КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ

Кафедра радіотехніки та радіоелектронних систем

«На правах рукопису»

Робота допущена до захисту в ЕК
рішенням кафедри радіотехніки та радіоелектронних систем
від _____2026 року, протокол № _____
В.о.завідувача кафедри к.фіз.-мат.наук, доцент
_____ Ігор БЕХ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА
на тему:

«МЕТОДИ ТА ЗАСОБИ ЗАХИЩЕНОГО ВІДДАЛЕНОГО КЕРУВАННЯ ЖИВЛЕННЯМ ТЕЛЕКОМУНІКАЦІЙНОГО ОБЛАДНАННЯ В УМОВАХ КРИТИЧНИХ ЗБОЇВ»

Виконав:
студент 2-го курсу магістратури
денної форми здобуття освіти
спеціальності 172 Електронні комунікації та радіотехніка
Інформаційна безпека телекомунікаційних систем і мереж
Моргун Вадим Васильович_____

Науковий керівник:
асистент кафедри радіотехніки та
радіоелектронних систем, кандидат
фізико-математичних наук
Котов Михайло Миколайович_____

Рецензент:
д. т. н., професор
Хлапонін Юрій Іванович _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів без
відповідних посилань
Студент _____ Вадим МОРГУН

РЕФЕРАТ

Дипломний проект, 72 с., 3 дод., 24 джерела.

Ключові слова: телекомунікаційне обладнання, віддалене керування, керування живленням, STM32F411RE, W5500, Cisco, VPN, SSH, SNMP, резервне живлення, UPS, мікроконтролерна система, моніторинг доступності, power cycle, безпечне вимкнення.

Об'єкт дослідження: система віддаленого керування живленням телекомунікаційного обладнання в умовах критичних збоїв та нестабільного електроживлення.

Мета роботи: розробити та дослідити методи і засоби захищеного віддаленого керування живленням телекомунікаційного обладнання, що забезпечують автоматичний моніторинг доступності пристроїв, відновлення їх працездатності шляхом контрольованого перезавантаження, а також безпечне вимкнення при вичерпанні ресурсу резервного живлення.

Методи дослідження – аналіз принципів функціонування порівняння методів моніторингу доступності мережевих пристроїв, розробка структурної та принципової схем мікроконтролерної системи, побудова алгоритмів керування живленням, моделювання аварійних режимів роботи та експериментальна перевірка працездатності запропонованого рішення.

Результати роботи – розроблено модельну систему захищеного віддаленого керування живленням телекомунікаційного обладнання на базі мікроконтролера STM32F411RE та Ethernet-модуля W5500. Система має забезпечувати моніторинг доступності Cisco-пристрою, виявлення критичних збоїв, виконання апаратного перезавантаження через силовий комутаційний вузол, контроль основного й резервного живлення та безпечне вимкнення обладнання при критичному зниженні заряду акумулятора. За результатами модельного тестування підтверджено працездатність системи в основних аварійних режимах.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

ADC – Analog-to-Digital Converter, аналого-цифровий перетворювач

ACL – Access Control List, список контролю доступу

API – Application Programming Interface, програмний інтерфейс застосунків

ARP – Address Resolution Protocol, протокол визначення MAC-адреси за IP-адресою

DNS – Domain Name System, система доменних імен

GPIO – General-Purpose Input/Output, універсальний цифровий вхід/вихід

HAL – Hardware Abstraction Layer, рівень абстракції апаратного забезпечення

HTTP – HyperText Transfer Protocol, протокол передавання гіпертексту

HTTPS – HyperText Transfer Protocol Secure, захищений протокол передавання гіпертексту

ICMP – Internet Control Message Protocol, протокол керуючих повідомлень Інтернету

IP – Internet Protocol, міжмережевий протокол

MAC – Media Access Control, апаратна адреса мережевого інтерфейсу

SNMP – Simple Network Management Protocol, простий протокол керування мережею

SPI – Serial Peripheral Interface, послідовний периферійний інтерфейс

SSH – Secure Shell, захищений протокол віддаленого доступу

SSR – Solid State Relay, твердотільне реле

TCP – Transmission Control Protocol, протокол керування передаванням

TCP/IP – набір мережевих протоколів, що забезпечує передавання даних у комп’ютерних мережах

TTL – Time To Live, час життя пакета

UART – Universal Asynchronous Receiver/Transmitter, універсальний асинхронний приймач-передавач

VPN – Virtual Private Network, віртуальна приватна мережа

ВСТУП

Сучасні телекомунікаційні системи відіграють ключову роль у забезпеченні функціонування інформаційних сервісів, корпоративних мереж та об’єктів критичної інфраструктури. Надійність їх роботи безпосередньо залежить не лише від якості програмного забезпечення чи мережевих протоколів, але й від стабільності електроживлення та можливості оперативного реагування на збої. На практиці часто виникають ситуації, коли сервер або мережеве обладнання перестає відповідати через програмні помилки, перевантаження або збої в роботі операційної системи. У таких випадках єдиним ефективним способом відновлення працездатності залишається примусове перезавантаження.

Проблема ускладнюється тим, що доступ до обладнання не завжди є фізично можливим. Це характерно для віддалених вузлів мережі, серверних приміщень із обмеженим доступом або об’єктів, що працюють в автономному режимі. Відповідно, виникає необхідність у створенні систем, які дозволяють не лише віддалено контролювати стан обладнання, а й виконувати керування його живленням без безпосередньої присутності обслуговуючого персоналу.

Окрему увагу слід приділити питанням безпеки таких систем. Засоби керування живленням фактично надають повний контроль над роботою обладнання, тому їх несанкціоноване використання може призвести до серйозних наслідків, включаючи виведення з ладу мережевих сервісів. Саме тому при організації віддаленого доступу необхідно застосовувати захищені методи взаємодії, зокрема використання VPN-тунелювання та криптографічних протоколів, таких як SSH, що забезпечують автентифікацію користувачів і захист переданих даних. Для контролю стану мережевого обладнання доцільно використовувати протоколи управління, наприклад SNMP, які дозволяють отримувати інформацію про роботу пристроїв у режимі реального часу.

Ще одним важливим аспектом є робота системи в умовах зникнення основного електроживлення. У таких ситуаціях обладнання переходить на резервні джерела енергії, проте їх ресурс є обмеженим. Неконтрольоване вимкнення серверів може призвести до втрати даних або пошкодження файлових систем. Тому

система керування повинна передбачати алгоритми безпечного завершення роботи обладнання при досягненні критичного рівня заряду резервного живлення.

Таким чином, актуальність теми роботи зумовлена необхідністю створення комплексного рішення, яке поєднує функції моніторингу стану телекомунікаційного обладнання, захищеного віддаленого доступу та керування електроживленням в умовах можливих збоїв.

Метою роботи є розробка методів та засобів захищеного віддаленого керування живленням телекомунікаційного обладнання, що забезпечують автоматичне відновлення його працездатності та безпечне вимкнення в умовах обмеженого енергоресурсу.

Об’єктом дослідження є процеси функціонування телекомунікаційного обладнання в умовах нестабільного живлення та відмов мережевих сервісів.

Предметом дослідження є методи моніторингу доступності обладнання, алгоритми керування живленням та засоби забезпечення захищеного віддаленого доступу.

Для досягнення поставленої мети у роботі вирішуються такі задачі: проведення аналізу існуючих підходів до забезпечення відмовостійкості телекомунікаційних систем; дослідження сучасних методів захищеного віддаленого доступу; розробка структури пристрою керування живленням на базі мікроконтролера; створення алгоритмів виявлення збоїв у роботі обладнання; реалізація механізму автоматичного перезавантаження; розробка алгоритму безпечного завершення роботи при зниженні рівня резервного живлення; оцінка ефективності запропонованого рішення.

Практична цінність отриманих результатів полягає у можливості їх використання для підвищення надійності роботи серверного та мережевого обладнання, особливо у випадках, коли доступ до нього є обмеженим або віддаленим.

І. ТЕОРЕТИЧНІ ОСНОВИ ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ ТА ЗАХИЩЕНОГО КЕРУВАННЯ ТЕЛЕКОМУНІКАЦІЙНИМ ОБЛАДНАННЯМ

1.1. Особливості функціонування телекомунікаційного обладнання та причини збоїв

Сучасні телекомунікаційні системи являють собою складні багаторівневі програмно-апаратні комплекси, що забезпечують передачу, обробку та маршрутизацію інформації між просторово розподіленими вузлами. Їх функціонування базується на стеку протоколів TCP/IP та ієрархічній архітектурі побудови мереж, яка передбачає розподіл на рівні доступу, агрегації та ядра мережі [1]. Така структура забезпечує масштабованість, ефективне управління трафіком і підвищену відмовостійкість, однак водночас створює залежність великої кількості сервісів від стабільності окремих вузлів.

Телекомунікаційне обладнання, зокрема маршрутизатори, комутатори та сервери, працює у режимі безперервного функціонування, що накладає підвищені вимоги до його надійності. У процесі роботи пристрої виконують обробку великого обсягу мережевого трафіку, реалізують складні алгоритми маршрутизації та забезпечують взаємодію між різними мережевими сегментами. При цьому програмна частина обладнання, яка представлена операційними системами та вбудованим програмним забезпеченням, є потенційним джерелом відмов через складність реалізації та наявність помилок [2].

Аналіз експлуатації телекомунікаційних систем показує, що збої у їх роботі можуть виникати з різних причин. Однією з основних є програмні помилки, що призводять до зависання системи або некоректної обробки мережевих пакетів. Такі ситуації часто виникають при перевантаженні процесорних ресурсів, витоку пам’яті або некоректній роботі служб. У цьому випадку пристрій може залишатися увімкненим, але перестає виконувати свої функції, що унеможливлює доступ до нього через стандартні засоби адміністрування.

Іншою важливою причиною збоїв є перевантаження мережі або окремих її елементів. Зростання обсягів трафіку, використання ресурсомістких сервісів та поява аномальних навантажень можуть призводити до деградації якості обслуговування або повної втрати доступності сервісів. У таких умовах мережеве обладнання може переходити в нестабільний режим роботи або втрачати здатність обробляти запити в реальному часі [3].

Значний вплив на працездатність телекомунікаційних систем має стан електроживлення. Перебої напруги, короткочасні відключення або нестабільність джерел живлення можуть викликати аварійне завершення роботи пристроїв, що супроводжується втратою даних або пошкодженням програмного забезпечення. Навіть при використанні джерел безперебійного живлення (UPS) час автономної роботи є обмеженим, що потребує реалізації механізмів контролю енергоспоживання та своєчасного реагування на критичні ситуації [4].

Окрему категорію становлять апаратні збої, пов’язані з деградацією електронних компонентів, перегрівом або фізичними пошкодженнями. Хоча такі відмови трапляються рідше, їх наслідки можуть бути більш критичними, оскільки потребують фізичного втручання для відновлення працездатності.

Важливою особливістю телекомунікаційного обладнання є те, що значна частина збоїв не призводить до повного вимкнення пристрою, а лише до втрати його функціональності. У таких випадках стандартні методи віддаленого керування, засновані на мережевому доступі, стають неефективними. Це обумовлює необхідність застосування альтернативних підходів до відновлення працездатності, зокрема керування електроживленням як універсального способу примусового перезавантаження системи.

Таким чином, аналіз особливостей функціонування телекомунікаційного обладнання свідчить про те, що забезпечення його надійної роботи потребує комплексного підходу, який враховує як програмні, так і апаратні аспекти, а також умови енергопостачання. Це визначає актуальність розробки систем, здатних автоматично виявляти збої та виконувати відновлення працездатності обладнання в автономному режимі.

1.2. Методи моніторингу та виявлення збоїв у телекомунікаційному обладнанні

Забезпечення надійної роботи телекомунікаційного обладнання неможливе без постійного контролю його стану та своєчасного виявлення відхилень у функціонуванні. Моніторинг дозволяє визначати як явні відмови, так і передумови до їх виникнення, що дає змогу оперативно реагувати на критичні ситуації та запобігати повній втраті працездатності системи.

Найпростішим і найбільш поширеним методом контролю доступності мережевих пристроїв є використання протоколу ICMP для виконання перевірок типу ping. Даний підхід дозволяє визначити, чи відповідає пристрій у мережі, а також оцінити затримки та втрати пакетів. Проте, незважаючи на свою простоту, цей метод має суттєві обмеження. Зокрема, пристрій може відповідати на ICMP-запити навіть у випадку, коли його основні функції частково або повністю порушені. Таким чином, використання лише ping не забезпечує достатнього рівня достовірності моніторингу [1].

Більш інформативним підходом є перевірка доступності конкретних сервісів на транспортному рівні, що реалізується шляхом встановлення TCP-з’єднань із заданими портами. Наприклад, перевірка доступності порту 22 дозволяє оцінити працездатність служби віддаленого доступу, а порту 80 — веб-сервісу. Такий метод дає змогу визначити не лише факт доступності пристрою, але й стан окремих функціональних компонентів. Разом із тим, він також не гарантує повної картини, оскільки можливі ситуації часткової деградації системи.

Для більш глибокого аналізу стану телекомунікаційного обладнання застосовуються протоколи управління, зокрема SNMP. Використання SNMP дозволяє отримувати інформацію про внутрішні параметри пристрою, такі як завантаження процесора, використання пам’яті, стан мережевих інтерфейсів та інші показники. Це дає змогу не лише фіксувати факт збою, але й визначати його причини та оцінювати динаміку зміни параметрів системи [2].

Окрім пасивного моніторингу, важливу роль відіграють активні методи діагностики, які передбачають періодичне виконання тестових операцій або аналіз

відповідей системи на контрольні запити. У сучасних мережах широко застосовується комбінований підхід, при якому рішення про наявність збою приймається на основі декількох критеріїв. Наприклад, відсутність відповіді на ICMP-запити може доповнюватися невдалою спробою встановлення TCP-з'єднання або відсутністю відповіді на SNMP-запит. Такий підхід дозволяє значно зменшити кількість хибних спрацювань і підвищити точність діагностики.

Особливу увагу слід приділити ситуаціям, коли обладнання переходить у так званий «частково працездатний» стан. У цьому випадку пристрій може залишатися доступним у мережі, але не виконувати свої основні функції або працювати з істотними затримками. Виявлення таких станів є складнішою задачею, оскільки вони не супроводжуються повною відсутністю відповіді. Для їх діагностики необхідно аналізувати комплекс показників, включаючи час відгуку, стабільність з'єднань та поведінку прикладних сервісів.

Важливою характеристикою систем моніторингу є їх здатність працювати в умовах обмеженої доступності мережі або повної втрати зв'язку з обладнанням. У таких випадках необхідно використовувати локальні засоби контролю, які не залежать від зовнішніх систем управління. Це особливо актуально для вбудованих рішень на базі мікроконтролерів, що здійснюють автономний моніторинг і приймають рішення без участі центрального сервера.

Таким чином, ефективне виявлення збоїв у телекомунікаційному обладнанні потребує застосування комплексного підходу, який поєднує прості методи перевірки доступності, аналіз роботи сервісів та використання протоколів управління. Отримані результати моніторингу можуть бути використані для формування алгоритмів автоматичного реагування, зокрема для ініціювання перезавантаження або інших дій, спрямованих на відновлення працездатності системи.

1.3. Методи захищеного віддаленого керування телекомунікаційним обладнанням

Віддалене керування є невід'ємною складовою експлуатації сучасних телекомунікаційних систем, оскільки дозволяє здійснювати конфігурацію, діагностику та обслуговування обладнання без фізичного доступу до нього. У реальних умовах, особливо для розподілених мереж або об'єктів із обмеженим доступом, віддалене адміністрування є основним способом взаємодії з пристроями.

Базовим підходом до віддаленого керування є використання мережевих протоколів прикладного рівня, що забезпечують доступ до командного інтерфейсу або сервісів керування. Історично для цих цілей використовувався протокол Telnet, який дозволяє організувати текстову взаємодію з пристроєм. Однак цей протокол не забезпечує шифрування переданих даних, що робить його вразливим до перехоплення облікових даних та атак типу «людина посередині» [2].

Сучасним стандартом для захищеного віддаленого доступу є протокол SSH, який забезпечує шифрування трафіку, автентифікацію користувачів та захист від несанкціонованого доступу. Використання SSH дозволяє безпечно передавати команди керування, виконувати конфігурацію пристроїв та отримувати службову інформацію навіть при роботі через незахищені мережі. Завдяки застосуванню криптографічних алгоритмів, SSH гарантує конфіденційність та цілісність переданих даних [3].

Разом із тим, прямий доступ до телекомунікаційного обладнання через глобальну мережу Інтернет створює додаткові ризики безпеки. Відкриття портів для віддаленого доступу може бути використане зловмисниками для проведення атак підбору паролів або експлуатації вразливостей. У зв'язку з цим доцільно застосовувати додаткові механізми захисту, зокрема використання віртуальних приватних мереж.

Віртуальні приватні мережі (VPN) дозволяють створювати захищені тунелі між віддаленими вузлами, забезпечуючи шифрування трафіку та автентифікацію сторін. При цьому доступ до обладнання здійснюється через внутрішні (приватні) IP-адреси, що виключає необхідність відкриття сервісів керування у глобальній

мережі. Такий підхід значно зменшує поверхню атаки та підвищує рівень захищеності системи [3].

Крім організації захищеного каналу зв’язку, важливим елементом є контроль доступу до системи керування. Для цього застосовуються механізми автентифікації та авторизації користувачів, обмеження доступу за IP-адресами, а також використання списків контролю доступу (ACL). Додатково може використовуватися журналювання дій користувачів, що дозволяє відслідковувати зміни конфігурації та виявляти несанкціоновану активність.

У контексті систем керування живленням телекомунікаційного обладнання питання безпеки набуває особливої важливості. Несанкціонований доступ до таких систем може призвести до навмисного вимкнення обладнання або порушення роботи мережевої інфраструктури. Тому архітектура системи повинна передбачати розділення функцій керування та моніторингу, а також використання багаторівневого захисту, який включає мережеву ізоляцію, шифрування та контроль доступу.

Таким чином, ефективне віддалене керування телекомунікаційним обладнанням повинно поєднувати використання сучасних захищених протоколів, таких як SSH, із додатковими механізмами захисту, зокрема VPN та системами контролю доступу. Це дозволяє забезпечити не лише зручність адміністрування, але й високий рівень інформаційної безпеки, що є критично важливим для стабільної роботи телекомунікаційних систем.

1.4. Керування живленням та забезпечення працездатності в умовах резервного енергопостачання

Одним із ключових аспектів забезпечення надійності телекомунікаційного обладнання є контроль та керування його електроживленням. Як було показано у попередніх підрозділах, значна частина збоїв не пов’язана з повним відключенням пристрою, а виникає внаслідок програмних зависань або нестабільної роботи системи. У таких випадках традиційні методи віддаленого керування стають

неефективними, що обумовлює необхідність застосування апаратних засобів відновлення працездатності.

Керування живленням передбачає можливість примусового вимкнення та повторного увімкнення обладнання, що дозволяє виконати його повне перезавантаження. Даний підхід є універсальним, оскільки не залежить від стану операційної системи чи мережевих сервісів. Реалізація такого механізму може здійснюватися за допомогою релейних модулів, твердотільних ключів або спеціалізованих пристроїв розподілу живлення, керованих мікроконтролером.

Разом із тим, застосування примусового перезавантаження потребує врахування ризиків, пов’язаних із можливим пошкодженням даних або некоректним завершенням роботи програм. У зв’язку з цим сучасні системи керування повинні реалізовувати алгоритми, які передбачають попередній аналіз стану обладнання та виконання перезавантаження лише у випадку підтвердженого збою.

Особливу увагу слід приділити роботі телекомунікаційного обладнання в умовах відсутності основного електроживлення. У таких ситуаціях використовується резервне живлення, зокрема джерела безперебійного живлення (UPS), які забезпечують тимчасову підтримку працездатності системи. Проте ресурс таких джерел є обмеженим і залежить від ємності акумуляторів та споживаної потужності навантаження [4].

У режимі резервного живлення важливим завданням є ефективне використання енергоресурсу. Неконтрольована робота обладнання може призвести до повного розряду акумуляторів і аварійного вимкнення системи, що супроводжується втратою даних або пошкодженням програмного забезпечення. Для запобігання таким ситуаціям необхідно реалізувати механізми контролю рівня заряду та прогнозування часу автономної роботи.

Одним із ефективних рішень є впровадження алгоритму безпечного завершення роботи обладнання. Суть даного підходу полягає у тому, що при досягненні критичного рівня заряду резервного джерела система ініціює процедуру коректного вимкнення серверів або інших пристроїв. Це може здійснюватися

шляхом передачі відповідних команд через захищені канали зв’язку або локальні інтерфейси керування.

Після завершення роботи обладнання система керування живленням може виконати його повне відключення, що дозволяє зберегти залишковий ресурс акумуляторів та запобігти їх глибокому розряду. Такий підхід також забезпечує можливість автоматичного відновлення роботи після появи основного електроживлення.

Важливою особливістю систем керування живленням є їх автономність. У випадку втрати зв’язку з центральними серверами або системами моніторингу рішення про виконання певних дій повинно прийматися локально. Це обумовлює доцільність використання мікроконтролерних платформ, здатних реалізовувати алгоритми аналізу стану системи та керування живленням без зовнішнього втручання.

Таким чином, керування живленням телекомунікаційного обладнання є ефективним засобом підвищення його надійності та забезпечення відмовостійкості в умовах як програмних збоїв, так і нестабільного енергопостачання. Поєднання методів моніторингу, захищеного віддаленого доступу та алгоритмів автономного керування дозволяє створити комплексну систему, здатну забезпечити безперервність функціонування телекомунікаційної інфраструктури.

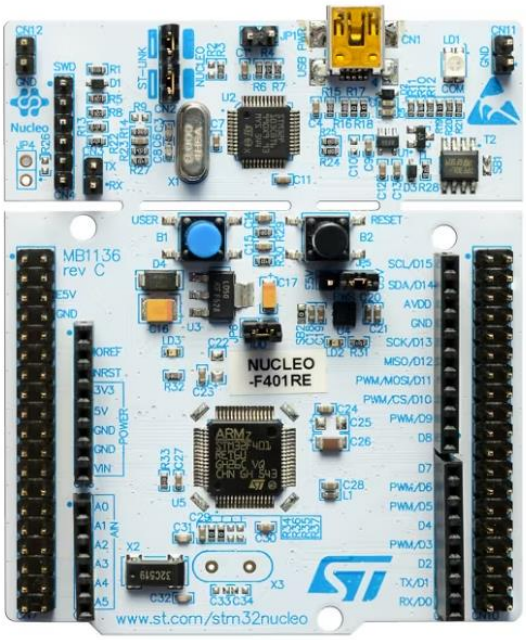
ІІ. РОЗРОБКА ТА ОБҐРУНТУВАННЯ СИСТЕМИ ЗАХИЩЕНОГО ВІДДАЛЕНОГО КЕРУВАННЯ ЖИВЛЕННЯМ

2.1. Розробка та обґрунтування архітектури системи

З урахуванням вимог до системи захищеного віддаленого керування живленням телекомунікаційного обладнання в умовах критичних збоїв, архітектура повинна забезпечувати: автономність роботи, надійність прийняття рішень, можливість мережевої взаємодії та керування силовими елементами. Для досягнення цих цілей система реалізується як вбудований мікроконтролерний пристрій із підключенням до локальної мережі та силового кола живлення.

Вибір мікроконтролера обумовлюється достатньою обчислювальною потужністю для реалізації логіки системи, наявністю необхідних периферійних інтерфейсів та широкою підтримкою програмних бібліотек.

Переліченим вимогам задовольняє мікроконтролер STM32F411RE (ядро ARM Cortex-M4) (рис.2.1), який забезпечуватиме достатній рівень продуктивності для виконання задач моніторингу, обробки мережевих запитів та реалізації алгоритмів керування [5, 6].



Тактова частота – до 100 МГц;

Обсяг Flash-пам’яті – 512 кБ;

Оперативна пам’ять (SRAM) – 128 кБ;

Наявність апаратного модуля DMA;

Інтерфейси: SPI, I2C, USART, ADC, GPIO;

12-бітний АЦП із кількома каналами;

Підтримка режимів низького енергоспоживання

Рисунок 2.1. Основні характеристики мікроконтролера STM32F411RE та зовнішній вигляд плати розробника.

Оскільки STM32F411 не має вбудованого Ethernet-контролера, для реалізації мережевої взаємодії використовується зовнішній модуль W5500, який підключається через інтерфейс SPI [7].

Підтримка TCP/IP-стека на апаратному рівні;

до 8 одночасних сокетів;

Швидкість SPI до 80 МГц;

Вбудований буфер до 32 КБ;

Підтримка протоколів TCP, UDP, ICMP, ARP.

Рисунок 2.2. Зовнішній вигляд та основні характеристики модуля W5500

Використання апаратного TCP/IP-стека дозволяє значно зменшити навантаження на мікроконтролер та спростити реалізацію мережевих функцій, зокрема виконання ping-запитів та TCP-перевірок.

Для керування живленням обладнання використовується силовий ключ на базі твердотільного реле. У даній роботі розглядається варіант із реле, керованим через транзисторний ключ.

Застосування гальванічної розв’язки є обов’язковим для захисту мікроконтролера від високих напруг та імпульсних перешкод.

Контроль стану електроживлення реалізується за допомогою АЦП мікроконтролера. Для цього використовується схема подільника напруги, яка дозволяє вимірювати напругу основного та резервного живлення

Структурно система будується за принципом локального контролера з мінімальною залежністю від зовнішніх вузлів, що зображено в структурній схемі (додаток А).

Основні зв’язки:

- STM32 ↔ W5500 - через SPI;

- STM32 ↔ реле - через GPIO;
- STM32 ↔ датчики напруги - через ADC;
- STM32 ↔ сервер - через Ethernet (TCP/IP).

Сервер або VPN-шлюз виконує функції віддаленого доступу та може використовувати захищені протоколи, зокрема SSH, для взаємодії з мережею [3].

Обрана архітектура має такі переваги:

1. Автономність – система здатна працювати без постійного з'єднання із сервером.
2. Надійність – мінімізація складних залежностей і використання апаратного TCP/IP.
3. Гнучкість – можливість масштабування та модифікації окремих модулів.
4. Безпека – відсутність прямого доступу до пристрою з Інтернету, використання VPN.

Для коректної роботи мікроконтролерної системи необхідно забезпечити стабільне живлення незалежно від стану основного навантаження. У запропонованій архітектурі живлення контролера реалізується від окремого джерела, що дозволяє системі функціонувати навіть у випадку повного відключення керованого обладнання.

Живлення STM32F411RE здійснюється від стабілізованої напруги 3,3 В, яка формується за допомогою лінійного стабілізатора (AMS1117-3.3).

Основні вимоги до підсистеми живлення:

- стабільність напруги (відхилення не більше ±5%);
- низький рівень шумів;
- наявність фільтрації (LC або RC-фільтри);
- захист від перенапруги та переполюсування.

Використання окремого живлення для контролера забезпечує можливість виконання алгоритмів навіть у критичних ситуаціях, коли основне обладнання вже вимкнене.

Однією з ключових вимог до системи є своєчасне реагування на збої. Для цього визначаються такі часові характеристики:

- період моніторингу: 1–10 с;
- кількість повторних перевірок: 3–5 спроб;
- затримка перед перезавантаженням: 5–30 с;
- тривалість відключення живлення: 2–10 с;
- час очікування відновлення: 30–120 с.

Обчислювальної потужності STM32F411 достатньо для виконання цих задач у реальному часі, оскільки більшість операцій є дискретними та не потребують складних обчислень. Використання апаратних таймерів дозволяє реалізувати точне керування часовими інтервалами без навантаження на процесор [5].

Програмна частина системи реалізується за принципом циклічного опитування (superloop). Основні функціональні модулі програми:

- модуль мережевого моніторингу;
- модуль аналізу стану;
- модуль керування реле;
- модуль контролю живлення;
- модуль обробки команд.

Обмін даними між модулями здійснюється через спільні змінні. Для підвищення надійності роботи системи передбачено використання апаратних та програмних механізмів захисту.

До апаратних засобів належать:

- використання watchdog-таймера для автоматичного перезапуску контролера;
- гальванічна розв’язка силових ланцюгів;
- фільтрація живлення;
- захист GPIO від перенапруги.

До програмних засобів:

- перевірка коректності отриманих даних;
- використання тайм-аутів;
- обмеження частоти перезавантажень (anti-flap логіка);
- журналювання подій.

Увагу приділено запобіганню циклічним перезавантаженням, коли система постійно виконує restart через нестабільний стан. Для цього вводиться обмеження кількості перезавантажень за певний інтервал часу.

Вибір STM32F411 та W5500 є компромісом між складністю реалізації та функціональністю. Використання більш потужних мікроконтролерів із вбудованим Ethernet (наприклад STM32F7) дозволило б реалізувати більш складні протоколи, однак значно ускладнило б розробку та підвищило енергоспоживання.

З іншого боку, використання простіших мікроконтролерів (наприклад AVR) обмежило б можливості системи щодо обробки мережевих запитів та реалізації складних алгоритмів.

2.2. Розробка алгоритму функціонування системи

Ефективність запропонованої системи визначається не лише апаратною реалізацією, але й алгоритмами, які забезпечують своєчасне виявлення збоїв, прийняття рішень та керування живленням обладнання. Алгоритм роботи системи повинен враховувати можливість хибних спрацювань, забезпечувати стабільність функціонування та запобігати неконтрольованим перезавантаженням.

Алгоритм функціонування системи реалізується у вигляді циклічного процесу, що складається з таких основних етапів:

1. Ініціалізація системи;
2. моніторинг доступності обладнання;
3. аналіз результатів перевірки;
4. прийняття рішення;
5. виконання дій (перезавантаження або очікування);
6. контроль відновлення працездатності;
7. моніторинг стану живлення.

Контроль доступності реалізується комбінованим методом, що включає ICMP-перевірку та TCP-з’єднання (наприклад, порт 22 або 80).

Послідовність перевірки:

1. надсилання ICMP-запиту;
2. при відсутності відповіді – повторна перевірка;
3. після N невдалих спроб – перехід до TCP-перевірки;
4. при невдачі – фіксація збою.

Комбінування методів дозволяє зменшити ймовірність хибного визначення відмови [1].

Рішення про виконання перезавантаження приймається лише після підтвердження збою. Для цього вводиться логічна змінна стану системи:

- NORMAL – нормальна робота;
- WARNING – підозрілий стан;
- FAIL – підтверджений збій.

У стані WARNING система виконує додаткові перевірки. Якщо збій підтверджується, виконується перехід у стан FAIL і ініціюється процедура перезавантаження.

Процедура перезавантаження включає:

1. фіксацію події збою;
2. затримку перед дією (5–10 с);
3. вимкнення реле (відключення живлення);
4. витримку (2–5 с);
5. увімкнення живлення;
6. перехід у режим очікування.

Час очікування після включення становить 30–120 с і залежить від типу обладнання.

Такий алгоритм забезпечує повне перезавантаження пристрою та очищення його внутрішнього стану.

Після перезавантаження система переходить у режим перевірки:

- виконується періодичний моніторинг;
- визначається момент відновлення доступності;
- у разі успіху – повернення у стан NORMAL;

- у разі невдачі – повторна спроба (з обмеженням).

При зникненні основного живлення система переходить у режим контролю енергоресурсу.

Основні етапи:

1. фіксація переходу на резервне живлення;
2. періодичне вимірювання напруги АКБ;
3. оцінка залишкового ресурсу;
4. при досягненні порогу – ініціація shutdown.

У критичному режимі система формує команду на завершення роботи через мережевий інтерфейс (наприклад, через сервер із використанням SSH) [3].

Після завершення роботи відключається живлення та система переходить у режим очікування відновлення напруги.

Для формалізації роботи систему можна представити у вигляді автомата станів:

- INIT – ініціалізація;
- NORMAL – нормальна робота;
- CHECK – перевірка;
- FAIL – збій;
- RESTART – перезавантаження;
- RECOVERY – відновлення;
- POWER_SAVE – режим економії.

Такий підхід дозволяє чітко визначити логіку переходів.

Для підвищення точності роботи системи доцільно формалізувати процес прийняття рішень у вигляді логічної функції. Нехай:

- P_{ICMP} – результат ICMP-перевірки (1 — успіх, 0 — відмова);
- P_{TCP} - результат TCP-перевірки;
- P_{SNMP}- результат SNMP-запиту;
- N_{Fail} - кількість послідовних невдалих перевірок.

Тоді умова фіксації збою може бути записана як:

$$FAIL = (N\_fail \geq N\_threshold)\ AND\ (P\_tcp = 0) \quad (2.1)$$

де $N_{threshhold}$ =3…5

Додатково може враховуватись середній час відгуку:

Однією з типових проблем систем автоматичного керування є виникнення циклічних перезавантажень (flapping), коли пристрій постійно переходить у стан перезапуску через нестабільну роботу.

Для запобігання цьому вводиться механізм обмеження:

- лічильник перезавантажень R;
- часове вікно T=10–15хв;
- максимальна кількість перезапусків Rmax=3.

Умова блокування:

$$IF\ R \geq R\_max \rightarrow disable\ restart \tag{2.2}$$

У цьому випадку система переходить у режим аварійного блокування і очікує втручання оператора або зміни умов роботи.

У реальних умовах можливі ситуації, коли відсутність зв’язку викликана не відмовою обладнання, а проблемами мережі. Для цього алгоритм доповнюється перевірками:

- доступності шлюзу (gateway);
- доступності сервера моніторингу;
- аналізу ARP-відповідей.

Якщо недоступні всі вузли мережі, система не виконує перезавантаження, оскільки причина може бути зовнішньою.

Такий підхід дозволяє уникнути помилкових дій у випадку глобальних мережевих збоїв [1].

Для реалізації алгоритму в мікроконтролері використовуються апаратні таймери:

- Timer1 – основний цикл моніторингу;
- Timer2 – затримки перезавантаження;
- Watchdog – контроль зависання системи.

Події формуються у вигляді прапорців:

- flag_check – запуск перевірки;

- flag_fail – підтвердження збою;
- flag_restart – запуск перезавантаження;
- flag_power_low – низький рівень живлення.

Такий підхід дозволяє розділити логіку на незалежні частини та спростити програмну реалізацію [6].

У режимі резервного живлення реалізується окремий сценарій:

1. фіксація переходу на АКБ;
2. запуск таймера автономної роботи;
3. періодичний контроль напруги;
4. при досягненні порогу:
    - передача команди shutdown;
    - очікування підтвердження;
    - вимкнення живлення.

Якщо підтвердження не отримано, система виконує примусове відключення через заданий тайм-аут.

2.3. Розробка принципової схеми системи

Структурна та принципова схеми системи захищеного віддаленого керування живленням телекомунікаційного обладнання наведені у додатках А та Б відповідно. Система побудована за модульним принципом і складається з таких основних функціональних вузлів:

1. Вхідний вузол живлення та захисту;
2. Стабілізатори живлення 5 В і 3,3 В;
3. Мікроконтролерний вузол STM32F411RE;
4. Ethernet-інтерфейс на базі W5500;
5. Вузол вимірювання напруги;
6. Вузол контролю наявності основного живлення;
7. Силовий вузол керування навантаженням;
8. Вузол індикації станів;

Основним керуючим елементом є STM32F411RE. Його живлення здійснюється від стабілізованої напруги 3,3 В. Для зменшення шумів у колах живлення біля виводів живлення мікроконтролера встановлюються керамічні конденсатори 0,1 мкФ, а також електролітичний конденсатор більшої ємності 47 мкФ. Це забезпечує фільтрацію короткочасних імпульсних завад і стабільну роботу цифрової частини схеми.

Мережевий зв'язок реалізується за допомогою модуля W5500, який підключається до STM32 через інтерфейс SPI. Для обміну даними використовуються сигнали SCK, MISO, MOSI та CS. Додатково можуть використовуватись сигнали INT та RESET, які дозволяють мікроконтролеру обробляти події Ethernet-контролера та виконувати його апаратне перезавантаження. Використання W5500 є доцільним, оскільки даний контролер має апаратну підтримку TCP/IP-стека, що зменшує навантаження на STM32 і спрощує реалізацію мережевого моніторингу [7].

Для керування живленням телекомунікаційного обладнання використовується силовий комутаційний вузол. Застосовується електромеханічне реле з котушкою. Оскільки GPIO мікроконтролера не може безпосередньо керувати котушкою реле через обмеження по струму, між STM32 та реле встановлюється транзисторний ключ. Використаний NPN-транзистор типу 2N2222. У колі бази транзистора встановлюється резистор 4,7кОм, а паралельно котушці реле - захисний діод 1N4148, для гасіння ЕРС самоіндукції.

Комутаційні контакти реле підключаються послідовно в коло живлення керованого обладнання.

Блок контролю живлення реалізується за допомогою аналогових входів STM32. Оскільки АЦП мікроконтролера працює на 3,3 В, напруга резервного джерела або акумулятора подається на ADC через резистивний подільник. Для вимірювання напруги акумулятора до 15 В використовується подільник із резисторів 100 кОм і 27 кОм. Напруга на вході АЦП у такому випадку визначається за формулою:

$$U\_ADC = U\_IN \times R2 / (R1 + R2) \quad (2.3)$$

де U_{in} - вимірювана напруга акумулятора, *R1* - верхній резистор подільника, *R2* - нижній резистор подільника, U_{adc} - напруга на вході АЦП.

Для зменшення впливу шумів паралельно нижньому резистору подільника встановлено конденсатор 0,1 мкФ, який утворює простий RC-фільтр. Це дозволяє згладити короткочасні коливання напруги та підвищити стабільність вимірювань.

Окремо у схемі передбачається вузол індикації стану, підключений до GPIO через струмообмежувальні резистори. Один світлодіод сигналізує про наявність живлення, другий – про мережеве з'єднання, третій – про аварійний режим або виконання перезавантаження. Така індикація спрощує налагодження та експлуатацію пристрою.

Для налагодження та програмування мікроконтролера використовується інтерфейс SWD, який включає сигнали SWDIO, SWCLK, NRST, GND та 3,3 В. Додатково може бути виведений UART-інтерфейс для діагностичного обміну з ПК або сервісним терміналом. Через UART можна передавати повідомлення про поточний стан системи, результати перевірок та коди помилок.

Важливим елементом принципової схеми є захист. У вхідному колі живлення передбачено запобіжник, захист від переполюсування за допомогою діода а також TVS-діод для обмеження імпульсних перенапруг. У силовій частині необхідно забезпечити розділення низьковольтної цифрової частини та силових ланцюгів. Це підвищує безпеку роботи пристрою та зменшує ризик пошкодження мікроконтролера.

Вхідне живлення системи (блок 1 принципової схеми) подається через роз'єм J1, на який надходить напруга 12 В. Ця напруга використовується як основне джерело живлення для всієї схеми та силового релейного вузла [4].

У вхідному колі передбачено декілька захисних елементів:

- J1 – вхідний роз'єм живлення 12 В;
- F1 – запобіжник;
- D1 / TVS – захист від імпульсних перенапруг;
- Qp1 – захист від переполюсування;
- C1, C2 – вхідні фільтрувальні конденсатори.

TVS-діод D1 використовується для захисту від короткочасних імпульсних перенапруг, які можуть виникати при комутації живлення, роботі реле або підключенні зовнішнього джерела. При перевищенні допустимої напруги TVS-діод переходить у провідний стан і обмежує амплітуду імпульсу, захищаючи наступні каскади [8]. Для такого вузла використовується діод серії SMBJ, оскільки такі TVS-діоди призначені для захисту вразливих кіл живлення та інтерфейсів від імпульсних перенапруг [8].

Елемент Qp1 реалізує захист від неправильної полярності підключення живлення. У практичній схемі для цього використовується P-канальний MOSFET AO4407A [22].

Конденсатори C1 і C2 виконують фільтрацію вхідної напруги. Електролітичний конденсатор більшої ємності згладжує повільні коливання та короткочасні просідання напруги, а керамічний конденсатор 100 нФ пригнічує високочастотні завади.

Після вузла захисту формується лінія 12В., Вона подається на понижувальний стабілізатор, вузол вимірювання основного живлення та релейний блок.

Оскільки мікроконтролер і Ethernet-контролер не можуть живитися безпосередньо від 12 В, у схемі передбачено два ступені стабілізації (блок 2 принципової схеми):

12 В → 5 В → 3,3 В

Перший ступінь реалізовано на імпульсному понижувальному стабілізаторі LM2596S-5.0. Він формує стабілізовану напругу +5 В із вхідної напруги 12 В [9].

До складу цього вузла входять:

- U1 – LM2596S-5.0;
- L1 – дросель;
- D3 – діод Шотткі;
- C3, C4, C5 – фільтрувальні конденсатори;
- R1, R2 – елементи зворотного зв’язку або навантаження, залежно від варіанта стабілізатора.

Імпульсний стабілізатор обрано тому, що він має вищий ККД порівняно з лінійним стабілізатором. Якщо б напруга 12 В знижувалась до 5 В за допомогою лінійного стабілізатора, зайва потужність розсіювалася б у вигляді тепла. Це небажано для автономних або резервних систем живлення.

Дросель L1 і діод Шотткі D3 є типовими елементами імпульсного перетворювача. Вони забезпечують накопичення та передавання енергії на вихід стабілізатора. Конденсатори на виході згладжують пульсації та формують стабільну напругу +5 В[9].

Другий ступінь живлення реалізовано на лінійному стабілізаторі AMS1117-3.3, який формує напругу +3,3 В[10].

- U2 – AMS1117-3.3;
- C6, C7, C8 – вхідні та вихідні конденсатори;
- +3V3 – основна логічна напруга схеми.

Напруга 3,3 В використовується для живлення: STM32F411RE, W5500, логічних входів, світлодіодної індикації та підтягувальних резисторів.

Використання AMS1117-3.3 після імпульсного перетворювача є зручним, оскільки він додатково згладжує напругу та забезпечує стабільне живлення цифрової частини. При цьому падіння напруги з 5 В до 3,3 В є невеликим, тому втрати потужності залишаються прийнятними[10].

Основним керуючим елементом системи є мікроконтролер STM32F411RE (блок 3 принципової схеми). Він виконує функції локального контролера, який приймає рішення про стан системи та керує периферійними вузлами[5].

У схемі до мікроконтролера підключені:

- лінії живлення 3,3 В;
- конденсатори розв’язки;
- кварцовий резонатор;
- кнопка RESET;
- лінія BOOT0;
- SWD-роз’єм для програмування;
- UART-роз’єм для діагностики;

- ADC-входи для контролю напруги;
- SPI-інтерфейс для W5500;
- GPIO для керування реле;
- GPIO для світлодіодної індикації.

Живлення мікроконтролера здійснюється від лінії +3V3. Біля виводів живлення встановлені керамічні конденсатори 100 нФ, які виконують локальну фільтрацію. Такі конденсатори необхідні для стабільної роботи цифрових мікросхем, оскільки під час перемикання логічних елементів виникають короткочасні імпульси струму[5].

Для формування тактового сигналу використовується кварцовий резонатор Y1 на 8 МГц з двома конденсаторами по 22 пФ. На основі цього сигналу внутрішня PLL мікроконтролера може формувати системну частоту, наприклад 100 МГц. Це забезпечує достатню продуктивність для виконання алгоритмів моніторингу, обробки мережевих подій і керування живленням[5, 11, 12].

Кнопка RESET підключена до лінії NRST. Вона дозволяє вручну перезапустити мікроконтролер під час налагодження або аварійної ситуації. Резистор підтягування забезпечує стабільний логічний рівень на вході скидання.

Лінія BOOT0 через резистор підтягується до землі. Це означає, що після подачі живлення мікроконтролер запускається з основної Flash-пам’яті, де зберігається прошивка системи.

Призначення основних виводів STM32**:**

У схемі виводи мікроконтролера використовуються таким чином:

- PA0 – BAT_SENSE, вимірювання напруги акумулятора через ADC;
- PA1 – MAIN_SENSE, вимірювання напруги основного живлення через ADC;
- PA5 – RELAY_CTRL, керування силовим реле;
- PB3 – SPI_SCK, тактовий сигнал SPI для W5500;
- PB4 – SPI_MISO, прийом даних від W5500;
- PB5 – SPI_MOSI, передавання даних до W5500;
- PB6 – SPI_CS, вибір модуля W5500;

- PB7 – SPI_RST, апаратне скидання W5500;
- PB8 – W5500_INT, сигнал переривання від W5500;
- PC0 – STATUS_LED, індикація нормальної роботи;
- PC1 – FAULT_LED, індикація аварії;
- PC2 – BAT_LED, індикація роботи від АКБ;
- PA9 – UART_TX, передавання діагностичних повідомлень;
- PA10 – UART_RX, прийом службових команд.

Такий розподіл виводів дозволяє чітко розділити функції схеми: аналогові вимірювання виконуються через ADC-входи, мережева взаємодія - через SPI, а силове керування та індикація - через GPIO[5, 7, 11, 12].

Для підключення системи до локальної мережі використовується Ethernet-контролер W5500. Він має апаратну підтримку TCP/IP-стека, що значно спрощує роботу мікроконтролера (блок 4 принципової схеми).

До складу Ethernet-вузла входять:

- U4 – W5500;
- J4 – RJ-45 MagJack;
- Y2 – кварцовий резонатор;
- C16, C17, C18 – фільтрувальні та задавальні конденсатори;
- R8, R9, R10 – послідовні резистори в сигнальних лініях SPI;
- D4, D5, D6 – світлодіоди стану Ethernet.

W5500 підключається до STM32 через SPI:

- SPI_SCK – тактовий сигнал;
- SPI_MOSI – дані від STM32 до W5500;
- SPI_MISO – дані від W5500 до STM32;
- SPI_CS – вибір мікросхеми;
- SPI_RST – апаратне скидання;
- W5500_INT – сигнал переривання.

Роз’єм RJ-45 MagJack містить вбудовані трансформатори гальванічної розв’язки. Це важливо для Ethernet-інтерфейсу, оскільки фізичний рівень мережі

повинен бути відокремлений від логічної частини пристрою. Таке рішення підвищує завадостійкість та захищає схему від потенційних різниць потенціалів між мережевими пристроями[13].

У схемі передбачено вимірювання двох напруг (блок 5 принципової схеми):

- BATTERY_12V – напруга резервного джерела;
- MAIN_12V – напруга основного живлення.

Оскільки входи ADC мікроконтролера STM32 працюють у діапазоні 0–3,3 В, безпосередньо подавати 12В. на мікроконтролер не можна. Для цього використовуються резистивні подільники напруги[5].

Вузол вимірювання акумулятора містить:

- R5 – верхній резистор подільника;
- R6 – нижній резистор подільника;
- C20 – фільтрувальний конденсатор;
- BAT_SENSE – сигнал на ADC-вхід PA0.

Вузол вимірювання основного живлення містить:

- R7 – верхній резистор подільника;
- R17 – нижній резистор подільника;
- C21 – фільтрувальний конденсатор;
- MAIN_SENSE – сигнал на ADC-вхід PA1.

Типовий коефіцієнт подільника:

$$V\_SENSE = V\_IN \times R_{нижній} / (R_{верхній} + R_{нижній}) \quad (2.4)$$

Якщо використано резистори 100 кОм і 27 кОм, коефіцієнт становить приблизно:

$$27 / (100 + 27) \approx 0{,}212$$

Тобто при напрузі 12 В на вході ADC буде:

$$12 \times 0{,}212 \approx 2{,}54 \text{ В}$$

Це безпечно для мікроконтролера, оскільки не перевищує 3,3 В.

Конденсатори C20 і C21 разом із резисторами подільника утворюють простий RC-фільтр. Він згладжує короткочасні імпульси та зменшує похибку вимірювання.

На основі цих вимірювань мікроконтролер визначає:

- чи присутнє основне живлення;

- чи система перейшла на АКБ;

- чи напруга АКБ досягла попереджувального рівня;

- чи потрібно виконати безпечне вимкнення;

- чи слід відключити навантаження для захисту акумулятора.

Вузол контролю наявності основного живлення:

Окрім аналогового вимірювання, у схемі передбачено цифровий сигнал MAIN_OK, який показує факт наявності основного живлення (блок 6 принципової схеми).

Вузол побудований на компараторі LM393[14].

До нього входять:

- U5A – компаратор LM393;
- R19, R20 – подільник напруги основного живлення;
- R21 – підтягувальний резистор виходу компаратора;
- MAIN_OK – цифровий сигнал для мікроконтролера.

Принцип роботи такий: якщо напруга основного живлення перевищує заданий поріг, на виході компаратора формується логічний рівень, який мікроконтролер сприймає як наявність основного живлення.

Цей вузол корисний тому, що цифрова ознака наявності живлення може бути оброблена швидше і простіше, ніж аналогове значення ADC. Аналогове вимірювання дає точне значення напруги, а компаратор – швидку логічну ознаку

У програмній логіці це дозволяє швидко перейти в режим BATTERY_MODE при зникненні основного живлення[12, 14].

Силовий вузол призначений для апаратного перезавантаження телекомунікаційного обладнання (блок 7 принципової схеми). Він реалізований на реле K1, яким керує мікроконтролер через транзисторний ключ [6, 15, 16].

До вузла входять:

- K1 – реле 12 В;
- Q1 – транзистор 2N2222;
- R18 – резистор бази транзистора;

- D2 – захисний діод котушки реле;
- J5 – вихід на навантаження;
- RELAY_CTRL – керуючий сигнал від STM32.

Мікроконтролер не може безпосередньо живити котушку реле, оскільки GPIO-вивід має обмежений струм. Тому використовується транзистор Q1, який працює як ключ [5, 11, 15]. Транзистор 2N2222 за даташитом є NPN-транзистором, придатним для комутації невеликих струмів у керуючих колах, тому його можна використовувати для вмикання котушки реле через GPIO мікроконтролера [15].

Коли на сигнал RELAY_CTRL подається логічна одиниця, через резистор R18 відкривається транзистор Q1. Через котушку реле протікає струм, реле спрацьовує і змінює стан контактів. Через контакти реле подається або розривається живлення навантаження.

Захисний діод D2 підключений паралельно котушці реле. Він потрібен для гасіння імпульсу самоіндукції, який виникає при вимкненні котушки. Без цього діода імпульс напруги міг би пошкодити транзистор або мікроконтролер [6, 17, 18]. Для цієї функції використовуються діоди 1N4148 [17, 18].

Реле K1 реалізоване на базі модуля типу SRD-12VDC-SL-C. Такі реле застосовуються для комутації навантажень і мають контакти, розраховані на значно більші струми, ніж може забезпечити мікроконтролерний GPIO [16].

Роз’єм J5 використовується для підключення телекомунікаційного обладнання. Через нього система керує подачею живлення на зовнішній пристрій.

Функція цього вузла:

1. Cisco-пристрій не відповідає після 3 перевірок.
2. STM32 переходить у стан FAIL.
3. RELAY_CTRL перемикає реле.
4. Живлення Cisco вимикається на 7 с.
5. Живлення подається повторно.
6. Система очікує 120 с. на відновлення.
7. Якщо Cisco відповідає - повернення у NORMAL.

Такий підхід використовується як аварійний спосіб відновлення працездатності у випадках, коли штатне програмне керування через SSH або інші мережеві засоби недоступні [2, 19, 20].

Для візуального контролю роботи системи передбачено три світлодіоди (блок 8 принципової схеми):

- D7 – POWER, індикація живлення;
- D8 – NETWORK, індикація мережевої активності;
- D9 – ALARM, індикація аварійного режиму.

Кожен світлодіод підключений через струмообмежувальний резистор.

Індикація дозволяє швидко оцінити стан системи без підключення до UART або веб-інтерфейсу.

Загальний принцип роботи схеми:

Після подачі живлення система проходить кілька етапів.

Спочатку через вхідний вузол захисту формується лінія +12V_RAW. Далі імпульсний стабілізатор LM2596S-5.0 формує +5 В., а лінійний стабілізатор AMS1117-3.3 - +3,3 В. Після стабілізації живлення запускається мікроконтролер STM32F411RE [9, 10, 11].

Мікроконтролер виконує ініціалізацію:

- GPIO;
- ADC;
- SPI;
- UART;
- W5500;
- таймерів;
- watchdog;
- початкових станів системи.

Після цього Ethernet-контролер W5500 підключається до локальної мережі. Система отримує або використовує задану IP-адресу та переходить у режим моніторингу [7, 12].

У нормальному режимі STM32 періодично перевіряє:

- наявність основного живлення;
- напругу резервного акумулятора;
- доступність Cisco-пристрою;
- стан власного Ethernet-з’єднання.

Мережевий контроль доступності Cisco-пристрою може виконуватись за допомогою TCP/IP-перевірок, SSH-доступу через сервер керування або SNMP-моніторингу [1, 7, 19, 21].

Якщо Cisco-пристрій відповідає, система залишається у стані NORMAL. Реле утримує живлення навантаження увімкненим, а система лише передає статус на сервер або виводить його через UART.

Якщо Cisco-пристрій перестає відповідати, контролер не виконує перезавантаження одразу. Спочатку збільшується лічильник невдалих перевірок. Лише після досягнення встановленого порогу, наприклад трьох невдалих перевірок, система переходить у стан FAIL [6, 12].

Після підтвердження збою перевіряються додаткові умови:

- чи дозволений автоматичний перезапуск;
- чи не активовано режим обслуговування;
- чи не перевищено ліміт перезапусків;
- чи не перебуває акумулятор у критичному стані.

Якщо всі умови виконані, мікроконтролер активує реле і виконує power cycle. Живлення навантаження відключається на заданий час, після чого подається знову. Далі система переходить у режим RECOVERY і очікує, поки обладнання завантажиться.

Якщо після відновлення Cisco знову відповідає, система повертається у стан NORMAL. Якщо ж пристрій не відновлюється після кількох спроб, активується стан LOCKOUT, у якому автоматичні перезапуски блокуються [6, 12].

Робота при зникненні основного живлення:

При зникненні основного живлення система визначає це за сигналом MAIN_OK або за напругою MAIN_SENSE. Після цього контролер переходить у режим BATTERY_MODE

У цьому режимі система продовжує працювати від резервного джерела та контролює напругу АКБ через ADC [4, 5, 11].

Можливі стани:

- АКБ у нормі – система продовжує роботу;
- напруга знижена – формується попередження;
- напруга критична – запускається safe shutdown;
- напруга нижче допустимої – навантаження вимикається.

При критичному рівні заряду автоматичне перезавантаження блокується. Це важливо, тому що перезапуск обладнання в умовах слабкого акумулятора може призвести до додаткового споживання та повного знеструмлення [4].

Якщо сервер доступний, система може надіслати команду про необхідність безпечного вимкнення. Для Cisco-пристрою це може означати збереження конфігурації, а для сервера — завершення роботи операційної системи [19, 20].

Після завершення тайм-ауту STM32 вимикає живлення навантаження через реле і переходить у стан POWER_SAVE [12].

Коли основне живлення з’являється знову, система не одразу вмикає навантаження. Спочатку запускається таймер стабілізації, наприклад 30 секунд. Це потрібно для захисту від ситуацій, коли живлення з’являється короткочасно і знову зникає [4, 6].

Після стабілізації контролер повторно перевіряє наявність живлення. Якщо воно стабільне, реле подає живлення на телекомунікаційне обладнання, система переходить у стан RECOVERY, а потім - у NORMAL [12].

2.4. Особливості підключення та взаємодії з обладнанням Cisco

У запропонованій системі телекомунікаційне обладнання Cisco розглядається як один із основних об’єктів моніторингу та керування. Це може бути

маршрутизатор, комутатор або інший мережевий пристрій, від стабільної роботи якого залежить доступність окремого сегмента мережі. Особливістю такого обладнання є наявність декількох способів адміністрування: через консольний порт, захищене мережеве підключення, протоколи моніторингу та системи централізованого керування.

У межах даної роботи розглядається не повна заміна штатних засобів адміністрування Cisco, а створення допоміжної системи, яка забезпечує контроль доступності пристрою та можливість апаратного відновлення його працездатності у випадку критичного збою. Такий підхід є практичним, оскільки при повному зависанні операційної системи мережеві методи керування можуть бути недоступні, тоді як керування живленням залишається універсальним способом відновлення роботи.

Підключення до обладнання Cisco може здійснюватися двома основними шляхами: через мережевий інтерфейс Ethernet та через консольний порт. Основним способом у запропонованій системі є підключення через локальну мережу Ethernet, оскільки воно дозволяє виконувати моніторинг доступності, перевірку роботи сервісів і взаємодію із сервером керування. Для первинного налаштування або аварійного доступу може використовуватись консольний порт Cisco, який у багатьох моделях реалізований у вигляді RJ-45 console port або USB-console. У документації Cisco зазначається, що RJ-45 console port використовується для підключення до пристрою через консольний кабель, а типова швидкість роботи консольного інтерфейсу для багатьох пристроїв становить 9600 біт/с [22].

У випадку використання консольного підключення необхідно враховувати електричну несумісність рівнів сигналів. UART мікроконтролера STM32 працює з логічними рівнями 3,3 В, тоді як класичний консольний інтерфейс Cisco використовує рівні RS-232. Тому пряме підключення STM32 до консольного порту є неприпустимим. Для такого варіанта необхідно застосовувати перетворювач рівнів, наприклад MAX3232, або USB-UART/console-рішення з відповідною гальванічною розв’язкою. У межах даної системи консольний порт доцільно

розглядати як резервний канал налаштування, а не як основний канал автоматичного керування.

Основним каналом взаємодії з Cisco в розроблюваній системі є мережевий доступ через локальну мережу. Для цього обладнання Cisco повинно мати налаштовану IP-адресу керування, доступну з мережевого сегмента, у якому працює сервер моніторингу або мікроконтролерний пристрій. При цьому не рекомендується відкривати інтерфейси керування безпосередньо в Інтернет. Доступ адміністратора повинен здійснюватися через VPN-шлюз, після чого пристрій стає доступним за внутрішньою адресою локальної мережі.

Для захищеного адміністрування обладнання Cisco доцільно використовувати SSH. На відміну від Telnet, SSH забезпечує шифрування переданих даних та автентифікацію користувача. Для увімкнення SSH на Cisco IOS необхідно задати ім’я пристрою, доменне ім’я, згенерувати криптографічний ключ і дозволити SSH-доступ на VTY-лініях [19].

У запропонованій архітектурі SSH використовується не стільки самим STM32, скільки сервером керування. Це пояснюється тим, що реалізація повноцінного SSH-клієнта на STM32F411RE є складною через обмежені ресурси мікроконтролера та необхідність підтримки криптографічних алгоритмів. Тому доцільно розподілити функції таким чином: STM32 виконує локальний моніторинг і керування живленням, а сервер керування виконує складні мережеві операції, зокрема SSH-команди до Cisco. Такий підхід підвищує надійність та спрощує програмну реалізацію вбудованої частини.

Для моніторингу стану обладнання Cisco доцільно застосовувати кілька методів. Найпростішим є ICMP-перевірка доступності, яка дозволяє визначити, чи відповідає пристрій у мережі. Додатково може використовуватись TCP-перевірка доступності керуючого порту, наприклад 22 для SSH. Для більш детального контролю параметрів роботи використовується SNMP, який дозволяє отримувати інформацію про завантаження процесора, стан інтерфейсів, час безперервної роботи пристрою та інші параметри [23]. При цьому перевагу слід надавати

SNMPv3, оскільки він підтримує автентифікацію та шифрування, на відміну від SNMPv1/SNMPv2c, де використовуються community strings.

У практичній реалізації система працює за таким сценарієм: У нормальному режимі контролер або сервер періодично перевіряє доступність Cisco-пристрою за допомогою ICMP та TCP. Якщо пристрій не відповідає, виконується повторна перевірка через заданий інтервал. Якщо відмова підтверджується, сервер може спробувати виконати програмне перезавантаження через SSH. Якщо SSH-доступ недоступний, система переходить до апаратного перезавантаження через силовий комутаційний блок. Такий порядок дій є більш безпечним, ніж негайне вимкнення живлення, оскільки спочатку використовується штатний спосіб керування, а апаратне втручання застосовується лише у випадку його неефективності.

Окремо слід врахувати особливості команди перезавантаження Cisco. У Cisco IOS для перезавантаження може використовуватися команда reload, яка зазвичай потребує підтвердження. Саме тому її автоматичне виконання з боку мікроконтролера є незручним. Натомість сервер керування може використовувати скрипт, який виконує SSH-підключення, передає команду та обробляє підтвердження. У випадку відсутності відповіді від пристрою сервер передає команду STM32 на виконання power cycle.

Важливою вимогою є розділення доступу до мережі керування та основної користувацької мережі. Для цього доцільно використовувати окремий VLAN або окрему підмережу керування. У такій архітектурі STM32, сервер моніторингу, VPN-шлюз та керуючі інтерфейси Cisco розміщуються в ізольованому сегменті. Це зменшує ризик несанкціонованого доступу до системи керування живленням і відповідає загальним принципам побудови захищених телекомунікаційних систем [2, 3].

З точки зору електроживлення Cisco-пристрій підключається до силового комутаційного блоку як навантаження. При цьому необхідно враховувати споживану потужність пристрою, пускові струми та допустимий струм контактів реле або твердотільного реле. Для мережевого обладнання з живленням 220 В AC доцільно використовувати комутаційний елемент із запасом по струму не менше

30–50 %. Якщо обладнання живиться від постійної напруги 12 В, 24 В або 48 В, схема комутації може бути реалізована на MOSFET-ключі або DC-реле відповідного номіналу.

У випадку роботи від резервного живлення система повинна коректно взаємодіяти з Cisco-пристроєм до моменту його вимкнення. Якщо рівень заряду акумулятора знижується до критичного значення, сервер або контролер формує команду безпечного завершення роботи. Для маршрутизаторів і комутаторів Cisco це може бути не повноцінний shutdown у сенсі серверної операційної системи, а збереження поточної конфігурації, завершення активних адміністративних операцій та подальше контрольоване відключення живлення. Для серверів, що також можуть входити до складу телекомунікаційного вузла, виконується стандартна команда завершення роботи операційної системи.

Таким чином, взаємодія з обладнанням Cisco у запропонованій системі будується за комбінованим принципом. Для моніторингу використовуються ICMP, TCP та SNMP, для захищеного адміністрування - SSH через ізольований канал доступу, а для аварійного відновлення працездатності - апаратне керування живленням. Така схема дозволяє поєднати штатні методи адміністрування з незалежним механізмом відновлення у випадку критичного збою.

ІІІ. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ПЕРЕВІРКА РОБОТИ СИСТЕМИ

3.1. Структура програмного забезпечення мікроконтролера

Програмне забезпечення мікроконтролерної системи є основою роботи пристрою захищеного віддаленого керування живленням телекомунікаційного обладнання. Якщо принципова схема визначає фізичну взаємодію між вузлами, то програмна частина задає послідовність дій системи: ініціалізацію периферії, моніторинг стану обладнання, контроль основного та резервного живлення, обмін із сервером керування, керування реле та виконання аварійних алгоритмів.

Відповідно до розробленої принципової схеми (додаток Б), центральним елементом програмної реалізації є мікроконтролер STM32F411RE. Він виконує роль локального автономного контролера, який може приймати рішення без постійної участі сервера керування. Такий підхід є важливим для системи критичного керування живленням, оскільки у випадку втрати зв’язку із сервером або часткової деградації мережі пристрій повинен зберігати здатність виконувати базові функції: контроль напруги, аналіз доступності обладнання та керування силовим комутаційним вузлом.

Програмне забезпечення розроблюється за модульним принципом. Це дозволяє розділити функції системи на окремі логічні блоки, спростити налагодження та забезпечити можливість подальшого розширення. Відповідно до принципової схеми, доцільно виділити такі основні програмні модулі:

– модуль ініціалізації апаратної частини;

– модуль роботи з Ethernet-контролером W5500;

– модуль мережевого моніторингу Cisco-пристрою;

– модуль обміну із сервером керування;

– модуль вимірювання напруги через ADC;

– модуль контролю наявності основного живлення;

– модуль керування реле;

– модуль автомата станів;

– модуль захисту від циклічних перезапусків;

– модуль UART-діагностики;

– модуль обробки режиму обслуговування.

Модуль ініціалізації апаратної частини виконується після подачі живлення або скидання мікроконтролера. На цьому етапі налаштовуються тактова система, порти введення-виведення, ADC, SPI, UART, таймери та watchdog. Оскільки у схемі використовується кварцовий резонатор Y1 на 8 МГц, програмна конфігурація тактування повинна передбачати запуск системної частоти через PLL. Для STM32F411RE можлива робота на частоті до 100 МГц, що забезпечує достатню швидкодію для виконання задач моніторингу та керування [5, 11].

Після запуску тактування налаштовуються виводи мікроконтролера відповідно до їх функціонального призначення у схемі. Аналогові входи PA0 та PA1 використовуються для вимірювання напруги акумулятора та основного живлення. Вивід PA5 використовується як цифровий вихід RELAY_CTRL для керування транзисторним ключем реле. Виводи PB3, PB4, PB5, PB6, PB7 та PB8 використовуються для взаємодії з Ethernet-контролером W5500 через SPI, сигнал скидання та лінію переривання. Виводи PA9 і PA10 використовуються для UART-діагностики, а SWD-лінії — для програмування та налагодження мікроконтролера [5, 11, 12]

Узагальнене призначення основних сигналів програмної частини наведено нижче:

- PA0 – BAT_SENSE, ADC-вимірювання напруги резервного джерела;
- PA1 – MAIN_SENSE, ADC-вимірювання напруги основного живлення;
- PA5 – RELAY_CTRL, керування силовим реле;
- PB3 – SPI_SCK, тактовий сигнал SPI для W5500;
- PB4 – SPI_MISO, прийом даних від W5500;
- PB5 – SPI_MOSI, передавання даних до W5500;
- PB6 – SPI_CS, вибір Ethernet-контролера;

- PB7 – SPI_RST, апаратне скидання W5500;
- PB8 – W5500_INT, сигнал переривання від W5500;
- PA9 – UART_TX, передавання діагностичних повідомлень;
- PA10 – UART_RX, прийом службових команд;
- NRST – апаратне скидання мікроконтролера;
- SWDIO/SWCLK – програмування та налагодження.

Модуль роботи з Ethernet-контролером W5500 забезпечує підключення системи до локальної мережі керування. Відповідно до принципової схеми, W5500 підключений до STM32 через SPI-інтерфейс, що дозволяє мікроконтролеру передавати та приймати мережеві дані без реалізації повного програмного TCP/IP-стека. Особливість W5500 полягає в тому, що він містить апаратний TCP/IP-стек, Ethernet MAC і PHY, а взаємодія із зовнішнім контролером здійснюється через SPI [7]. Це зменшує навантаження на STM32F411RE та спрощує реалізацію мережевої частини системи.

Під час ініціалізації W5500 програмно задаються такі параметри:

- MAC-адреса пристрою;
- IP-адреса контролера;
- маска підмережі;
- адреса шлюзу;
- IP-адреса сервера керування;
- IP-адреса Cisco-пристрою;
- порти для TCP-перевірки та обміну із сервером.

Для системи такого типу доцільно використовувати статичну IP-адресу. Це спрощує адміністрування, дозволяє точно вказати адресу пристрою в мережі керування та уникнути залежності від DHCP-сервера. Наприклад, контролер може мати адресу 192.168.1.50, сервер керування - 192.168.1.100, а Cisco-пристрій - 192.168.1.10. У реальній мережі ці значення мають відповідати адресному плану службового VLAN або окремого сегмента керування.

Модуль мережевого моніторингу виконує перевірку доступності Cisco-пристрою. У програмній реалізації доцільно використовувати не одну перевірку, а

комбіновану логіку. Найпростішим варіантом є TCP-перевірка доступності порту 22, який використовується для SSH-доступу. Якщо Cisco-пристрій відповідає на TCP-з'єднання, система вважає, що обладнання доступне для керування. Якщо пристрій не відповідає, збільшується лічильник невдалих перевірок. Проте рішення про аварійний стан не приймається одразу, оскільки одинична невдала перевірка може бути наслідком короткочасної мережевої затримки [1, 2, 19].

Для зменшення ймовірності хибного спрацювання в програмі вводиться параметр MAX_FAIL_COUNT.

Такий підхід дозволяє не виконувати перезавантаження обладнання при одиничній втраті пакета або короткочасному перевантаженні мережі. Якщо ж Cisco-пристрій не відповідає протягом декількох послідовних циклів перевірки, система переходить до аварійного сценарію.

Модуль обміну із сервером керування використовується для передавання поточного стану системи та приймання команд адміністратора. Пристрій може передавати на сервер такі параметри:

- поточний стан системи;
- стан Cisco-пристрою;
- напругу резервного джерела;
- наявність основного живлення;
- стан реле;
- кількість виконаних перезапусків;
- ознаку аварійного режиму;
- ознаку режиму обслуговування.

Сервер керування, у свою чергу, може передавати команди:

- RESTART - виконати апаратний power cycle;
- MAINTENANCE_ON - увімкнути режим обслуговування;
- MAINTENANCE_OFF - вимкнути режим обслуговування;
- DISABLE_AUTO - заборонити автоматичні перезапуски;
- ENABLE_AUTO - дозволити автоматичні перезапуски;
- SHUTDOWN_LOAD - вимкнути навантаження.

Оскільки команда керування живленням є критичною, програмна частина повинна передбачати перевірку джерела команди. У спрощеному варіанті це може бути перевірка IP-адреси сервера та службового токена. У більш захищеній архітектурі обмін між сервером і контролером виконується всередині ізольованої мережі керування, доступ до якої здійснюється через VPN. Такий підхід відповідає загальній логіці захищеного віддаленого керування, де відкритий доступ до пристрою керування живленням із глобальної мережі є небажаним [3, 19, 21].

Для підвищення стабільності результату програмно використовується усереднення декількох вимірювань.

Модуль контролю основного живлення використовує два джерела інформації: аналогове значення MAIN_SENSE та цифровий сигнал MAIN_OK від компаратора LM393. Аналогове вимірювання дозволяє оцінити фактичний рівень напруги, а цифровий сигнал MAIN_OK швидко показує факт наявності або втрати основного живлення. Компаратор LM393 призначений для порівняння аналогових напруг і формування логічного сигналу на виході, тому його використання у вузлі швидкого контролю основного живлення є доцільним [14].

Програмно стан живлення поданий як перелік режимів:

- POWER_MAIN – основне живлення присутнє;
- POWER_BATTERY – робота від резервного джерела;
- POWER_BATTERY_WARNING – знижений рівень заряду;
- POWER_BATTERY_CRITICAL – критичний рівень заряду;
- POWER_CUTOFF - необхідне відключення навантаження.

Для 12-вольтового резервного джерела у модельній системі використовуються такі порогові значення:

- $U > 12{,}0$ В. – нормальний стан;
- 11,5–12,0 В. – робота від резервного живлення;
- 10,8–11,5 В. – попереджувальний рівень;
- 10,5–10,8 В. – критичний рівень;
- $U \leq 10{,}5$ В. – відключення навантаження.

Ці значення не є універсальними і мають уточнюватися залежно від типу акумулятора та характеристик UPS. Проте вони дозволяють продемонструвати принцип роботи алгоритму контролю резервного живлення. У системах безперебійного живлення важливо не доводити акумулятор до глибокого розряду, оскільки це зменшує його ресурс і може призвести до аварійного вимкнення обладнання [4].

Модуль керування реле реалізує керування силовим комутаційним вузлом. Відповідно до принципової схеми, GPIO-вихід STM32 не керує реле безпосередньо, а подає сигнал RELAY_CTRL на транзисторний ключ Q1. У такій схемі транзистор 2N2222 працює як ключ, що дозволяє комутувати струм котушки реле, який значно перевищує допустимий струм GPIO мікроконтролера [11, 15]. Паралельно котушці реле встановлено захисний діод, який гасить імпульс самоіндукції при вимкненні котушки [17, 18].

Програмно стан реле можна описати двома режимами:

- RELAY_ON – живлення на навантаження подається;
- RELAY_OFF – живлення навантаження відключене.

Під час запуску системи необхідно встановити безпечний початковий стан реле. Для телекомунікаційного обладнання доцільною є логіка, за якої перезапуск самого мікроконтролера не призводить до необґрунтованого відключення навантаження. Тому після ініціалізації GPIO програма повинна встановити вихід керування реле у стан, що відповідає нормальній подачі живлення на обладнання.

Основною дією модуля реле є процедура power cycle. Вона включає:

1. перевірку дозволу на перезапуск;
2. фіксацію аварійної події;
3. вимкнення реле;
4. витримку паузи;
5. повторне ввімкнення реле;
6. запуск таймера відновлення;
7. перехід у стан RECOVERY.

У модельній реалізації живлення може вимикатися на 7 секунд, після чого система очікує приблизно 120 секунд для завантаження Cisco-пристрою. Таке очікування необхідне, оскільки після подачі живлення мережевому обладнанню потрібен час для старту операційної системи, ініціалізації інтерфейсів та відновлення доступності керуючої IP-адреси [2, 20].

Центральним елементом програмного забезпечення є автомат станів. Він забезпечує контрольований перехід між режимами роботи системи та не дозволяє виконувати небезпечні дії без перевірки умов. Основні стани системи можна подати так:

- INIT – ініціалізація системи;
- NORMAL – нормальна робота;
- CHECK – додаткова перевірка доступності;
- FAIL – підтверджений збій;
- RESTART – виконання power cycle;
- RECOVERY – очікування відновлення обладнання;
- BATTERY_MODE – робота від резервного живлення;
- SHUTDOWN – підготовка до безпечного вимкнення;
- POWER_SAVE – навантаження вимкнено, контролер очікує повернення живлення;
- MAIN_POWER_RECOVERY – перевірка стабільності основного живлення;
- LOCKOUT – блокування повторних автоматичних перезапусків.

У стані INIT програма налаштовує периферію та перевіряє готовність основних вузлів. Після успішної ініціалізації система переходить у NORMAL. У цьому стані регулярно виконуються перевірки Cisco-пристрою, сервера керування та напруги живлення. Якщо Cisco-пристрій недоступний, система переходить до додаткових перевірок. Після досягнення порогу помилок формується стан FAIL, після чого за умови дозволу запускається RESTART.

Стан RECOVERY використовується після повторної подачі живлення. У цьому стані система не виконує нових перезапусків, а очікує завершення

завантаження обладнання. Після завершення таймера відновлення мережевий моніторинг перевіряє, чи знову доступний Cisco-пристрій. Якщо доступність відновлена, система повертається у NORMAL. Якщо ні — може бути виконана повторна спроба або активований стан LOCKOUT.

Стан LOCKOUT потрібен для захисту від циклічних перезапусків. Якщо обладнання не відновлюється після кількох power cycle, подальше багаторазове вмикання та вимикання може бути шкідливим. Тому після перевищення ліміту, наприклад трьох перезапусків за 15 хвилин, система блокує автоматичні перезапуски до втручання адміністратора. Така логіка відповідає загальним принципам побудови надійних вбудованих систем, у яких аварійні дії повинні бути обмежені та контрольовані [6].

У режимі BATTERY_MODE пріоритет роботи системи змінюється. Якщо основне живлення зникає, система продовжує працювати від резервного джерела, але починає частіше контролювати напругу акумулятора. При переході до попереджувального рівня формується повідомлення серверу. При досягненні критичного рівня автоматичні перезапуски блокуються, а система переходить у SHUTDOWN. Це необхідно, оскільки при низькому заряді акумулятора виконання power cycle може призвести до додаткового навантаження на джерело живлення та некоректного запуску обладнання [4].

У стані SHUTDOWN система ініціює безпечне вимкнення. Якщо сервер керування доступний, йому передається повідомлення про критичний рівень заряду. Для Cisco-пристрою це може означати спробу збереження конфігурації через SSH, а для сервера — запуск процедури завершення роботи операційної системи [19, 20]. Після завершення тайм-ауту мікроконтролер вимикає живлення навантаження через реле та переходить у стан POWER_SAVE.

Після повернення основного живлення система не повинна одразу подавати живлення на навантаження. Для цього використовується стан MAIN_POWER_RECOVERY, у якому контролер очікує стабілізації напруги протягом 30 секунд. Якщо живлення залишається стабільним, реле вмикає навантаження, система переходить у RECOVERY, а потім – у NORMAL.

Головний цикл програми повинен бути організований так, щоб уникати довгих блокуючих затримок. Замість використання тривалих delay-функцій доцільно використовувати таймери та періодичні задачі. Це дозволяє системі одночасно обслуговувати мережевий модуль, оновлювати watchdog, вимірювати напругу та контролювати стан реле.

Узагальнена логіка основного циклу:

```
    while (1)
{
update_timers();
read_adc_values();
update_power_state();
network_service();
monitor_cisco_device();
process_server_commands();
process_state_machine();
power_control_task();
refresh_watchdog();
}
```

Функція *update_timers()* відповідає за зменшення програмних таймерів. Функція *read_adc_values()* виконує вимірювання напруги акумулятора та основного живлення. Функція *update_power_state()* визначає поточний режим живлення. Функція *network_service()* обслуговує W5500. Функція *monitor_cisco_device()* перевіряє доступність Cisco-пристрою. Функція *process_server_commands()* обробляє команди від сервера. Функція *process_state_machine()* реалізує автомат станів, а *power_control_task()* керує реле та таймерами power cycle. Наприкінці циклу виконується оновлення watchdog-таймера.

Використання watchdog-таймера є важливим для підвищення надійності роботи прошивки. Якщо програма зависне або перестане виконувати головний цикл, watchdog перезапустить мікроконтролер. При цьому важливо, щоб перезапуск самого STM32 не призводив до неконтрольованого перезапуску

телекомунікаційного обладнання. Тому початковий стан реле та логіка відновлення після reset мають бути визначені явно.

Для налагодження та перевірки роботи системи використовується UART DEBUG. Через нього виводяться службові повідомлення про запуск системи, ініціалізацію W5500, IP-адресу, результати перевірки Cisco, напругу акумулятора, стан основного живлення, запуск power cycle, перехід у battery mode та виконання shutdown. Наявність UART-логування значно спрощує тестування, оскільки дозволяє бачити внутрішні стани системи без підключення складного програмного налагоджувача.

Програмна структура реалізована у вигляді окремих файлів:

- main.c – головний цикл програми;
- config.h – основні параметри системи;
- app_types.h – типи станів і структур;
- system_state.c – автомат станів;
- network_w5500.c – робота з Ethernet-контролером;
- monitoring.c – перевірка доступності Cisco;
- adc_measure.c – вимірювання напруги;
- power_control.c – керування реле та power cycle;
- commands.c – обробка команд від сервера;
- logger.c – UART-логування.

Такий поділ коду відповідає модульній структурі принципової схеми. Окремі апаратні вузли мають відповідні програмні модулі: Ethernet-вузол обслуговується модулем network_w5500.c, вузол ADC – adc_measure.c, силовий релейний вузол – power_control.c, а загальна логіка роботи системи – system_state.c.

Таким чином, структура програмного забезпечення мікроконтролера безпосередньо відповідає розробленій принциповій схемі. STM32F411RE виконує роль центрального керуючого вузла, W5500 забезпечує мережеву взаємодію, ADC-входи використовуються для контролю основного та резервного живлення, GPIO-вихід керує силовим реле, а UART і SWD забезпечують налагодження та

обслуговування. Запропонована програмна структура дозволяє реалізувати автономну, контрольовану та захищену систему керування живленням телекомунікаційного обладнання в умовах критичних збоїв.

3.2. Реалізація керування живленням і контролю резервного живлення

Керування живленням є основною виконавчою функцією розробленої системи. Якщо мережевий моніторинг визначає факт збою або нестабільного стану телекомунікаційного обладнання, то модуль керування живленням забезпечує практичну реакцію на цей стан: виконання апаратного перезавантаження, вимкнення навантаження при критичному розряді резервного джерела та відновлення роботи після повернення основного живлення.

Відповідно до розробленої принципової схеми, керування живленням здійснюється мікроконтролером STM32F411RE через цифровий вихід RELAY_CTRL. Цей сигнал формується GPIO-виводом мікроконтролера і подається на транзисторний ключ Q1, реалізований на базі NPN-транзистора 2N2222. Транзистор, у свою чергу, керує котушкою реле K1, через контакти якого комутується живлення зовнішнього телекомунікаційного обладнання [11, 15, 26].

GPIO-вивід мікроконтролера не може безпосередньо живити котушку реле, оскільки допустимий струм виходу STM32 є значно меншим за струм, необхідний для спрацювання реле. Саме тому між мікроконтролером і реле використовується транзисторний ключ. У такій схемі STM32 формує лише малопотужний логічний сигнал, а основний струм котушки реле проходить через транзистор Q1 [11, 15].

Програмна логіка керування реле повинна враховувати два базові стани:

RELAY_ON – живлення на телекомунікаційне обладнання подається;

RELAY_OFF – живлення телекомунікаційного обладнання відключене.

У нормальному режимі роботи реле перебуває у стані RELAY_ON, тобто живлення на обладнання подається постійно. Система не втручається в роботу навантаження, доки не буде підтверджено збій, критичний стан резервного живлення або команда адміністратора. Такий підхід важливий з погляду надійності:

пристрій керування не повинен створювати зайві перемикання живлення, якщо обладнання працює стабільно.

Під час запуску мікроконтролера необхідно правильно встановити початковий стан GPIO-виходу RELAY_CTRL. Якщо після reset вихід мікроконтролера короткочасно перебуватиме у невизначеному стані, це може призвести до небажаного перемикання реле. Тому в програмній ініціалізації вихід керування реле налаштовується одним із перших, після чого йому одразу задається безпечний стан. Для даної системи безпечним вважається стан, у якому живлення навантаження подається, а перезапуск самого STM32 не викликає автоматичного вимкнення Cisco-пристрою або сервера [5, 6, 11].

Функція керування реле:

```
void relay_set(relay_state_t state)
{
 if (state == RELAY_ON)
{
HAL_GPIO_WritePin(RELAY_CTRL_GPIO_Port, RELAY_CTRL_Pin,
GPIO_PIN_SET);
 }
 else
{
HAL_GPIO_WritePin(RELAY_CTRL_GPIO_Port, RELAY_CTRL_Pin,
GPIO_PIN_RESET);
}
```

Основною функцією модуля керування живленням є процедура power cycle, тобто контрольоване вимкнення живлення навантаження на заданий проміжок часу з подальшим повторним увімкненням. Така процедура використовується для відновлення працездатності обладнання у випадках, коли воно перестає відповідати на мережеві запити і штатні програмні засоби керування недоступні [2, 19, 20].

Послідовність процедури power cycle має такий вигляд:

1. Отримання підтвердженого стану FAIL від автомата станів.
2. Перевірка дозволу на перезапуск.
3. Фіксація події у журналі.
4. Вимкнення реле та відключення живлення навантаження.
5. Витримка паузи POWER_OFF_TIME_MS.
6. Повторне ввімкнення реле.
7. Запуск таймера RECOVERY_TIME_MS.
8. Перехід системи у стан RECOVERY.

У модельній реалізації для системи було прийнято такі часові параметри:

- POWER_OFF_TIME_MS = 7000 мс;
- RECOVERY_TIME_MS = 120000 мс;
- MAX_RESTART_COUNT = 3;
- RESTART_WINDOW_MS = 900000 мс.

Параметр POWER_OFF_TIME_MS визначає тривалість відключення живлення. Значення 7 секунд обране для того, щоб забезпечити повне знеструмлення внутрішніх ланцюгів обладнання. Занадто коротке вимкнення може не призвести до повного скидання пристрою, оскільки частина внутрішніх ємностей у блоці живлення або на платі обладнання може залишатися зарядженою.

Параметр RECOVERY_TIME_MS визначає час очікування після повторної подачі живлення. Для телекомунікаційного обладнання Cisco цей час необхідний для проходження процедури завантаження операційної системи, ініціалізації інтерфейсів і відновлення доступності керуючої IP-адреси [2, 20]. У модельній системі використано значення 120 секунд, що є достатнім для більшості невеликих маршрутизаторів і комутаторів.

Програмна реалізація запуску power cycle:

```
void power_cycle_start(app_context_t *ctx)
{
 if (!restart_allowed(ctx))
 {
```

```
 ctx->system_state = SYSTEM_LOCKOUT;
log_event(EVENT_LOCKOUT);
 return;
 }
 log_event(EVENT_POWER_CYCLE_START);
 relay_set(RELAY_OFF);
 ctx->power_cycle_timer = POWER_OFF_TIME_MS;
ctx->system_state = SYSTEM_RESTART;
}
```

У стані SYSTEM_RESTART програма не використовує блокуючу затримку. Замість цього застосовується програмний таймер power_cycle_timer, який зменшується в головному циклі програми. Після завершення таймера реле знову вмикається:

```
void process_restart_state(app_context_t *ctx)
{
 if (ctx->power_cycle_timer == 0)
{
 relay_set(RELAY_ON);
ctx->restart_count++;
ctx->recovery_timer = RECOVERY_TIME_MS;
ctx->system_state = SYSTEM_RECOVERY;
log_event(EVENT_RELAY_ON);
}
}
```

Використання неблокуючих таймерів є важливим, оскільки під час виконання процедури перезавантаження мікроконтролер повинен продовжувати обслуговувати watchdog, контролювати напругу живлення та зберігати можливість обробки критичних подій. Довгі блокуючі затримки типу *HAL_Delay(7000)* у такій системі небажані, оскільки вони тимчасово припиняють виконання інших задач [6, 12].

Окремо в програмній частині реалізується захист від циклічних перезапусків. Якщо Cisco-пристрій або інше телекомунікаційне обладнання має апаратну несправність, багаторазове вимкнення й увімкнення живлення не усуне проблему, але може додатково навантажувати блок живлення та скорочувати ресурс обладнання. Тому після заданої кількості невдалих спроб система переходить у стан LOCKOUT.

У модельній реалізації встановлено обмеження, не більше 3 перезапусків протягом 15 хвилин.

У стані LOCKOUT система продовжує моніторинг і передає повідомлення серверу, але не виконує нових перемикань реле без участі адміністратора.

Функція перевірки дозволу на перезапуск:

```
bool restart_allowed(app_context_t *ctx)
{
 if (!ctx->auto_restart_enabled)
return false;
 if (ctx->maintenance_mode)
return false;
if (ctx->restart_count >= MAX_RESTART_COUNT)
return false;
 if (ctx->power_state == POWER_BATTERY_CRITICAL ||
ctx->power_state == POWER_CUTOFF)
return false;
return true;
}
```

Ця функція перевіряє чотири основні умови. По-перше, автоматичний перезапуск має бути дозволений. По-друге, система не повинна перебувати в режимі обслуговування. По-третє, не має бути перевищено допустиму кількість перезапусків. По-четверте, резервне живлення не повинно перебувати у критичному стані.

Режим обслуговування потрібний для випадків, коли адміністратор виконує планові роботи з обладнанням: оновлення прошивки, зміну конфігурації або перевірку мережевого підключення. У цей момент обладнання може тимчасово не відповідати на запити, але система не повинна сприймати це як аварію та виконувати power cycle.

Другою важливою частиною програмної реалізації є контроль резервного живлення. У принциповій схемі напруга акумулятора подається на ADC-вхід PA0 / BAT_SENSE через резистивний подільник, а напруга основного джерела – на PA1 / MAIN_SENSE. Оскільки ADC STM32F411RE працює в діапазоні 0–3,3 В, подільник зменшує рівень вимірюваної напруги до безпечного значення [5, 11].

Функції вимірювання напруги:

```
float read_battery_voltage(void)
{
 uint32_t adc_value = adc_read_average(ADC_CHANNEL_BATTERY, 16);
 float u_adc = ((float)adc_value * 3.3f) / 4095.0f;
float u_in = u_adc * ((100.0f + 27.0f) / 27.0f);
return u_in;
}
```

У цій функції використовується усереднення 16 вимірювань. Це необхідно тому, що одиничний результат ADC може містити випадкову похибку через шум живлення, комутацію реле або електромагнітні завади. Усереднення дозволяє отримати більш стабільне значення без значного навантаження на мікроконтролер.

Для уникнення частих переходів між станами живлення бажано використовувати гістерезис. Наприклад, якщо критичний рівень встановлено на 10,8 В, то повернення з критичного стану можна дозволити лише після підвищення напруги вище 11,0 В. Це запобігає “дрижанням” логіки при напрузі, близькій до порогу.

Для модельної системи прийнято такі порогові значення:

- U > 12,0 В – нормальний стан;

- 11,5–12,0 В – робота від резервного живлення;
- 10,8–11,5 В – попереджувальний рівень;
- 10,5–10,8 В – критичний рівень;
- U ≤ 10,5 В – відключення навантаження.

У принциповій схемі для швидкого визначення наявності основного живлення також використовується цифровий сигнал MAIN_OK, сформований компаратором LM393. Компаратор порівнює напругу основного джерела із заданим порогом і формує логічний сигнал, який зчитується мікроконтролером [14]. Такий підхід дозволяє швидко визначити факт втрати основного живлення, не чекаючи чергового циклу ADC-усереднення.

Функція визначення стану живлення:

```
power_state_t get_power_state(float battery_voltage, bool main_power_present)
{
 if (main_power_present)
return POWER_MAIN;
if (battery_voltage > 11.5f)
return POWER_BATTERY;
if (battery_voltage > 10.8f)
 return POWER_BATTERY_WARNING;
if (battery_voltage > 10.5f)
return POWER_BATTERY_CRITICAL;
return POWER_CUTOFF;
}
```

При зникненні основного живлення система переходить у стан BATTERY_MODE.

Послідовність дій при переході на резервне живлення:

1. Сигнал MAIN_OK переходить у стан відсутності основного живлення.
2. STM32 фіксує подію EVENT_MAIN_POWER_LOST.
3. Система переходить у BATTERY_MODE.
4. Починається частіший контроль напруги акумулятора.

5. При досягненні попереджувального рівня формується повідомлення серверу.
6. При досягненні критичного рівня запускається алгоритм SHUTDOWN.

У стані POWER_BATTERY_WARNING система ще продовжує роботу, але передає серверу повідомлення про зниження напруги. У стані POWER_BATTERY_CRITICAL автоматичні перезапуски блокуються, а система переходить до безпечного вимкнення.

Безпечне вимкнення має виконуватись у два етапи. Спочатку, якщо сервер керування доступний, STM32 передає повідомлення про критичний рівень заряду. Сервер може виконати відповідні дії: для Cisco-пристрою – збереження поточної конфігурації через SSH, для сервера – команду завершення роботи операційної системи [19, 20]. Після цього мікроконтролер очікує завершення тайм-ауту SHUTDOWN_TIMEOUT_MS і вимикає живлення навантаження через реле:

BATTERY_CRITICAL → повідомлення серверу → очікування тайм-ауту → RELAY_OFF → POWER_SAVE

Якщо сервер недоступний, STM32 все одно виконує вимкнення після завершення тайм-ауту. Це необхідно для захисту акумулятора від глибокого розряду. Такий сценарій не є ідеальним для серверного обладнання, але він є кращим, ніж повне неконтрольоване знеструмлення внаслідок розряду АКБ [4].

У режимі POWER_SAVE навантаження вимкнене, а мікроконтролер продовжує мінімальний контроль стану живлення. Частота мережевих перевірок у цьому режимі може бути зменшена, щоб знизити власне енергоспоживання системи. Основною задачею стає очікування повернення основного живлення.

Після повернення основного живлення система не повинна одразу подавати живлення на навантаження. Спочатку запускається таймер стабілізації MAIN_POWER_STABLE_MS, наприклад 30 секунд. Якщо після завершення таймера сигнал MAIN_OK залишається активним, реле подає живлення на обладнання, система переходить у стан RECOVERY, а потім – у NORMAL:

POWER_SAVE → MAIN_POWER_RECOVERY → RELAY_ON → RECOVERY → NORMAL

Окремо слід зазначити, що в режимі критичного заряду АКБ функція power cycle блокується. Це зроблено тому, що перезапуск обладнання може викликати підвищене споживання струму під час старту, а в умовах низького заряду це може призвести до просідання напруги та некоректного запуску пристрою.

Узагальнено логіку керування живленням можна подати так:

- NORMAL: реле увімкнене, обладнання працює
- FAIL: підтверджений збій, перевірка дозволу на power cycle
- RESTART: реле вимкнене, живлення навантаження відключене
- RECOVERY: реле увімкнене, очікування завантаження обладнання
- BATTERY_MODE: робота від резервного джерела, контроль напруги АКБ
- SHUTDOWN: підготовка до безпечного вимкнення
- POWER_SAVE: навантаження вимкнене, очікування повернення основного живлення
- LOCKOUT: перезапуски заблоковані, очікується втручання адміністратора

Таким чином, програмна реалізація керування живленням тісно пов’язана з апаратною частиною принципової схеми. GPIO-вихід STM32 керує транзисторним ключем реле, ADC-входи забезпечують вимірювання основного та резервного живлення, компаратор LM393 формує цифровий сигнал наявності основної напруги, а програмний автомат станів визначає, коли саме виконувати перезапуск, вимкнення або відновлення навантаження.

3.3. Тестування системи та аналіз ефективності її роботи

Після розробки принципової схеми та програмної структури було виконано модельне тестування окремих режимів роботи системи в середовищі Proteus Design Suite, яке призначене для схемотехнічного проєктування, моделювання електронних схем і перевірки вбудованих систем [24]. Метою тестування була перевірка правильності взаємодії мікроконтролера STM32F411RE, Ethernet-

контролера W5500, вузла контролю живлення, компаратора LM393, транзисторного ключа на 2N2222 та силового реле K1 [7, 11, 14, 15, 16, 24].

Тестування проводилось для перевірки таких функцій: запуск системи, мережевий моніторинг Cisco-пристрою, виявлення втрати доступності, виконання апаратного перезавантаження, контроль резервного живлення, безпечне вимкнення навантаження, відновлення після повернення основного живлення та блокування повторних перезапусків.

Для модельної перевірки було прийнято такі параметри роботи системи:

- Період перевірки Cisco-пристрою: 5 с
- Кількість невдалих перевірок до збою: 3
- Час підтвердження збою: 15 с
- Час вимкнення живлення при power cycle: 7 с
- Час очікування відновлення: 120 с
- Максимальна кількість перезапусків: 3
- Часове вікно контролю перезапусків: 15 хв
- Тайм-аут безпечного вимкнення: 60 с
- Затримка після повернення живлення: 30 с

На першому етапі було перевірено запуск системи. Після подачі живлення стабілізатор LM2596S-5.0 формує напругу 5 В, а стабілізатор AMS1117-3.3 — напругу 3,3 В для цифрової частини схеми [9, 10]. Після цього мікроконтролер виконує ініціалізацію GPIO, ADC, SPI, UART, W5500 та встановлює реле у початковий безпечний стан. У нормальному режимі система переходить у стан NORMAL, реле залишається увімкненим, а Cisco-пристрій отримує живлення.

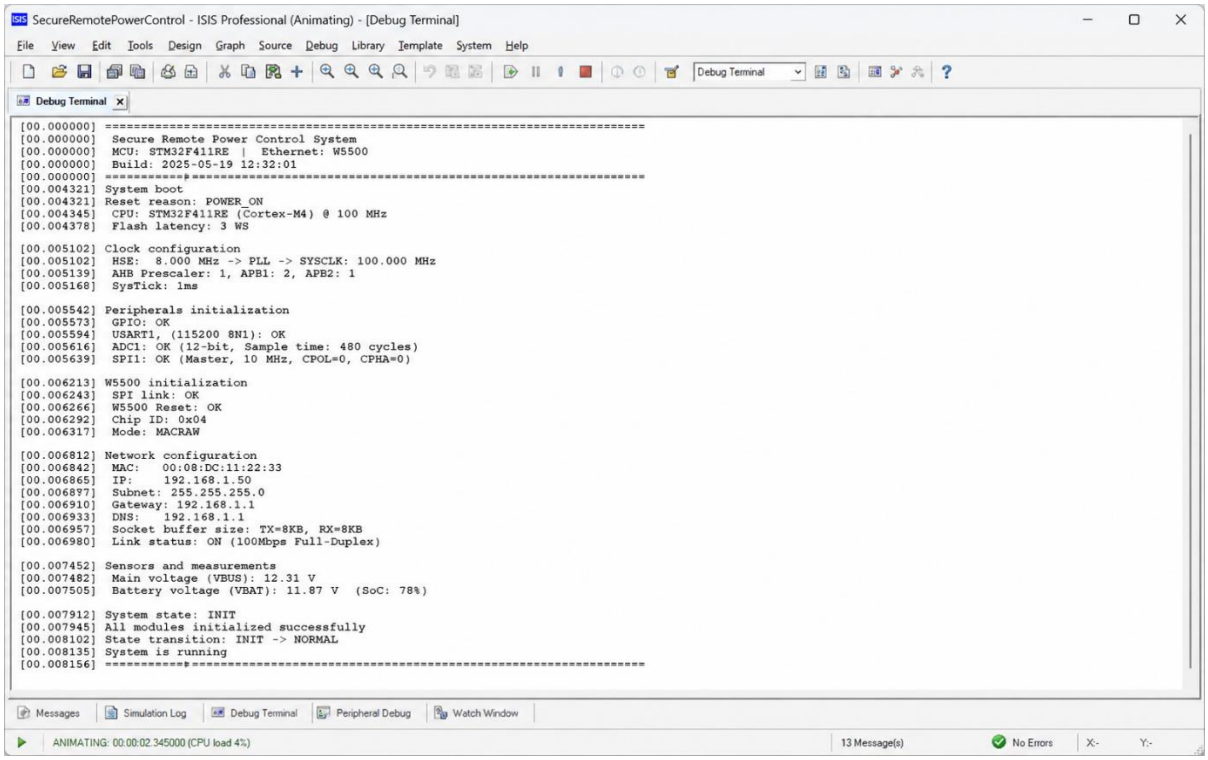


Рисунок 3.1 – Ініціалізація мікроконтролера STM32F411RE, W5500 та периферійних модулів

У нормальному режимі система періодично перевіряє доступність Cisco-пристрою через мережевий інтерфейс. Для цього використовується Ethernet-контролер W5500, який забезпечує TCP/IP-з’єднання через SPI-інтерфейс із STM32 [7]. Якщо Cisco-пристрій відповідає на перевірку порту TCP/22, стан обладнання визначається як CISCO_OK, а лічильник помилок залишається рівним нулю.

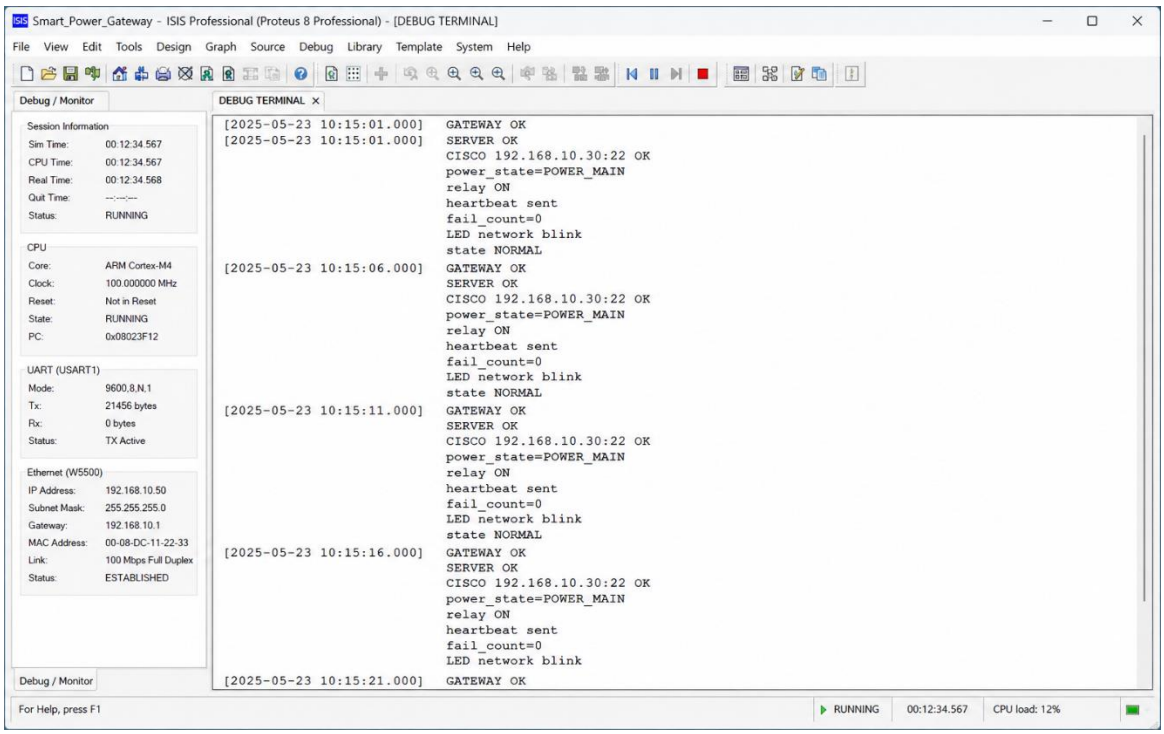


Рисунок 3.2 – Нормальний режим моніторингу доступності телекомунікаційного обладнання

Для перевірки аварійного режиму було змодельовано втрату доступності Cisco-пристрою. Після першої невдалої перевірки система не виконує перезавантаження, а лише збільшує лічильник fail_count. Після трьох послідовних невдалих перевірок система переходить у стан FAIL.

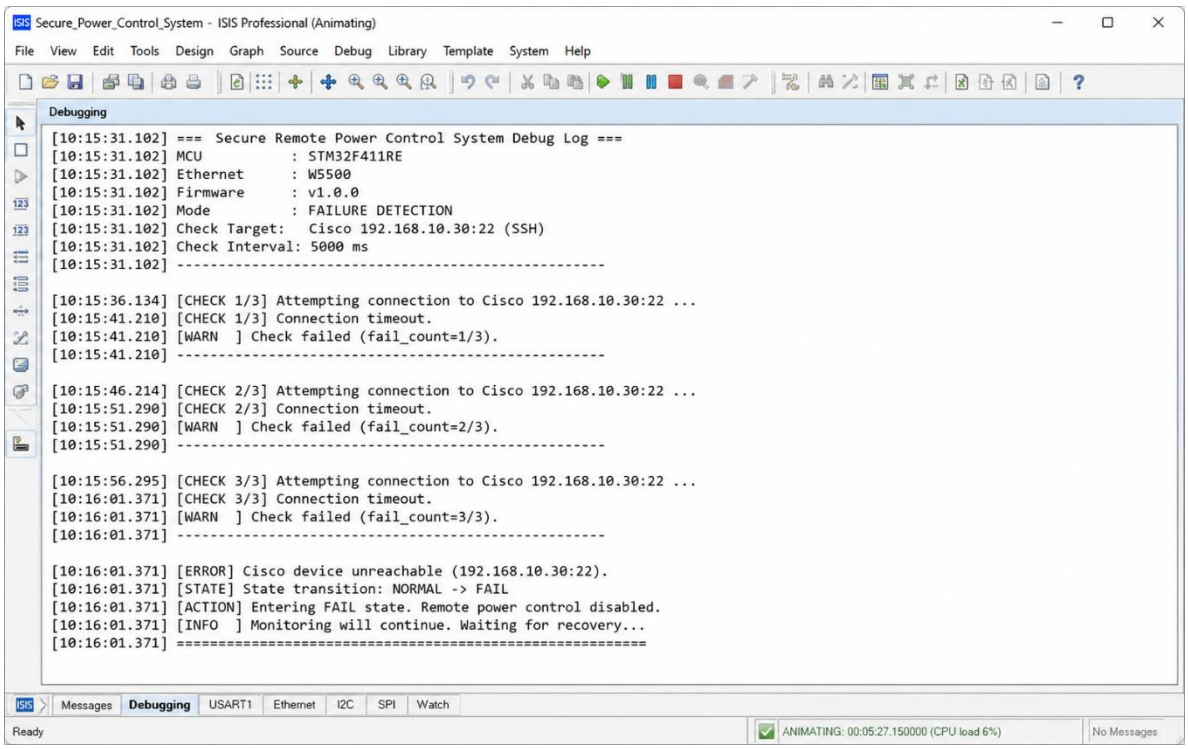


Рисунок 3.3 – Виявлення втрати доступності Cisco-пристрою

Після підтвердження збою система перевіряє, чи дозволено виконувати апаратне перезавантаження. Якщо режим обслуговування не активований, ліміт перезапусків не перевищено, а напруга резервного джерела не перебуває у критичному стані, запускається процедура power cycle. У цьому режимі мікроконтролер через сигнал RELAY_CTRL керує транзисторним ключем і реле K1, яке вимикає живлення навантаження на 7 с, після чого живлення подається повторно [15, 16].

Після повторної подачі живлення система переходить у стан RECOVERY і очікує 120 с, необхідних для завантаження Cisco-пристрою та відновлення мережевої доступності. Якщо після завершення цього часу пристрій знову відповідає на перевірку, система повертається у стан NORMAL. Якщо доступність не відновлюється, система може виконати повторну спробу або перейти у стан LOCKOUT після перевищення встановленого ліміту перезапусків.

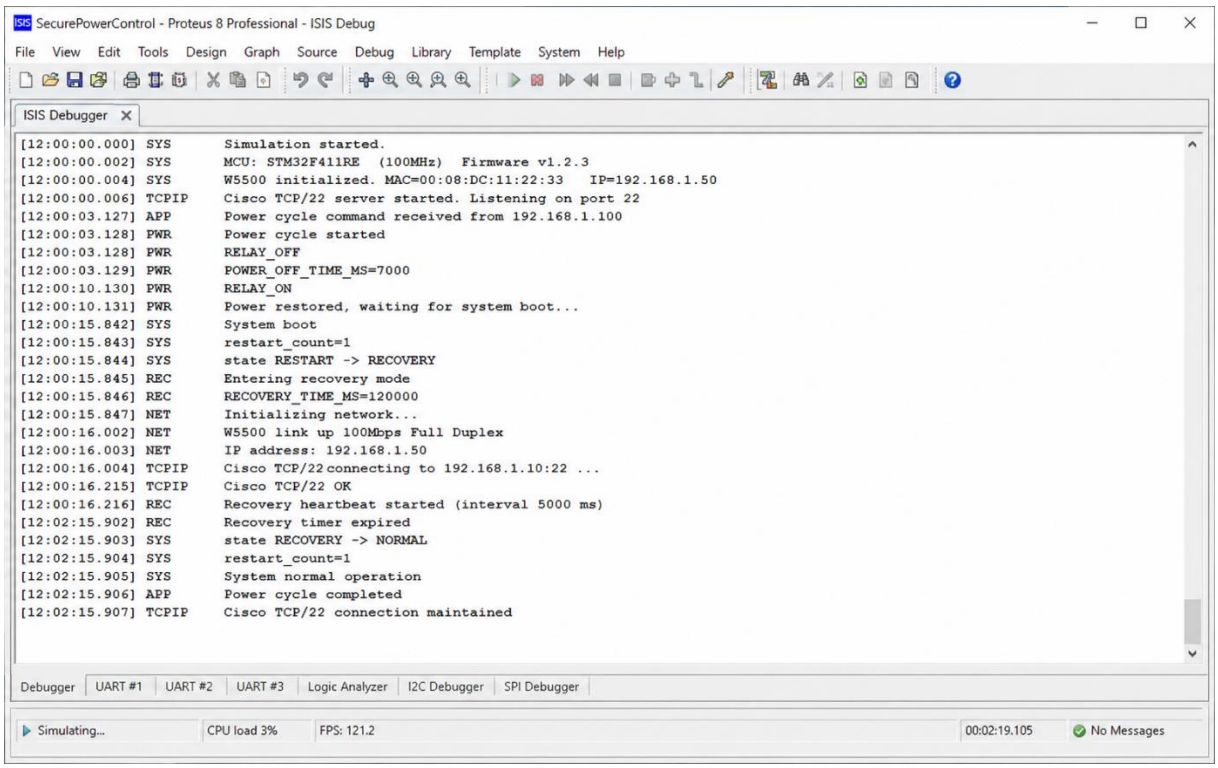


Рисунок 3.4 – Відновлення роботи системи після повторної подачі живлення

Також перевірено роботу системи при втраті основного живлення. Факт зникнення основного живлення визначається за сигналом MAIN_OK, сформованим компаратором LM393, а також за аналоговим сигналом MAIN_SENSE [16]. Після втрати основного живлення система переходить у режим BATTERY_MODE і продовжує працювати від резервного джерела.

Напруга акумулятора контролюється через ADC-вхід BAT_SENSE. У модельній системі використовуються такі порогові значення:

- U > 12,0 В — нормальний стан;
- 11,5–12,0 В — робота від резервного живлення;
- 10,8–11,5 В — попереджувальний рівень;
- 10,5–10,8 В — критичний рівень;
- U ≤ 10,5 В — відключення навантаження.

При досягненні критичного рівня заряду система блокує виконання power cycle і переходить до безпечного вимкнення. Якщо сервер керування доступний, йому передається повідомлення про критичний стан живлення. Після завершення

тайм-ауту 60 с мікроконтролер вимикає реле, відключає навантаження та переходить у стан POWER_SAVE [4, 19, 20].

Після повернення основного живлення система не вмикає навантаження одразу. Спочатку виконується затримка стабілізації 30 с. Якщо протягом цього часу основне живлення залишається присутнім, реле знову подає живлення на обладнання, система переходить у стан RECOVERY, а після відновлення доступності Cisco-пристрою — у стан NORMAL.

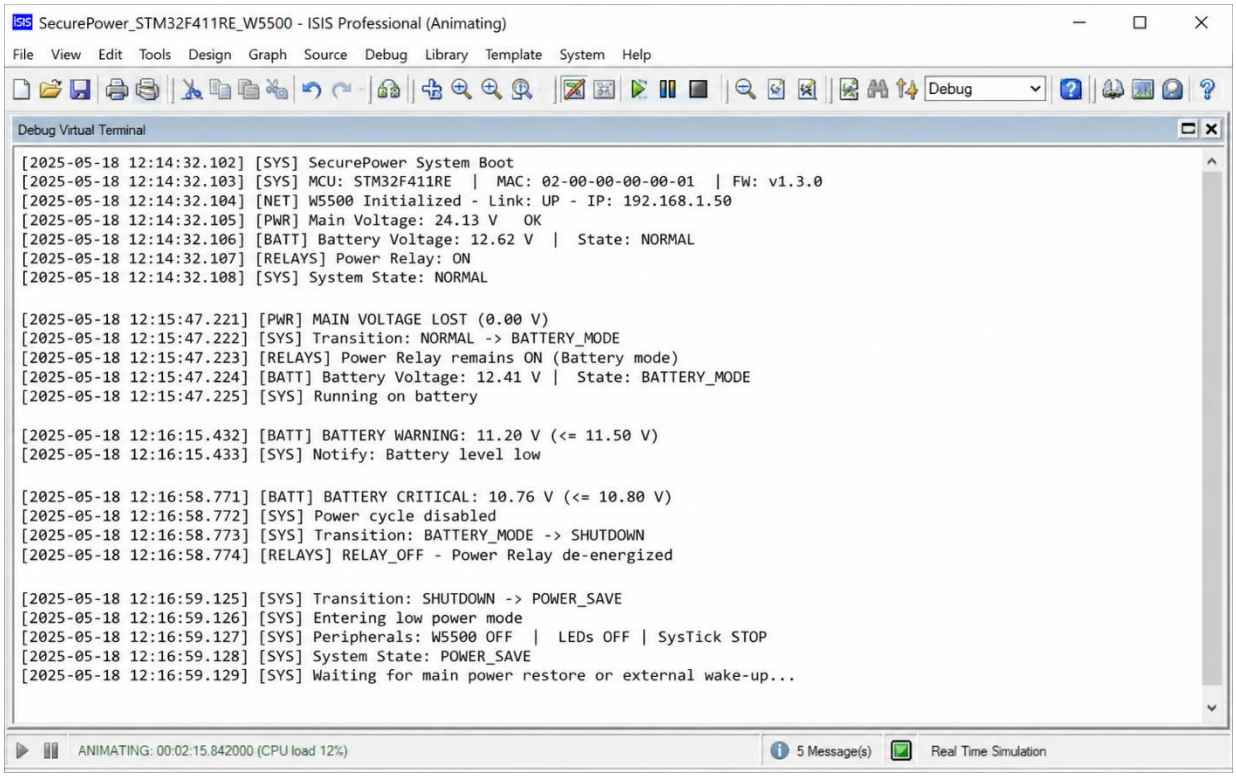


Рисунок 3.4. – Перехід системи на резервне живлення

За результатами тестування було проведено оцінку ефективності системи. Одним із головних показників є час реакції на втрату доступності Cisco-пристрою. У модельній конфігурації система підтверджує збій за 15 с. Після цього виконується відключення живлення на 7 с і очікування відновлення протягом 120 с.

Таким чином, система здатна виявити збій, виконати апаратне перезавантаження та перейти до перевірки відновлення приблизно за 142 с. Для віддалених телекомунікаційних вузлів це є прийнятним результатом, оскільки ручне втручання персоналу зазвичай потребує значно більшого часу.

Важливою перевагою системи є те, що вона не приймає рішення на основі одиничної події. Перехід у стан FAIL відбувається тільки після кількох послідовних

невдалих перевірок. Це зменшує ризик хибного перезапуску при короткочасному мережевому збої. Крім того, апаратне перезавантаження виконується лише після перевірки додаткових умов: режиму обслуговування, кількості попередніх перезапусків і стану резервного живлення.

У режимі резервного живлення система змінює пріоритет роботи. Якщо в нормальному режимі основною задачею є відновлення працездатності обладнання, то при критичному заряді акумулятора пріоритетом стає безпечне вимкнення. Це дозволяє уникнути глибокого розряду АКБ та неконтрольованого знеструмлення обладнання [4].

Порівняно з простими пристроями типу watchdog-реле, запропонована система має ширші функціональні можливості. Вона не лише виконує перезавантаження при втраті відповіді, а й контролює основне та резервне живлення, аналізує кількість попередніх спроб перезапуску, підтримує режим обслуговування, передає повідомлення серверу та переходить до безпечного вимкнення при критичному заряді акумулятора.

Разом із тим система має певні обмеження. Перевірка доступності через TCP/22 не завжди повністю відображає стан усього Cisco-пристрою: обладнання може відповідати на SSH-порт, але мати проблеми з окремими інтерфейсами або службами. Для підвищення точності моніторингу доцільно додати SNMP-опитування параметрів пристрою, зокрема стану інтерфейсів, uptime, завантаження процесора та пам’яті [21]. Також порогові значення напруги АКБ мають налаштовуватися під конкретний тип резервного джерела. Апаратне вимкнення живлення слід розглядати як аварійний метод, який не повинен замінювати штатне адміністрування через SSH або інші засоби Cisco IOS [19, 20].

Система здатна виявляти критичні збої, виконувати контрольоване апаратне перезавантаження, запобігати циклічним перезапускам і забезпечувати безпечне вимкнення обладнання при вичерпанні ресурсу резервного живлення.

ВИСНОВКИ

У кваліфікаційній роботі досягнуто поставленої мети: розроблено та обґрунтовано систему захищеного віддаленого керування живленням телекомунікаційного обладнання в умовах критичних збоїв. Запропоноване рішення забезпечує моніторинг доступності обладнання, контроль основного та резервного живлення, апаратне перезавантаження у разі підтвердженого збою та безпечне вимкнення при критичному зниженні заряду акумулятора.

У роботі проаналізовано причини збоїв телекомунікаційного обладнання, методи моніторингу доступності та засоби захищеного віддаленого керування. Встановлено, що використання лише програмних засобів адміністрування, таких як SSH або SNMP, не завжди є достатнім, оскільки при повному зависанні пристрою мережевий доступ може бути втрачено. Тому доцільним є поєднання захищеного доступу через VPN, мережевого моніторингу та апаратного керування живленням.

Розроблено архітектуру системи на базі мікроконтролера STM32F411RE та Ethernet-контролера W5500. До складу системи входять вузли стабілізації живлення, вимірювання напруги, контролю наявності основного живлення, релейний комутаційний вузол, LED-індикація, SWD-інтерфейс програмування та UART-інтерфейс діагностики.

Програмна логіка побудована на автоматі станів, який забезпечує перехід між режимами нормальної роботи, виявлення збою, перезавантаження, відновлення, роботи від резервного живлення, безпечного вимкнення та блокування повторних перезапусків. Це дозволяє уникнути хибних спрацювань і циклічного перезавантаження обладнання.

За результатами модельного тестування підтверджено працездатність системи в основних режимах. Система виявляє втрату доступності Cisco-пристрою, виконує power cycle через реле, контролює напругу АКБ та переходить до безпечного вимкнення при критичному рівні заряду.

Отже, запропонована система може бути використана як основа для автономного засобу керування живленням телекомунікаційного обладнання. Подальше вдосконалення може передбачати підтримку SNMPv3, декількох каналів керування живленням, збереження журналів подій та інтеграцію із системами моніторингу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tanenbaum A. S., Wetherall D. J. *Computer Networks*. 5th ed. Boston: Pearson, 2011. 960 p. Дата звернення: 20.04.2026 URL: https://www.amazon.com/Computer-Networks-5th-Andrew-Tanenbaum/dp/0132126958

2. Cisco Systems. *Internetworking Technologies Handbook*. 4th ed. Cisco Press, 2003. Дата звернення: 25.04.2026 URL: https://www.ciscopress.com/store/internetworking-technologies-handbook-9781587051197

3. Stallings W. *Data and Computer Communications*. 10th ed. Pearson, 2014. 912 p. Дата звернення: 04.05.2026 URL: https://www.pearson.com/en-us/subject-catalog/p/data-and-computer-communications/P200000003298

4. Eaton. *UPS and Power Management Fundamentals Handbook*. Eaton Corporation. Дата звернення: 09.05.2026 URL: https://www.eaton.com/content/dam/eaton/products/backup-power-ups-surge-it-power-distribution/backup-power-ups/eaton-ups-fundamentals-handbook-en-us-2025.pdf

5. STMicroelectronics. *RM0383 Reference Manual. STM32F411xC/E advanced Arm-based 32-bit MCUs*. STMicroelectronics. Дата звернення: 05.05.2026 URL: https://www.st.com/resource/en/reference_manual/rm0383-stm32f411xce-advanced-armbased-32bit-mcus-stmicroelectronics.pdf

6. Barr M., Massa A. *Programming Embedded Systems: With C and GNU Development Tools*. 2nd ed. Sebastopol: O'Reilly Media, 2006. 336 p. Дата звернення: 28.04.2026 URL: https://www.oreilly.com/library/view/programming-embedded-systems/0596009836/

7. WIZnet. *W5500 Datasheet. Hardwired TCP/IP Embedded Ethernet Controller*. Version 1.1.0. WIZnet Co., Ltd. Дата звернення: 29.04.2026 URL: https://docs.wiznet.io/img/products/w5500/W5500_ds_v110e.pdf

8. Littelfuse. *SMBJ Series TVS Diodes Datasheet*. Littelfuse. Дата звернення: 30.04.2026 URL: https://www.littelfuse.com/assetdocs/tvs-diodes-smbj-series-datasheet?assetguid=ba555e99-a12d-4f72-a0b6-86b06c67171e

9. Texas Instruments. *LM2596 SIMPLE SWITCHER Power Converter 150-kHz, 3-A Step-Down Voltage Regulator Datasheet*. Texas Instruments. Дата звернення: 29.04.2026 URL: https://www.ti.com/lit/ds/symlink/lm2596.pdf

10. Advanced Monolithic Systems. *AMS1117 Low Dropout Voltage Regulator Datasheet*. Advanced Monolithic Systems. Дата звернення: 29.04.2026 URL: https://www.advanced-monolithic.com/pdf/ds1117.pdf

11. STMicroelectronics. *STM32F411xC / STM32F411xE Datasheet*. STMicroelectronics. Дата звернення: 29.04.2026 URL: https://www.st.com/resource/en/datasheet/stm32f411ce.pdf

12. STMicroelectronics. *UM1725. Description of STM32F4 HAL and Low-Layer Drivers*. STMicroelectronics. Дата звернення: 29.04.2026 URL: https://www.st.com/resource/en/user_manual/um1725-description-of-stm32f4-hal-and-lowlayer-drivers-stmicroelectronics.pdf

13. HanRun Electronics. *HR911105A RJ45 Connector with Integrated Magnetics Datasheet*. HanRun Electronics. Дата звернення: 30.04.2026 URL: https://core-electronics.com.au/attachments/localcontent/cc11be56d66bb63d5f1eeb85492439c0_335950a6d06.pdf

14. Texas Instruments. *LM393 Dual Differential Comparator Datasheet*. Texas Instruments. Дата звернення: 29.04.2026 URL: https://www.ti.com/lit/ds/symlink/lm393.pdf

15. onsemi. *P2N2222A NPN Silicon Transistor Datasheet*. onsemi. URL: https://www.onsemi.com/pdf/datasheet/p2n2222a-d.pdf

16. Ningbo Songle Relay. *SRD Series Power Relay Datasheet*. Ningbo Songle Relay. Дата звернення: 30.04.2026 URL: https://www.alldatasheet.com/datasheet-pdf/pdf/1131793/SONGLERELAY/SRD-12VDC-SL-C.html

17. Vishay Semiconductors. *1N4148 Small Signal Fast Switching Diode Datasheet*. Vishay Semiconductors. Дата звернення: 30.04.2026 URL: https://www.vishay.com/docs/81857/1n4148.pdf

18. onsemi. *1N4001–1N4007 General Purpose Rectifier Datasheet*. onsemi. Дата звернення: 30.04.2026 URL: https://www.onsemi.com/products/discrete-power-modules/rectifiers/1n4007

19. Cisco Systems. *Configure SSH on Routers and Switches / Configure SSH on Routers*. Cisco Systems. Дата звернення: 27.04.2026 URL: https://www.cisco.com/c/en/us/support/docs/security-vpn/secure-shell-ssh/4145-ssh.html

20. Cisco Systems. *Rebooting and Reloading — Cisco IOS Configuration Guide*. Cisco Systems. Дата звернення: 02.05.2026 URL:

https://www.cisco.com/c/en/us/td/docs/ios/fundamentals/configuration/guide/15_0s/cf_15_0S_book/cf_rebooting.html

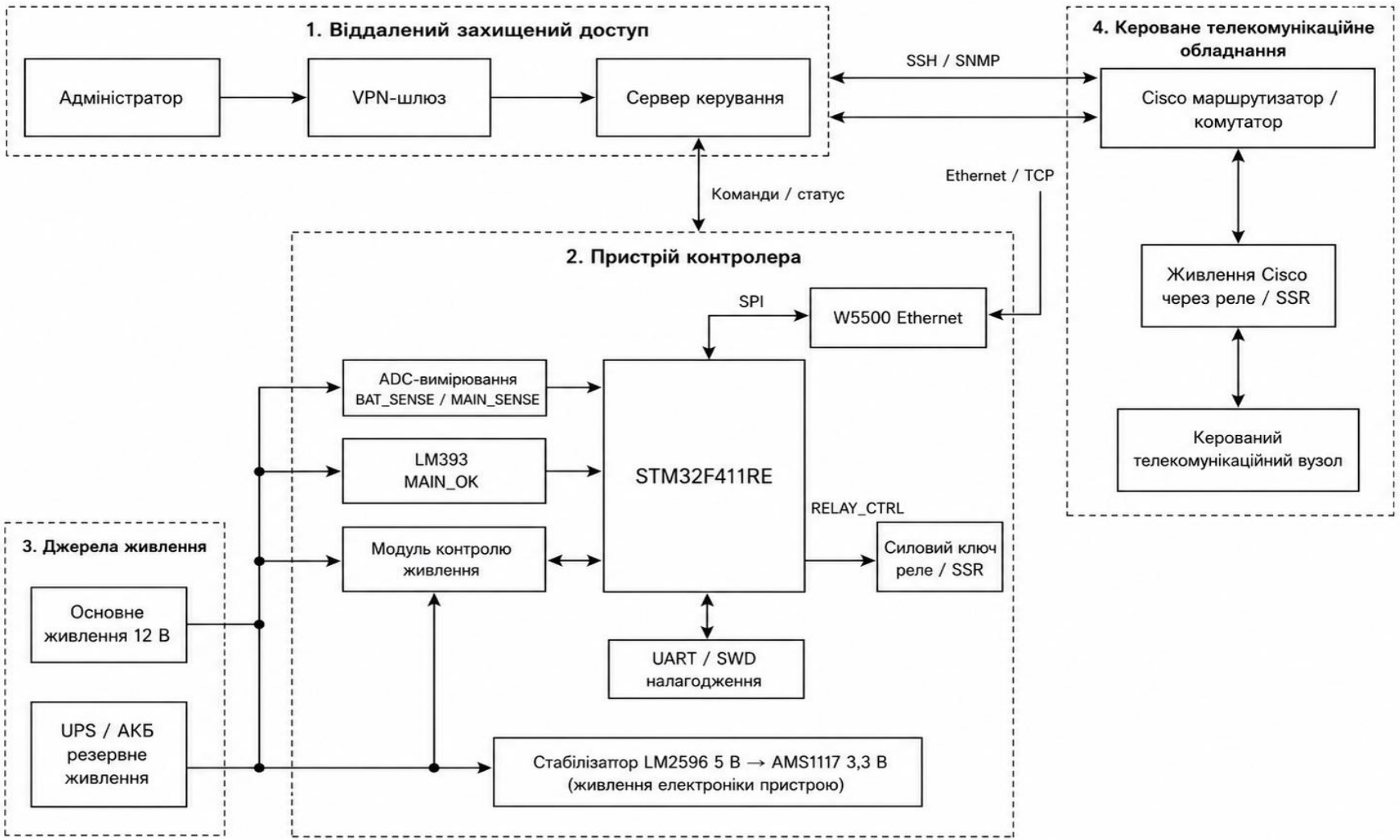
21. Cisco Systems. *SNMP Configuration Guide, Cisco IOS XE*. Cisco Systems. Дата звернення: 28.04.2026 URL: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/snmp/configuration/xe-17/snmp-xe-17-book.html

22. Cisco Systems. *Understanding Cisco Console and AUX Port Cable Requirements*. Cisco Systems. Дата звернення: 03.05.2026 URL: https://www.cisco.com/c/en/us/support/docs/switches/catalyst-5000-series-switches/12016-2.html

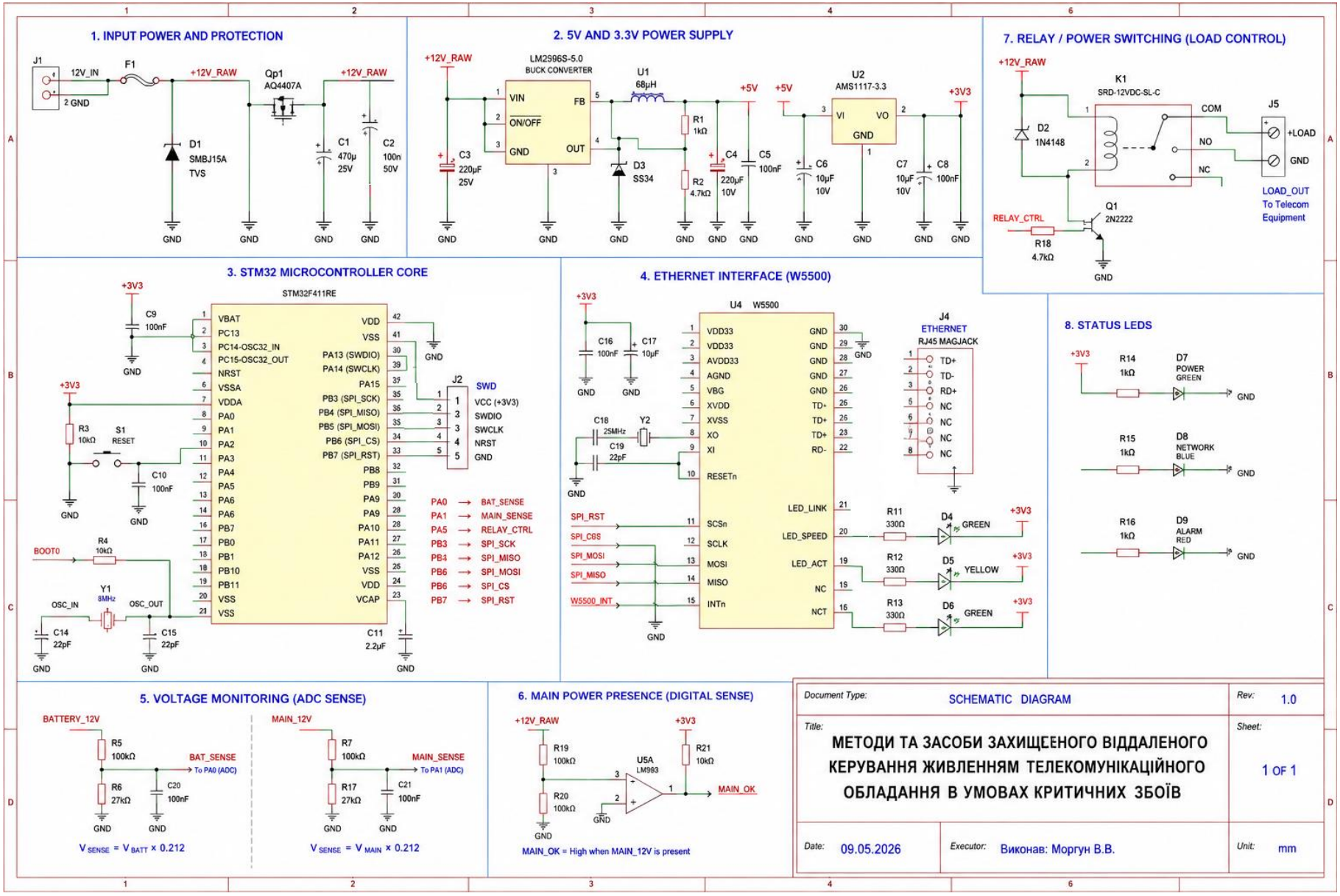
23. Cisco Systems. *SNMP Configuration Guide, Cisco IOS XE*. Cisco Systems. Дата звернення: 28.04.2026 URL: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/snmp/configuration/xe-17/snmp-xe-17-book.html

24. Labcenter Electronics. *Proteus Design Suite: PCB Design and Circuit Simulator Software*. Labcenter Electronics Ltd. Дата звернення: 10.05.2026 URL: https://www.labcenter.com/

ДОДАТОК А СТРУКТУРНА СХЕМА ПРИСТРОЮ



ДОДАТОК Б ПРИНЦИПОВА СХЕМА ПРИСТРОЮ



ДОДАТОК В ПЕРЕЛІК ЕЛЕМЕНТІВ

Поз.	*Найменування*	*Кіл.*	*Прим.*
	Конденсатори		
C1	470 мкФ, 25 В	1	Електролітичний, вхідний фільтр
C2, C5, C8, C9, C10, C12, C13, C20, C21	100 нФ	9	Керамічні, корпус 0805
C3, C4	220 мкФ, 25 В / 10 В	2	Електролітичні, фільтр DC/DC
C6, C7	10 мкФ, 10 В	2	Керамічні або танталові
C11	2,2 мкФ	1	VCAP для STM32
C14, C15, C18, C19	22 пФ	4	Для кварцових резонаторів
C16	100 нФ	1	Розв’язка живлення W5500
C17	10 мкФ	1	Фільтр живлення W5500
	Резистори		
R1	1 кОм	1	Ланцюг зворотного зв’язку DC/DC
R2	4,7 кОм	1	Ланцюг зворотного зв’язку DC/DC
R3, R4, R8, R21	10 кОм	4	Підтягувальні резистори
R5, R7, R19, R20	100 кОм	4	Подільники напруги
R6, R17	27 кОм	2	Подільники BAT_SENSE / MAIN_SENSE
R11, R12, R13	330 Ом	3	Обмеження струму LED Ethernet
R14, R15, R16	1 кОм	3	Обмеження струму індикаторів
R18	4,7 кОм	1	Базовий резистор транзистора реле
	Індуктивності		
L1	68 мкГн	1	Дросель понижувального перетворювача
	Мікросхеми		
U1	LM2596S-5.0	1	Імпульсний стабілізатор 5 В
U2	AMS1117-3.3	1	LDO стабілізатор 3,3 В
U3	STM32F411RE	1	Мікроконтролер системи
U4	W5500	1	Ethernet-контролер з TCP/IP стеком
U5	LM393	1	Компаратор MAIN_OK
	Транзистори		
Qp1	AO4407A	1	P-канальний MOSFET, захист від переполюсування
Q1	2N2222 / P2N2222A	1	NPN ключ керування реле
	Діоди		
D1	SMBJ15A	1	TVS-захист лінії +12V_RAW
D2	1N4148 або 1N4007	1	Захист котушки реле від ЕРС самоіндукції
D3	SS34	1	Діод Шотткі для LM2596
D4, D6, D7	LED GREEN	3	Індикація Ethernet / POWER
D5	LED YELLOW	1	Індикація Ethernet activity
D8	LED BLUE	1	NETWORK
D9	LED RED	1	ALARM
	Кварцові резонатори		
Y1	8 МГц	1	Тактування STM32F411RE
Y2	25 МГц	1	Тактування W5500
	Реле та комутаційні елементи		
K1	SRD-12VDC-SL-C	1	Реле комутації живлення Cisco-навантаження
	Запобіжники та захист		
F1	PTC 3 А або запобіжник 2 А	1	Захист вхідної лінії живлення
	Перемикачі та кнопки		
S1	RESET	1	Кнопка скидання мікроконтролера
	З’єднувачі		
J1	TBLOCK-2	1	Вхід основного живлення 12 В
J2	SWD HEADER 1×5	1	Програмування та налагодження
J3	TBLOCK-2 / TBLOCK-3	1	Вихід живлення на Cisco-обладнання
J4	RJ45 MagJack HR911105A	1	Ethernet із вбудованими трансформаторами
J5	UART DEBUG 1×3	1	Сервісний інтерфейс UART