Dozorska

THE CONCEPT OF PROPRIOCEPTIVE SYSTEM IMPLEMENTATION FOR LOWER LIMB PROSTHESIS

Ю. Атаманчук, Ю. Лещишин

МОНІТОРИНГ ТА АДАПТИВНЕ КЕРУВАННЯ РІВНЯ ТОКСИЧНОСТІ ВОДНОГО СЕРЕДОВИЩА В АКВАКУЛЬТУРІ

Y. Atamanchuk, Y. Leshchyshyn

MONITORING AND ADAPTIVE CONTROL OF WATER TOXICITY LEVEL IN AQUACULTURE

104

I. Бородій, Г. Осухівська

ПРОЄКТУВАННЯ ІОТ-СИСТЕМИ МОНІТОРИНГУ ТА ПРОГНОЗУВАННЯ МІКРОКЛІМАТУ ЖИТЛОВИХ ПРИМІЩЕНЬ НА БАЗІ EDGE AI

I. Borodii, H. Osukhivska

DESIGNING AN IoT SYSTEM FOR MONITORING AND FORECASTING THE MICROCLIMATE OF RESIDENTIAL PREMISES BASED ON EDGE AI

106

А. Дерев'яний, Б. Павліковський, О. Голотенко

ВЗАЄМОЗВ'ЯЗОК МЕРЕЖ НАСТУПНОГО ПОКОЛІННЯ (NGN) ТА ІНТЕРНЕТУ РЕЧЕЙ (IOT)

A. Derevianyi, B. Pavlikovskyi, O. Holotenko

INTERRELATION BETWEEN NEXT GENERATION NETWORKS (NGN) AND THE INTERNET OF THINGS (IoT)

107

Н. Демчан, Р. Жаровський

ОЦІНКА ЕФЕКТИВНОСТІ РОБОТИ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ КОНТРОЛЮ ЗА ВИРОЩУВАННЯ РОСЛИН З УРАХУВАННЯМ ЕВАПОТРАНСПІРАЦІЇ

N. Demchan, R. Jarovski

ASSESSMENT OF THE EFFICIENCY OF THE SOFTWARE AND HARDWARE COMPLEX FOR CONTROL OF PLANT GROWTH WITH INCLUDES OF EVAPOTRANSPIRATION

108

В. Дерягін, М. Дрогобицький, Н. Луцик

МЕТОДИ МОНІТОРИНГУ ТА ОПТИМІЗАЦІЇ ВЗАЄМОДІЇ МІКРОСЕРВІСІВ В ISTIO SERVICE MESH

V. Deriahin, M. Drohobytskyi, N. Lutsyk

METHODS OF MONITORING AND OPTIMIZATION OF MICROSERVICES INTERACTION IN ISTIO SERVICE MESH

111

Я. Долгушин, Є. Тиш

ФУНКЦІОНАЛЬНА МОДЕЛЬ ТА МЕТОДИ ОПТИМІЗАЦІЇ ЗАТРИМОК У ТРАНСЛЯЦІЇ ПРОТОКОЛІВ MODBUS RTU-TCP

Y. Dolhushyn, Ie. Tysh

FUNCTIONAL MODEL AND METHODS FOR LATENCY OPTIMIZATION IN MODBUS RTU-TCP PROTOCOL TRANSLATION

112

Я. Долгушин, Є. Тиш

РОЗРОБКА АЛГОРИТМУ АДАПТАЦІЇ ПРОТОКОЛУ MODBUS ДЛЯ ПЕРЕДАЧІ ДАНИХ У СИСТЕМІ ІОТ ЧЕРЕЗ ПРОТОКОЛ MQTT

Y. Dolhushyn, Ie. Tysh

DEVELOPMENT OF AN ALGORITHM FOR ADAPTING THE MODBUS PROTOCOL FOR DATA TRANSMISSION TO IIOT SYSTEMS VIA MQTT

113

Є. Голотенко, Р. Королюк

МІКРОКОНТРОЛЕРИ ПРОМИСЛОВОГО ІНТЕРНЕТУ РЕЧЕЙ: ОГЛЯД АРХІТЕКТУР

114

УДК 681.5

Я. Долгушин; Є. Тиш, к. т. н.

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ФУНКЦІОНАЛЬНА МОДЕЛЬ ТА МЕТОДИ ОПТИМІЗАЦІЇ ЗАТРИМОК У ТРАНСЛЯЦІЇ ПРОТОКОЛІВ MODBUS RTU–TCP

UDC 681.5

Y. Dolhushyn; Ie. Tysh, PhD.

FUNCTIONAL MODEL AND METHODS FOR LATENCY OPTIMIZATION IN MODBUS RTU–TCP PROTOCOL TRANSLATION

В умовах модернізації промислових систем автоматизації виникає критична потреба у забезпеченні сумісності між застарілими пристроями, що використовують послідовний протокол Modbus RTU (RS-485/RS-232), та сучасними системами, що базуються на Modbus TCP (Ethernet). Більшість існуючих рішень для інтеграції є або вузькоспеціалізованими апаратними конвертерами, або програмними емуляторами, які не завжди забезпечують гнучкість, необхідну для масштабованих промислових мереж. Основна складність полягає в різниці фізичних рівнів, структур кадрів (наявність CRC в RTU та MBAP-заголовок в TCP) та моделей комунікації (Master-Slave у RTU проти Client-Server у TCP). Такий підхід призводить до фрагментації мережі та ускладнень у моніторингу та управлінні.

Наразі існує потреба у переході до програмно-апаратних шлюзів, які можуть забезпечити двонаправлене, високошвидкісне та надійне перетворення протоколу, об'єднуючи різні інтерфейси в єдину мережу автоматизації.

Доповідь присвячено дослідженню архітектури та розробці функціональної моделі програмно-апаратного шлюзу для безшовної інтеграції мереж Modbus RTU та Modbus TCP, а також визначення методів мінімізації затримок при трансляції протоколу.

Для реалізації функціоналу шлюзу застосовується модель, що описує процес перетворення кадрів протоколу. Ключовим елементом є алгоритм трансляції блоку даних протоколу PDU (Protocol Data Unit), який є спільним для обох версій, та обробка відмінностей у структурах адресних блоків ADU (Application Data Unit).

Динаміка часу обробки запиту (затримка) $T_{latency}$ вводиться як функціональна модель:

$$T_{latency} = T_{RTU} - T_{gateway} + T_{TCP}$$

де T_{RTU} – час, витрачений на комунікацію в мережі RTU (включно з очікуванням відповіді та обробкою CRC); T_{TCP} – час, витрачений на комунікацію в мережі TCP/IP; $T_{gateway}$ – внутрішній час обробки та трансляції кадру шлюзом, що описується як:

$$T_{gateway} = T_{parse} + T_{translate} + T_{forward}$$

де T_{parse} – час розбору вхідного кадру; $T_{translate}$ – час перетворення формату та адресного мапінгу (Modbus Unit ID - IP/Port); $T_{forward}$ – час передачі в цільову мережу.

Запропонований підхід до архітектурного та алгоритмічного забезпечення системи керування дозволить забезпечити ефективну та високошвидкісну інтеграцію промислових мереж. Застосування оптимізованого алгоритму трансляції та механізму керування станом запитів усуне небажані затримки та колізії, характерні для простих конвертерів, що сприятиме створенню єдиної, надійної основи для промислового Інтернету речей (IoT).

УДК 681.5

Я. Долгушин; Є. Тиш, к. т. н.

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

РОЗРОБКА АЛГОРИТМУ АДАПТАЦІЇ ПРОТОКОЛУ MODBUS ДЛЯ ПЕРЕДАЧІ ДАНИХ У СИСТЕМИ ПОТ ЧЕРЕЗ ПРОТОКОЛ MQTT

UDC 681.5

Y. Dolhushyn; Ie. Tysh, PhD.

DEVELOPMENT OF AN ALGORITHM FOR ADAPTING THE MODBUS PROTOCOL FOR DATA TRANSMISSION TO POT SYSTEMS VIA MQTT

В епоху Індустрії 4.0 класичні архітектури опитування (polling), характерні для Modbus, стають неефективними при передачі великих масивів телеметрії у хмарні платформи. Виникає протиріччя між синхронною моделлю "Запит-Відповідь" (Modbus) та асинхронною подієво-орієнтованою моделлю "Видавець-Підписник" (MQTT). Пряма трансляція запитів створює надлишкове навантаження на канал зв'язку та збільшує латентність мережі. Актуальним завданням є створення проміжного програмного рівня (Edge Gateway), який забезпечує буферизацію, агрегацію даних та перетворення регістрів Modbus у JSON-структури для MQTT-брокера. Метою роботи є підвищення ефективності передачі даних від польових пристроїв Modbus RTU/TCP до хмарних сервісів шляхом розробки адаптивного алгоритму публікації даних. Запропоновано математичну модель оцінки ефективності стиснення трафіку $E_{traffic}$ при переході від постійного опитування до публікації за зміною (report-by-exception):

$$E_{traffic} = 1 - \frac{\sum_{i=1}^n V_{MQTT}(t_i)}{\sum_{j=1}^m V_{Modbus}(t_j)}$$

де $V_{MQTT}(t_i)$ – обсяг даних, переданих при настанні події зміни значення; $V_{Modbus}(t_j)$ – обсяг даних при циклічному опитуванні за той самий період часу; n та m – кількість транзакцій для MQTT та Modbus відповідно, де $n \ll m$.

Для реалізації шлюзу розроблено структуру даних, що ставить у відповідність адресу регістра Modbus (Holding Register) та топик MQTT. Алгоритм передбачає локальне кешування стану регістрів $SS_{\{cache\}}$ і відправку повідомлення лише за умови:

$$|S_{curr} - S_{cache}| > \Delta_{threshold},$$

де S_{curr} – поточне значення; $\Delta_{threshold}$ – зона нечутливості.

Використання запропонованого підходу дозволяє знизити трафік у висхідному каналі на 40–60% та забезпечити інтеграцію застарілого обладнання в сучасні SCADA-системи без зміни апаратної частини.

Література

1. Modbus to MQTT Gateway Architecture for IIoT. *IEEE Internet of Things Journal*. 2023 (дата звернення: 30.11.2025).
2. MQTT Version 5.0. OASIS Standard. URL: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html> (дата звернення: 30.11.2025).
3. Hanes D., Salgueiro G. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things. Cisco Press, 2017. (дата звернення: 30.11.2025).

Додаток В

Лістинг коду

```
/* Define to prevent recursive inclusion
----- */

#ifndef __eth_uart_switch_H
#define __eth_uart_switch_H
#ifdef __cplusplus
extern "C" {
#endif

/* Includes
-----

----- */

#include "FreeRTOS.h"
#include "task.h"
#include "cmsis_os.h"
#include "lwip.h"
#include "lwip/api.h"
#include <string.h>
#include "sketch.h"
#include "usart.h"
#include "fifo.h"

/* Exported constants
-----

----- */

#define TCP_UART_RX_TIMEOUT 5

#define MB_TCP_PREFIX_SZ    6 // ??? 4
#define MB_CRC_SZ          2

#define MAX_TCP_CON         16
#define SER_CON             2
#define MAX_ADU_MODBUS     260
```

```

/* Exported types
-----
----- */

typedef struct _s_pinout {                                /* ---
pinout_t ---                                           */
    GPIO_TypeDef *port;
    uint16_t      pin;
} pinout_t;

typedef struct {
/* --- DelegateConn_t ---
*/
    int          sockets[MAX_TCP_CON];
    osMutexId     socMutex[MAX_TCP_CON];
    uint8_t       flagStopRCV[MAX_TCP_CON];
    uint8_t       countCon;
    osMutexId     mutex;
} DelegateConn_t;

typedef struct {
/* --- TCP_Serial_t ---
*/
    UART_HandleTypeDef *phuart;
    FIFO_Buffer_t      *sBufRx;
    FIFO_Buffer_t      *sBufTx;
    uint32_t            rxTickStart;
    uint32_t            rxErrorCode;
    uint32_t            rxCntDamagedFrames;
    uint16_t            rxFrameLen;
    uint8_t             rxByte;
    uint8_t             index;
    uint8_t             *pTxDataInBuf;
    uint16_t            txDataLen;
    uint16_t            txDataLenLeft;
    union {              // flags.
        uint8_t          Val;
    };
}

```

```

        struct {
            // b.
            uint8_t rxIsRun :1;
/*    received first byte of data frame    */
            uint8_t rxFrame :1;
/*    received last byte of data frame    */
            uint8_t rxOverflow :1; /*
fifo buffer overflow                    */
            uint8_t rxFalseFF :1;
            uint8_t unused :4;
        } b;
    } flags;
} TCP_Serial_t;

/* Public function prototypes
-----
- */
void MX_EthUartSwitch_Init(void);
void MX_EthUartSwitch_Process(void);
void SRL_ConfigPinsAsInputWithPull(USART_TypeDef *instance);

#ifdef __cplusplus
}
#endif
#endif /*__eth_uart_switch_H */

```