Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра спеціалізованих комп'ютерних систем

ВОЗНЯК Вікторія Сергіївна

КОМП'ЮТЕРНО-ІНТЕГРОВАНА СИСТЕМА КОНТРОЛЮ СТАНУ ТА КЕРУВАННЯ ЖИВЛЕННЯМ МЕРЕЖЕВОГО ОБЛАДНАННЯ/ A COMPUTER-INTEGRATED CONTROL SYSTEM WILL BE USED TO CONTROL THE PROPERTY

спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології
освітньо-професійна програма – Автоматизація та комп'ютерно-інтегровані технології

Випускна кваліфікаційна робота
здобувача першого (бакалаврського) рівня освіти

Виконала: студентка групи АКІТ–41
В.С. Возняк
_____
Науковий керівник:
д.т.н., професор Н.Я. Возна
_____

Випускну кваліфікаційну роботу
допущено до захисту:
"____"_____2024 р.

Завідувач кафедри СКС
_____ А. І. Сегін

Тернопіль 2024

АНОТАЦІЯ

Возняк В.С. Комп’ютерно-інтегрована система контролю стану та керування живленням мережевого обладнання. – Рукопис.

Дослідження на здобуття освітнього ступеня «бакалавр» за спеціальністю 151 «Автоматизація та комп’ютерно-інтегровані технології», освітньо-професійна (наукова) програма. – Західноукраїнський національний університет, Тернопіль, 2024.

У роботі досліджено комп’ютерні мережі та причини і фактори, що впливають на їх надійність, зроблено порівняння та аналіз систем безперебійного живлення.

Проаналізовано способи контролю та моніторингу стану мережевих пристроїв. Досліджено одноплатні комп’ютери. Розроблена та побудована блок-схема послідовності виконання задач програмно-апаратним модулем.

ANNOTATION

Voznyak V.S. A computer-integrated control system will be used to control the property. - Manuscript.

Research for obtaining a bachelor's degree in the specialty 151 "Automation and computer-integrated technologies", educational and professional (scientific) program. – Western Ukrainian National University, Ternopil, 2024.

The paper examines computer networks and causes and factors affecting their reliability, compares and analyzes uninterruptible power supply systems.

Methods of controlling and monitoring the status of network devices are analysed. Single-board computers are investigated. A flowchart of the sequence of tasks performed by a hardware and software module is developed and constructed.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

FTP – file transfer protocol, протокол передачі файлів;

HTTP – hypertext transfer protocol, протокол передачі гіпертексту;

TCP/IP – transmission control protocol/internet protocol, протокол керування передачею/протокол Інтернету;

LTE – long term evolution, довготривала еволюція;

VPN – virtual private network, віртуальна приватна мережа;

ББЖ – блок беперебійного живлення;

PSU – power supply unit блок живлення;

КМ – комп'ютерні мережі;

CPU – central processing unit, центральний процесор;

ЦП – центральний процесор;

DNS – domain name system, система доменних імен;

ЕМП – електромагнітні перешкоди;

EMI – electro-magnetic interference, електромагнітні перешкоди;

WDT – WatchDog Timer, Таймер WatchDog;

ОС – операційні системи;

ОП – основна програма;

СТ – сторожовий таймер;

BIOS – basic input/output system, базова система введення/виведення;

DoS – denial of service, відмова в обслуговуванні;

СПЗ – сторожове програмне забезпечення ;

IPMI – Intelligent Platform Management Interface, Інтелектуальний інтерфейс керування платформою;

UEFI – unified extensible firmware interface, уніфікований розширюваний інтерфейс прошивки;

KVM – keyboard, video and mouse клавіатура, відео та миша;

BMC – baseboard management computing, обчислення керування базовою платою;

I2C – Inter-Integrated Circuit, міжінтегральна схема;

UDP – User Datagram Protocol, протокол дейтаграм користувача;

SBC – Single Board Computer, одноплатний комп'ютер;

ПК – персональний комп’ютер;

GPIO – general-purpose input/output, введення/виведення загального призначення;

MND – Managed network device, керований мережевий пристрій;

OID – object identifier, ідентифікатор об'єкта;

MIB – management information base, управлінська інформаційна база.

ВСТУП

Актуальність теми. Сучасне життя неможливо уявити без Інтернету, який став невід'ємною частиною людського існування, пронизуючи всі сфери діяльності від особистого спілкування до професійної діяльності. Цей глобальний мережевий простір також відіграє важливу роль у розвитку економіки, політики та суспільства в цілому.

Основою структури Інтернету є комп'ютерні мережі, які дозволяють об'єднувати пристрої в єдину систему комунікації. Ці мережі дозволяють передавати дані між комп'ютерами та іншими мережевими пристроями, такими як маршрутизатори та комутатори, що забезпечує зв'язок між різними частинами Інтернету.

Протокол TCP/IP (Transmission Control Protocol/Internet Protocol) у функціонуванні Інтернету надає набір правил та стандартів для передачі даних через мережу і забезпечуючи надійну та ефективну передачу даних.

Комп’ютерні мережі відіграють ключову роль у функціонуванні Інтернету, виконуючи важливі функції: передача даних, надійність та безпеку. Передача даних забезпечує обмін інформації між комп'ютерами, що дозволяє користувачам мати доступ до веб-сайтів, електронної пошти, онлайн-сервісів та інших ресурсів. Надійність комп'ютерних мереж забезпечують резервування каналів зв'язку, що робить Інтернет більш стійким до збоїв та перебоїв**.** А також вони відповідають за захист даних від несанкціонованого доступу та інших загроз.

Україна, знаходячись у військовому конфлікті, особливо залежить від доступу до Інтернету для забезпечення комунікації, отримання інформації та забезпечення безпеки в умовах кризових ситуацій. Проте блек-аути, які стали частим явищем в Україні через війну, створюють значні проблеми для доступу до мережевих послуг. Відключення електроенергії призводить до відключення мережевого обладнання, що робить неможливим доступ до Інтернету для багатьох людей.

Мета кваліфікаційної роботи полягає в розробці комп'ютерно-інтегрованої системи контролю стану та керування живленням мережевого обладнання для моніторингу мережевого пристрою та керування його живленням.

Для досягнення мети потрібно виконати такі завдання:

– дослідити причини та фактори, що впливають на надійність компютерних мереж;

– проаналізувати системи безперебійного живлення та одноплатні комп'ютери;

– зробити аналіз способів контролю стану мережевих пристроїв;

– на основі досліджень спроектувати модуль системи контролю стану та керування живленням мережевого обладнання;

– побудувати функціональну схему програмного забезпечення.

Предметом дослідження є комп'ютерна мережа.

Об'єктом дослідження є мережеві пристрої.

Методи дослідження – огляд факторів, що впливають на комп'ютерні мережі, огляд та порівняння блоків безперебійного живлення. Аналіз систем для контролю та моніторингу стану мережевих пристроїв.

Практичне значення одержаних результатів. Запропонована КІС дозволить автоматизувати роботу оператора по контролю стану мережевого обладнання.

Апробація. Кореляційні моделі в полярній системі координат / А.І. Сегін, Ю.І. Попик, В.С. Возняк [та ін.] // Матеріали проблемно-наукової міжгалузевої конференції ISCM – 2023, 2023. – С.188-191.

1 НАДІЙНІСТЬ КОМПЮТЕРНИХ МЕРЕЖ ТА СИСТЕМИ БЕЗПЕРЕБІЙНОГО ЖИВЛЕННЯ

1.1 Характеристики комп’ютерних мереж та фактори їх надійності.

Основними складовими комп'ютерної мережі є пристрої мережі, вузли мережі (або хости), мережеве програмне забезпечення і мережеві протоколи.

Пристрої мережі: маршрутизатори, шлюзи, комутатори та концентратори, а також різноманітні мережеві пристрої, такі як точки доступу Wi-Fi, медіа-конвертери, повторювачі, мости, брендмауери, фаєрволи тощо [30].

Вузли мережі: сервери, комп'ютери, плашети, термінали, телефони.

До мережевого програмного забезпечення входять операційні системи, програми для обміну даними (наприклад, FTP, HTTP), а також програми для безпеки мережі.

Мережеві протоколи – це стандартизовані набори правил, які визначають, як дані передаються через мережу. Найбільш відомий приклад - це протокол TCP/IP, який використовується в Інтернеті [16,19].

Топологія мережі (таблиця 1.1.) визначає спосіб, якими пристрої підключені між собою.

Таблиця 1.1 – Основні типи топологій

Тип	Опис
Зірка	Кожен пристрій підключений до центрального вузла, що спрощує адміністрування та забезпечує надійність мережі
Лінійна (або шинна)	Пристрої підключені один до одного у формі лінії. Це менш надійна топологія через її вразливість до відмов
Кільце	Пристрої з'єднані у кільце, де кожен пристрій має два сусіди. Він забезпечує високу швидкодію та стійкість, але вимагає складної інфраструктури
Дерево	Комбінація зіркової топології, де центральні вузли є мережами зірки, а їх вузли з'єднані між собою.

Класифікація комп'ютерних мереж (таблиця 1.2) за типом підключених пристроїв враховує спосіб, яким пристрої з'єднані в мережу та їхній спосіб взаємодії.

Таблиця 1.2 – Класифікація мереж за масштабом та їх характеристики.

Клас мережі	Опис
Локальні мережі (LAN)	Обмежені територією одного офісу, будинку або кампусу.
Місцеві мережі розподіленого доступу (DAN)	Розширення LAN на великі території, такі як місто або регіон.
Метрополітені мережі (MAN)	Охоплюють ціле місто або регіон.
Глобальні мережі (WAN)	Охоплюють великі географічні області, такі як країни, континенти або весь світ. Найвідомішим прикладом є Інтернет.

Основні класи включають:

– провідні мережі, вони часто використовуються там, де потрібна висока швидкість передачі даних або де безпека є пріоритетом, забезпечують стабільне з'єднання, оскільки фізичний кабель менше піддається перешкодам. Однак вони можуть бути менш гнучкими в установці та налаштуваннях;

– бездротові мережі – ідеально підходять для ситуацій, коли фізичне підключення кабелю є неефективним або неможливим, наприклад, для мобільних пристроїв або великих відкритих просторів. Однак вони можуть бути більш схильними до перешкод, таких як стіни або обмежена площа покриття сигналу;

Класифікація комп'ютерних мереж за технологією бездротового зв'язку (таблиця 1.3) включає декілька основних типів мереж, що використовують різні стандарти та протоколи бездротового зв'язку [33].

Ці різні типи бездротових мереж мають різні застосування та особливості, і вони широко використовуються в різних сферах, від домашнього використання до промислових та мобільних застосувань.

Таблиця 1.3 – Основні класи бездротових мереж

Клас	Опис
Wireless LAN (WLAN)	Використовує стандарт Wi-Fi (IEEE 802.11) для створення бездротових мереж, які дозволяють підключати комп'ютери, смартфони, планшети, принтери та інші пристрої до мережі без потреби в проводах. Вона широко використовується в домашніх, офісних та громадських мережах для забезпечення бездротового доступу до Інтернету та обміну даними між пристроями.
Wireless WAN (WWAN)	Використовує мобільні мережі, такі як 3G, 4G, LTE, для забезпечення бездротового доступу до Інтернету та мережевих послуг. Вона широко використовується для мобільного Інтернет-з'єднання на смартфонах, планшетах, ноутбуках та інших пристроях.
Wireless Mesh Network	Мережа складається з вузлів, які спілкуються один з одним безпосередньо або через інші вузли, утворюючи мережу з маршрутізацією, яка адаптується до змін у структурі мережі. Вона застосовується для створення бездротових мереж у великих містах, мереж датчиків та мереж між суден.
Wireless PAN (WPAN)	Бездротова мережа з особистого доступу, яка охоплює невелику область, таку як одне приміщення або особистий простір користувача. Найпоширеніший стандарт WPAN – Bluetooth, який використовується для бездротового з'єднання між пристроями, такими як гарнітури, клавіатури, миші та інші периферійні пристрої.

Класифікація комп'ютерних мереж за призначенням відображає їхню основну функціональність та роль у спілкуванні, обміні даними та доступі до ресурсів. Основні класи включають [30]:

– публічні мережі (Public Networks) – це мережі, які доступні для загального користування, такі як Інтернет. Вони надають можливість доступу до різноманітних ресурсів та послуг, включаючи веб-сайти, електронну пошту, соціальні мережі, онлайн-магазини тощо. Такі мережі

використовуються для глобального спілкування, обміну інформацією та доступу до світового об’єму знань та ресурсів;

– приватні мережі (Private Networks) – мережі, які створені для внутрішнього використання в організаціях, компаніях або приватних осіб і не доступні загальному публічному доступу. Цей тип мереж використовуються для внутрішнього обміну даними, спілкування та співпраці між співробітниками, підключеними пристроями та підсистемами;

– віртуальні приватні мережі (Virtual Private Networks, VPN) – це безпечний зв'язок між двома або більше пристроями або мережами через неприватну мережу, таку як Інтернет. Вони забезпечують зашифрований канал зв'язку, що дозволяє забезпечити конфіденційність та безпеку обміну даними. VPN використовуються для забезпечення безпечного доступу до корпоративних ресурсів з віддалених місць, з'єднання філіалів організацій та забезпечення захисту конфіденційної інформації від зовнішніх загроз;

– інтернет служби (Internet Services) – це мережі, які надають конкретні сервіси або функціональність через Інтернет. Вони можуть включати електронну пошту, стрімінгове відео, онлайн-ігри, хмарні сервіси та інші інтерактивні послуги. Інтернет-сервіси стали невід'ємною частиною сучасного життя, надаючи доступ до різноманітних ресурсів та послуг через глобальну мережу [15].

Ці аспекти комп'ютерних мереж важливі для розуміння їх структури, функціонування та управління.

Надійність передачі даних залежить від якості кабелів та їх монтажу. Використання якісних кабелів і правильний їх монтаж зменшує ймовірність пошкодження та запобігає перешкодженню сигналу. Класифікація кабелів наведена в таблиці 1.4.

Для локальних мереж рекомендується використовувати кабель типу вита пара, якщо не потрібна дуже висока швидкість передачі даних, а для

високошвидкісних мереж рекомендується використовувати оптоволоконний кабель. Важливо використовувати якісні з'єднувальні елементи та кабелі [30].

Таблиця 1.4 – Види кабелів

Вид	Опис	Недоліки
Вита пара	Найбільш поширений тип кабелю для локальних мереж. Забезпечує добрий баланс швидкості, надійності та ціни.	Чутливість до заломів та пошкоджень.
Коаксіальний кабель	Раніше широко використовувався, зараз менш поширений. Забезпечує високу швидкість та надійність	Складний в монтажі
Оптоволоконний кабель	Найнадійніший та найшвидший тип кабелю. Стійкий до перешкод та електромагнітного випромінювання.	Висока вартість

Негативний вплив електромагнітного поля може спричинити інтерференцію із сигналом. Екранування кабелів та застосування електромагнітно-сумісного обладнання допомагає запобігти цьому.

Фізичні пошкодження також впливають на надійність передачі даних. Руйнування кабелів під час будівельних або ремонтних робіт може призвести до втрати зв'язку. Розташування кабелів у захищених каналах або використання надійних кабельних каналів допомагає уникнути таких проблем. Кабелі повинні бути прокладені в безпечних місцях, захищених від механічних пошкоджень, вологи та електромагнітних перешкод. Необхідно регулярно проводити тестування та моніторинг каналів передачі даних.

Важливо зазначити, що надійність каналів передачі даних залежить не лише від вищеперелічених факторів, але й від умов експлуатації, кваліфікації персоналу, який обслуговує мережу.

Швидкодія та коректність обробки даних також впливає на роботу мережі. Цей фактор залежить від навантаження на мережу, адже чим більше користувачів і пристроїв підключено до мережі, тим більшою буде кількість пакетів даних, які необхідно обробити – це може призвести до зниження швидкості та збільшення затримок. Важливим є якість програмного

забезпечення, тому що використання надійних та оновлюваних програмних продуктів для мережевого управління та безпеки дозволяє уникнути вразливостей та забезпечити коректну обробку даних [14].

Продуктивність мережевого обладнання – це маршрутизатори, комутатори та інші мережеві пристрої, які повинні мати достатню потужність процесора, оперативної пам'яті та пропускну здатність для обробки потоку даних без затримок та помилок [30].

Умови, в яких знаходиться обладнання є дуже важливим, тому охолодження є ключовим елементом. Було розглянуто основні чинники від яких залежить справність роботи:

– ефективність системи охолодження – це забезпечення належних умов температури та вологості в серверних приміщеннях, що допомагає уникнути перегріву обладнання та підтримує його працездатність;

– робота кліматичного обладнання – регулярна перевірка та обслуговування систем охолодження та вентиляції гарантує їх ефективну роботу та надійність;

– керування температурою та вологістю – використання автоматизованих систем контролю та регулювання параметрів середовища, що допомагає уникнути проблем, пов'язаних із перегрівом або переохолодженням серверного обладнання.

Тому для забезпечення оптимальної швидкодії та коректної обробки даних в комп'ютерних мережах важливо уважно враховувати всі ці фактори та вживати необхідні заходи для їх оптимізації та підтримки.

1.2 Аналіз проблем збоїв мережевого обладнання.

Зважаючи на різноманітність комп'ютерних пристроїв та їх складових, існують різні проблеми, які можуть виникати і призводити до збоїв мережевого обладнання.

Розглянемо основні з них:

– проблеми з живленням;

– перегрівання обладнання;

– зависання;

– збої каналів зв’язку.

Живлення є життєво важливим аспектом для правильної роботи комп'ютерних систем [34]. Воно забезпечує енергію, необхідну для живлення всіх електронних компонентів, включаючи центральний процесор, жорсткий диск, відеокарти, пам'ять та інші пристрої. Дотримання нормального рівня електропостачання дозволяє комп'ютеру працювати без перебоїв і попереджує можливі пошкодження або втрату даних.

Основні компоненти живлення комп'ютера включають в себе: блок живлення (Power Supply Unit, PSU), стабілізатори напруги та блоки безперебійного живлення (ББЖ, Uninterruptible Power Supply, UPS).

Блок живлення відповідає за перетворення змінного струму зі стандартної домашньої розетки на постійний струм з відповідною напругою та струмом, необхідним для роботи всіх компонентів комп'ютера. Він має різні роз'єми та кабелі для підключення до материнської плати, жорсткого диска, відеокарти, процесора та інших пристроїв.

Стабілізатори напруги забезпечують стабільну напругу, що подається на блок живлення комп'ютера. Вони допомагають захистити компоненти від перевищення або падіння напруги, що може виникнути через перепади в електричній мережі.

Блоки безперебійного живлення (ББЖ) забезпечують живлення комп'ютера в разі відключення або перебоїв у електропостачанні. Вони зберігають енергію у внутрішній батареї, яка може використовуватися в разі потреби. UPS можуть мати різні часи автономної роботи, вимірювані у хвилинах або годинах, які визначають, як довго вони можуть забезпечити живлення безперебійно у разі відключення електропостачання.

Проблеми, що виникають з живленням можуть бути досить різноманітними та мати потенціал спричинити серйозні проблеми для комп'ютерних систем. Неправильне підключення до джерела електроживлення, таке як неправильно підключений кабель живлення до розетки або до комп'ютера, може призвести до збою в роботі системи. Перепади напруги, що можуть виникнути через різкі зміни напруги в електричній мережі, можуть викликати перегрівання кабелів або розетки, що може спричинити пожежу. Дефект або несправність блоку живлення також може призвести до збою в роботі системи, або навіть до пожежі в разі перегрівання.

Перевантаження електричних мереж виникає внаслідок підключення занадто великої кількості електричних пристроїв до однієї розетки або електричного кола. Це може призвести до пошкодження обладнання, втрати даних або навіть пожежі. Неякісне електроживлення, таке як перевантаження лінії живлення або переривання напруги, може призвести до неправильної роботи електронних пристроїв або навіть до їх пошкодження.

Пошкодження кабелів живлення може бути спричинено механічними пошкодженнями, зносом або неправильним використанням. Пошкоджені кабелі живлення можуть призвести до перерв в живленні та несправностей в роботі системи.

Пристрої комп’ютерних мереж (КМ) можуть виявляти несправності через зміни у постачанні електроенергії. Це може включати в себе напругові падіння або різкі коливання напруги, які можуть бути спричинені бурями,

перевантаженням мережі або іншими факторами. Такі перешкоди можуть призвести до несправностей у роботі компонентів КМ, включаючи мікропроцесори, жорсткі диски та інші електронні пристрої.

Перегрівання обладнання – це серйозна проблема, яка може виникнути з різних причин та мати серйозні наслідки для продуктивності та навіть життєвого циклу обладнання [6]. Перегрівання може стати результатом недостатньої вентиляції корпусу пристрою, забруднення вентиляторів, перевантаження компонентів, поганого контакту теплових роз'ємів або неправильного розташування пристрою.

Недостатня вентиляція може призвести до накопичення тепла всередині корпусу мережевого обладнання. Це може стати результатом поганої конструкції корпусу, обмеженої кількості вентиляційних отворів або блокування потоку повітря об'єктами навколо мережевого обладнання. Недостатня циркуляція повітря може призвести до перегріву компонентів, особливо таких як центральний процесор (CPU) та відеокарта, які виробляють багато тепла.

Забруднення вентиляторів також може призвести до перегріву мережевого обладнання. Пил та інші забруднення можуть накопичуватися на лопатках вентиляторів, перешкоджаючи їхньому нормальному функціонуванню та обмежуючи потік повітря. Це може призвести до збільшення температури всередині корпусу та перегріву компонентів.

Перенавантаження компонентів також може викликати перегрів. Завдання, які вимагають великих обчислювальних ресурсів, можуть збільшити температуру компонентів, таких як CPU та відеокарта. Це може призвести до перегріву цих компонентів та зниження їх продуктивності.

Поганий контакт теплових роз'ємів може також призвести до перегріву. Некоректний монтаж теплових роз'ємів між компонентами та системою охолодження може призвести до поганого відведення тепла. Це може збільшити температуру компонентів та призвести до їхнього перегріву.

Неправильне розташування мережевого обладнання також може сприяти перегріву. Розташування мережевого обладнання у тісних або закритих просторах, де немає достатньої циркуляції повітря, може призвести до підвищення температури навколо мережевого обладнання та перегріву компонентів.

Загальною мірою заходів для уникнення перегріву може бути регулярне обслуговування мережевого обладнання, забезпечення належної вентиляції та охолодження, а також уникнення перевантаження компонентів. Також важливо розміщувати мережевого обладнання у добре провітрюваному місці та уникати блокування вентиляційних отворів.

Зависання обладнання – це стан, коли операційна система або програмне забезпечення перестає реагувати на введення даних користувача та зависає. Це може статися з різних причин і мати різний характер, від тимчасового зависання програми до повного блокування операційної системи.

Однією з основних причин зависання мережевого обладнання є надмірне навантаження ресурсів системи. Коли програми використовують багато ресурсів, таких як CPU, пам'ять або диск, це може призвести до перенавантаження системи і блокування програм або операційної системи. Наприклад, запуск багатьох важких програм одночасно або виконання великих обчислень може спричинити зависання.

Іншою причиною може бути проблема з програмним забезпеченням. Погано написані програми або програми з помилками можуть викликати зависання системи під час їх роботи. Це може бути пов'язано з некоректним використанням ресурсів системи або неправильним управлінням пам'яттю.

Також зависання може бути спричинено конфліктом між програмами або пристроями. Коли дві або більше програми або пристрої намагаються використовувати одні й ті ж ресурси системи одночасно, це може призвести до блокування системи. Ще одна з причин може бути спричинена іншими

факторами, такими як неправильні налаштування операційної системи, віруси або шкідливе програмне забезпечення, або навіть проблеми з апаратним забезпеченням, такі як дефектні компоненти.

Збої каналів зв'язку – ця проблема виникає з різних причин та має різний характер. Далі розглянуто наступні основні причини: переривання зв'язку, інтерференція та шуми, помилки у конфігурації мережі, проблеми зі з'єднанням, відмова пристроїв а також проблеми з програмним забезпеченням. Їх опис дає змогу зрозуміти важливість справності каналів зв’язку.

Переривання зв'язку між різними мережевими пристроями є наслідком розриву кабелю, пошкодження роз'ємів або несправності мережевого обладнання. Часто воно призводить до тимчасової втрати з'єднання між комп'ютерами, серверами або іншими мережевими пристроями.

Виникнення інтерференції або шумів у каналах зв'язку може бути, наприклад, внаслідок електромагнітних перешкод або впливу інших мережевих пристроїв. Здебільшого це призводить до спотворення сигналу і порушення передачі даних, що в свою чергу стає причиною втрати пакетів і зниження швидкості передачі.

Помилки у конфігурації мережевого обладнання або програмного забезпечення є причиною різних проблем з підключенням до мережі та доступом до інтернету [28]. Ось деякі з найпоширеніших помилок:

– у випадку неправильної IP-адреса або маски підмережі, некоректного налаштування маршрутизатора за замовчуванням неможливо підключитися до інших пристроїв у мережі і отримати доступ до веб-сайтів;

– коли неправильні налаштування DNS стає неможливим отримати доступ до веб-сайтів за їхніми іменами і доменні імена не перетворються в IP-адреси;

– некоректно налаштоване мережеве обладнання призводить до збоїв в роботі та ускладнює підключення до мережі.

Проблеми зі з'єднанням, такі як неналежне закріплення або пошкодження роз'ємів Ethernet або інших мережевих кабелів, можуть спричинити збої каналів зв'язку або бути причиною неправильного з'єднання і збоїв зв'язку.

Відмова мережевих пристроїв, таких як маршрутизатори, комутатори або мережеві карти, спричиняє збої каналів зв'язку. Наприклад, відмова маршрутизатора може призвести до втрати доступу до мережі для всіх підключених пристроїв.

Неправильна робота мережевого програмного забезпечення або проблеми зі зв'язком є однією з основних причин збоїв каналів зв'язку. Це може статися, наприклад, якщо програмне забезпечення для мережевого аналізу або моніторингу викликає конфлікти з іншими програмами або мережевими пристроями.

Враховуючи ці проблеми та їх можливі наслідки, важливо вживати заходів для їх запобігання та вирішення, щоб забезпечити безперебійну та надійну роботу мережевого обладнання. Використання сучасних блоків безперебійного живлення є одним із вирішень вищеперерахованих проблем.

1.3 Огляд засобів безперебійного живлення та постановка задачі

Надійні блоки безперебійного живлення є важливою складовою інфраструктури комп'ютерних систем, особливо там, де неперервне живлення є критично важливим, наприклад, в центрах обробки даних, серверних кімнатах, медичних установах та важливих корпоративних мережах. Далі буде розглянуто кожну складову ББЖ.

Батарея є ключовою складовою ББЖ. Вона забезпечує живлення у разі відключення основного джерела електроживлення. Висока якість батареї

гарантує тривалість її роботи під час відключення, що може бути критично важливим для збереження даних та неперервної роботи системи.

Інвертор перетворює постійний струм з батареї на змінний струм, який використовується для живлення підключених пристроїв. Якість інвертора визначає стабільність напруги та частоти, а також ефективність конвертації електроенергії.

Стабілізатор напруги: деякі ББЖ мають вбудований стабілізатор напруги, який регулює вихідну напругу, щоб забезпечити стабільне живлення підключених пристроїв. Це дозволяє уникнути пошкоджень обладнання через високу або низьку напругу в електромережі.

Блок керування: багато сучасних ББЖ мають вбудоване програмне забезпечення для моніторингу стану батареї, навантаження та автоматичного вимикання пристроїв в разі тривалого відключення живлення. Це дозволяє зберегти заряд батареї та забезпечити безперебійну роботу системи [29].

Додаткові функції, такі як захист від перенапруги, захист від короткого замикання, функції енергозбереження та можливість підключення додаткових батарей для збільшення тривалості автономної роботи.

Фізична конструкція: ББЖ можуть мати різні форм-фактори та конструкції, що відповідають різним потребам користувачів. Вони можуть бути компактними настільними моделями або великими стійковими системами, які призначені для розміщення в серверних кімнатах.

Надійність та сервісна підтримка – при виборі ББЖ важливо звернути увагу на надійність виробника та наявність гарантії та сервісної підтримки. Це допоможе забезпечити безперебійне функціонування ББЖ та оперативне вирішення будь-яких проблем.

Розрізняють такі типи ББЖ:

– ББЖ з резервним ходом – найпростіший тип і на ринку в таких ББЖ доступна ціна. Недоліками є те, що перехід на живлення від акумулятора

відбувається з затримкою а також такі ББЖ не підходять для живлення обладнання, що є чутливим до перепадів напруги;

– лінійно-інтерактивні ББЖ (рисунок 1.1) забезпечують кращий захист від перепадів напруги, порівняно з ББЖ з резервним ходом і на ринку вони також за доступною ціною. Такі ББЖ не підходять для живлення потужного обладнання;

– ББЖ з подвійним перетворенням забезпечують найвищий рівень захисту від проблем з електропостачанням і підходять для живлення будь-якого обладнання. В таких ББЖ найвища вартість і вони є більш складними в облуговувані.

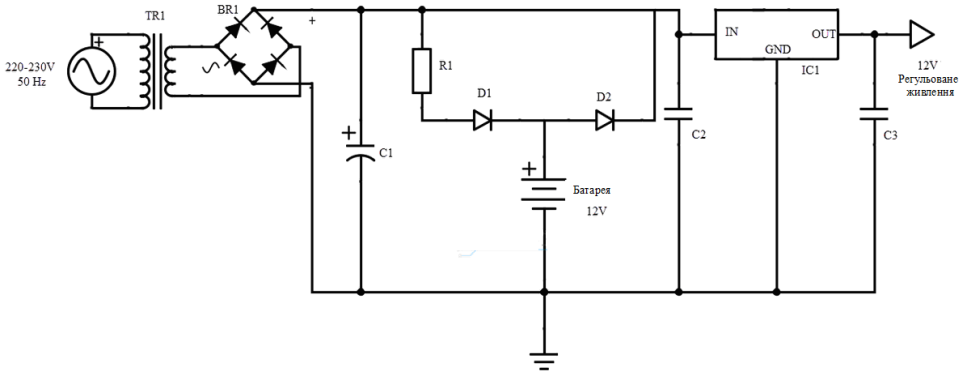


Рисунок 1.1 – Лінійно-інтерактивні ББЖ.

Лінійно-інтерактивні ББЖ є одним з найпоширеніших типів блоків безперебійного живлення і використовуються в більшості домашніх та офісних середовищ.

Принцип роботи лінійно-інтерактивних ББЖ є те, що вони постійно відстежують напругу в електромережі. Якщо напруга знижується або підвищується понад допустимі межі, ББЖ використовує вбудований автотрансформатор для корекції цієї напруги та забезпечення стабільного живлення підключених пристроїв. У випадку відключення основного джерела живлення, ББЖ автоматично переключається на резервний режим живлення з використанням енергії з акумуляторів.

Функціональні можливості лінійно-інтерактивних ББЖ є те, що вони зазвичай мають додаткові функції, такі як фільтрація напруги, захист від перенапруги та можливість моніторингу стану ББЖ через програмне забезпечення або інтерфейси користувача. Деякі моделі можуть мати розширені функції автоматизації, такі як автоматичне відновлення живлення після відновлення основного джерела живлення або можливість програмування режимів роботи.

Лінійно-інтерактивні ББЖ часто використовуються для захисту комп'ютерів, серверів, мережевого обладнання, систем відеоспостереження та інших електронних пристроїв в домашніх, офісних та невеликих комерційних середовищах. Вони дозволяють зберегти робочі дані та запобігти моливому збитку в разі відключення електропостачання.

Переваги лінійно-інтерактивних ББЖ включають високу ефективність, надійність та доступність за помірною ціною. Обмеження можуть включати обмежену потужність для більш великих систем та обмежені можливості автоматизації порівняно з деякими іншими типами ББЖ.

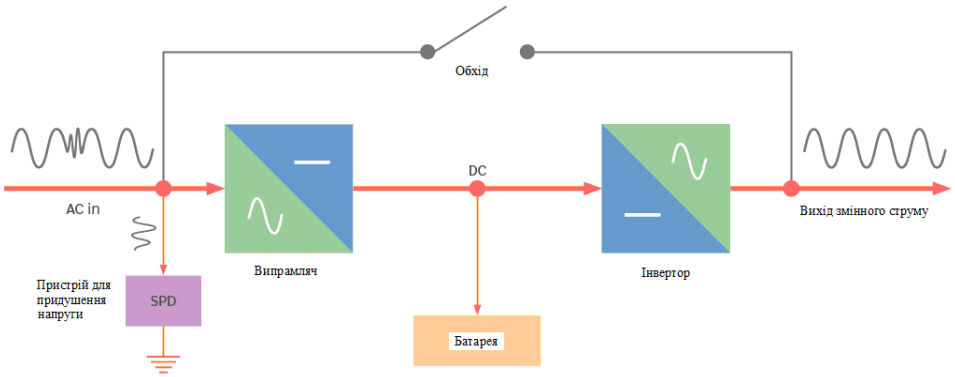


Рисунок 1.2 – Схема ББЖ з подвійним перетворенням

ББЖ з подвійним перетворенням (рисунок 1.2) є високотехнологічними системами, спроектованими для забезпечення безперебійного живлення критичних систем і обладнання, їх ще називають онлайн ББЖ. Далі буде висвітлено докладніші відомості про цей тип ББЖ.

ББЖ з подвійним перетворенням використовують два незалежні процеси, які працюють паралельно. Кожен процес контролює свої власні функції і робить свої власні розрахунки. У разі відмови одного з процесів другий може продовжувати працювати без перебоїв. Це забезпечує вищу надійність системи в порівнянні з іншими ББЖ, які мають лише один процес заряджання батареї.

Онлайн ББЖ можуть мати різноманітні функції, такі як автоматичне відновлення живлення після відновлення основного джерела живлення, захист від перенапруги та можливість програмування режимів роботи. Вони можуть також мати інтегровані системи моніторингу та управління, що дозволяють віддалено контролювати та керувати роботою ББЖ.

ББЖ з подвійним перетворенням широко використовуються в критичних системах, таких як дата-центри, фінансові установи, медичні заклади та інші сфери, де висока надійність є критичною. Вони можуть бути використані для захисту комп'ютерів, серверів, телекомунікаційного обладнання та інших важливих систем, де неперервна робота є обов'язковою.

Переваги онлайн ББЖ включають високу надійність, можливість продовження роботи при відмові одного з процесорів та розширені функціональні можливості. Однак вони можуть бути відносно дорогими та вимагати більш складного налаштування та обслуговування.

Онлайн ББЖ є одним із найбільш надійних та ефективних типів блоків безперебійного живлення, особливо в областях, де вимагається найвищий рівень захисту електронного обладнання[29].

Основний принцип роботи онлайн ББЖ полягає в тому, що підключене обладнання живиться напряму від внутрішнього інвертора ББЖ. Електроенергія з мережі живлення використовується для заряду акумуляторів і подальшого живлення обладнання через інвертор. Це означає, що електронне обладнання постійно живиться від внутрішнього інвертора, незалежно від того, чи є струм в електромережі чи ні. Це забезпечує

максимальний рівень захисту від перепадів напруги, перенапруг та інших відмов у мережі.

ББЖ з подвійним перетворенням зазвичай мають функції, такі як автоматичне відновлення живлення після відновлення основного джерела живлення, фільтрація напруги, захист від перенапруги та можливість програмування режимів роботи. Вони також можуть мати інтегровані системи моніторингу та управління, що дозволяють віддалено контролювати та керувати роботою UPS.

Онлайн ББЖ використовуються в критичних системах, таких як дата-центри, медичні установи, фінансові установи та інші високовимогливі ділові середовища. Вони є ідеальним рішенням для захисту серверів, мережевого обладнання, важливих даних та іншого обладнання, де будь-яка перерва у живленні може призвести до серйозних наслідків. Переваги онлайн ББЖ включають високий рівень надійності, неперервне живлення обладнання та розширені функціональні можливості. Недоліками є висока вартість та більший споживання електроенергії в порівнянні з іншими типами ББЖ.

ББЖ забезпечує різні типи захисту, щоб гарантувати безперебійне живлення та надійну роботу підключеного обладнання. Типи захисту, які може надати ББЖ: захист від перепадів напруги, захист від перенапруги, фільтрація шумів, захист від перевантажень, захист від короткого замикання, захист від викидів та електромагнітних перешкод (EMI).

Захист від перепадів напруги є одним з ключових функціональних елементів ББЖ. У разі перепадів або коливань напруги в електромережі, ББЖ виявляє зміни та автоматично реагує на них. У випадку зниження або підвищення напруги, ББЖ переключає живлення на внутрішню акумуляторну батарею або використовує свої вбудовані стабілізатори напруги, які автоматично коригують зміни напруги в електромережі. Це дозволяє забезпечити стабільний вихідний струм навіть при значних коливаннях напруги в електромережі.

Захист від перепадів напруги допомагає запобігти пошкодженню підключеного обладнання, так як воно не отримує великих коливань напруги, які можуть бути шкідливими для електронних пристроїв. Після того, як напруга в електромережі повернеться до нормального рівня, ББЖ автоматично переключиться на зовнішнє джерело живлення та продовжить живити підключене обладнання з нього – це забезпечить безперебійну роботу системи після перепаду напруги та допоможе уникнути цих проблем та зберегти інформацію та обладнання.

Захист від перенапруги допомагає запобігти пошкодженню підключеного обладнання, так як воно не отримує великих імпульсів напруги, які можуть бути шкідливими для електронних пристроїв. Перенапруга в електромережі виникає внаслідок великого імпульсу напруги, який перевищує нормальний рівень. Це може статися, наприклад, внаслідок різкого вимкнення електроустаткування або удару блискавки в електромережу.

ББЖ виявляє перенапруги у вхідній електромережі та автоматично відключає підключене обладнання від мережі, переключаючись на внутрішні джерела живлення, такі як акумуляторні батареї. Під час перенапруги, ББЖ фільтрує вхідну напругу та забезпечує стабільний вихідний струм, щоб захистити підключене обладнання від можливих пошкоджень. Після того, як перенапруга в електромережі завершиться і напруга повернеться до нормального рівня, ББЖ автоматично переключиться на зовнішнє джерело живлення та продовжить живити підключене обладнання з нього [29].

Захист від перенапруги

Фільтрація шумів – це процес очищення електричного струму від небажаних електромагнітних перешкод та інтерференції, що можуть виникати в електромережі. Для ефективної роботи електронних пристроїв, таких як комп'ютери, сервери та інші чутливі пристрої, необхідно, щоб живлення було стабільним і вільним від шумів. Електричні шуми можуть

бути викликані різними чинниками, такими як електромагнітні поля від інших електроприладів, комутаційні перешкоди в електромережі, електричні розряди або навіть перешкоди, що надходять через вхідні лінії живлення.

ББЖ використовує вбудовані фільтри (таблиця 1.5), що здатні виявляти та фільтрувати небажані шуми та перешкоди в електромережі. Ці фільтри можуть бути спроектовані для поглинання або блокування шумів на певних частотах, що дозволяє очищувати електричний струм від небажаних інтерференцій.

Таблиця 1.5 – Типи вбудованих фільтрів

Тип фільтра	**Принцип роботи**	**Ефективність**
LC-фільтри	Використовують комбінацію індуктивних (L) та конденсаторних (C) елементів для фільтрації шумів та перешкод. Індуктивність (L) використовується для створення індуктивного опору шумам, що мають високу частоту, тоді як конденсатор (C) блокує низькочастотні шуми, відповідно.	Вони ефективно фільтрують шуми на певних частотах, але можуть мати обмежену діапазон фільтрації.
Фільтри з гасінням частоти	Використовують гасіння на певних частотах, що дозволяє знизити амплітуду шумів на цих частотах.	Вони є ефективними у фільтрації шумів на конкретних частотах, але можуть бути менш ефективними у широкосмуговому спектрі шумів.
Активні фільтри	Використовують активні компоненти, такі як операційні підсилювачі, для підсилення або приглушення певних частот в електричній мережі.	Програмовані для фільтрації шумів на різних частотах і забезпечують вищу ступінь гнучкості порівняно з LC-фільтрами або фільтрами з гасінням частоти.

Захист від перевантажень спрямований на захист підключеного обладнання від потенційно небезпечних ситуацій перевищення електричного навантаження може зберегти підключене обладнання від пошкоджень, які можуть виникнути внаслідок перевантаження електричного струму.

Перевантаження в електричній системі виникає, коли потужність, яка використовується або вимагається підключеним обладнанням, перевищує максимальну допустиму потужність ББЖ або електричного ланцюга [32].

ББЖ мають вбудований механізм виявлення перевантажень, який моніторить електричний струм, що витрачається підключеним обладнанням. Якщо електричне навантаження перевищує максимально допустимий рівень, ББЖ може виконати декілька дій, таких як відключення від електромережі або перехід на резервний джерела живлення [29].

Захист від короткогозамикання призначений для запобігання можливим пошкодженням підключеного обладнання у разі виникнення короткого замикання в електричній мережі.

Захист від викидів та електромагнітних перешкод (EMI). Електромагнітні перешкоди можуть бути викликані різними факторами, такими як електромагнітні поля, радіочастоти, перешкоди в електромережі та інші джерела . ББЖ зазвичай має вбудований електромагнітний фільтр, який призначений для фільтрації шумів та перешкод, що можуть виникати в електричній мережі. Цей фільтр допомагає знизити рівень електромагнітних перешкод, які можуть впливати на роботу підключеного обладнання.

Отже, ББЖ пропонує широкий спектр функцій та можливостей для захисту електронного обладнання від різних електричних проблем. Недоліками є те, що ці системи є дорогими в придбанні та утриманні, особливо для малих організацій або домашніх користувачів. ББЖ зазвчиай має великі розміри та вагу, що робить його менш практичним для використання в обмежених просторах.

ББЖ використовує кілька методів для захисту від короткого замикання і забезпечення безпеки підключеного обладнання наведено у таблиці 1.6.

Таблиця 1.6 – Типи захисних функцій ББЖ

Назва	Принцип роботи
Система виявлення короткого замикання	Негайно реагує на зміни струму та напруги в електричній мережі. Ця система постійно моніторить електричний струм, що надходить в ББЖ, і виявляє будь-які відхилення, які можуть свідчити про коротке замикання.
Автоматичне відключення живлення	Після виявлення короткого замикання ББЖ автоматично відключає подачу електричного струму до підключеного обладнання. Це допомагає запобігти подальшому пошкодженню обладнання від великого струму, що може протікати через коротке замикання.
Ізоляція короткого замикання	Ізолює коротке замикання від підключеного обладнання, щоб запобігти передачі шкідливих струмів до обладнання. Це дозволяє захистити обладнання від можливих пошкоджень та забезпечити безпеку його роботи.
Повідомлення про стан	Надсилає повідомлення та сигнали операторам або системам моніторингу про виявлення короткого замикання. Це дозволяє операторам швидко реагувати на проблему та приймати необхідні заходи для її вирішення.

Корпоративні комп’ютерні мережі та мережі Інтернет-провайдерів можуть мати велику кількість географічно віддаленого мережевого обладнання. Додавання якісного ББЖ до кожного такого пристрою є дорого і недоцільно. Також може бути збій у самому ББЖ і він після розряджання батареї може не увімкнути електроживлення. А інколи навіть при ідеальній роботі ББЖ, мережеве обладнання може зависнути або некоректно працювати, а тоді буде необхідно посилати до нього людину-оператора, яка його вимкне і ввімкне. Тому необхідно розробити недорогу комп’ютерно-

інтегровану систему контролю стану та керування живленням мережевого обладнання, яка б могла перевмикати недорогий віддаленний мережевий пристрій в автоматичному режимі або по команді оператора.

2. ВИБІР ТА ПРОЕКТУВАННЯ КОНТРОЛЕРА КЕРУВАННЯ ЖИВЛЕННЯМ

2.1. Способи контролю стану мережевих пристроїв

Контроль стану мережевих пристроїв включає в себе різні методи і засоби для моніторингу, аналізу та керування їхньою роботою. В дипломній роботі буде розглянуто основні системи перевірки стану пристроїв та системи моніторингу.

Термін «WatchDog» перекладається як «Сторожевий Пес», а у контексті комп'ютерів відноситься до реалізації двох підсистем: сторожовий таймер (WatchDog Timer, WDT) і програмне забезпечення Watchdog (рисунок 2.1).. WDT та сторожове програмне забезпечення діють як охоронці стабільності системи. Вони автоматично виявляють та реагують на несправності, запобігають системним збоям, підвищують надійність системи та забезпечують безперебійну роботу, особливо для операційних систем (ОС) та критичних програм.

Рисунок 2.1 – Мікросхема сторожевого таймера P8-WDT24/PLC

Сторожовий таймер — це окрема апаратна схема (рисунок 2.2) або фрагмент коду, який працює незалежно від основної програми. Операційна система або основна програма (ОП), відповідає за періодичне «скидання» сторожового таймера (СТ) протягом певного періоду часу. Якщо ОС або ОП

аварійно завершує роботу, зависає або стикається з помилкою, яка не дозволяє їй скинути таймер протягом відведеного часу, лічильник CT переходить критичне значення та запускає попередньо запрограмовану дію.

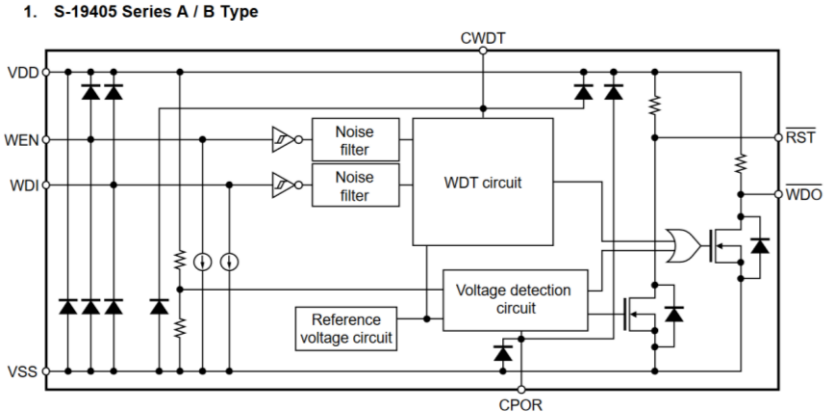


Рисунок. 2.2 – Схема сторожового таймера S-19405

Сторожовий таймер можна описати як лічильник, який починає відраховувати від нуля і до певного значення [13]. Якщо лічильник досягає певного значення, апаратне забезпечення WDT генерує повідомлення про необхідність скидання таймера [26]. Щоб уникнути перезавантаження системи або програми, ОС або ОП має вимкнути сторожовий таймер, тобто скинути лічильник до нуля.

Якщо програмне забезпечення зависло в нескінченному циклі, система не зможе відключити сторожовий таймер, тому лічильник досягаючи певного значення перезавантажує систему або програму.

Схема роботи WDT. Ця схема ілюструє основний принцип сторожового таймера. Основна програма повинна періодично скидати таймер протягом певного вікна часу. Якщо ОС або ОП це не вдається, сторожовий таймер запускає попередньо запрограмовану дію, наприклад перезавантаження системи, як показано на рисунку 2.3.

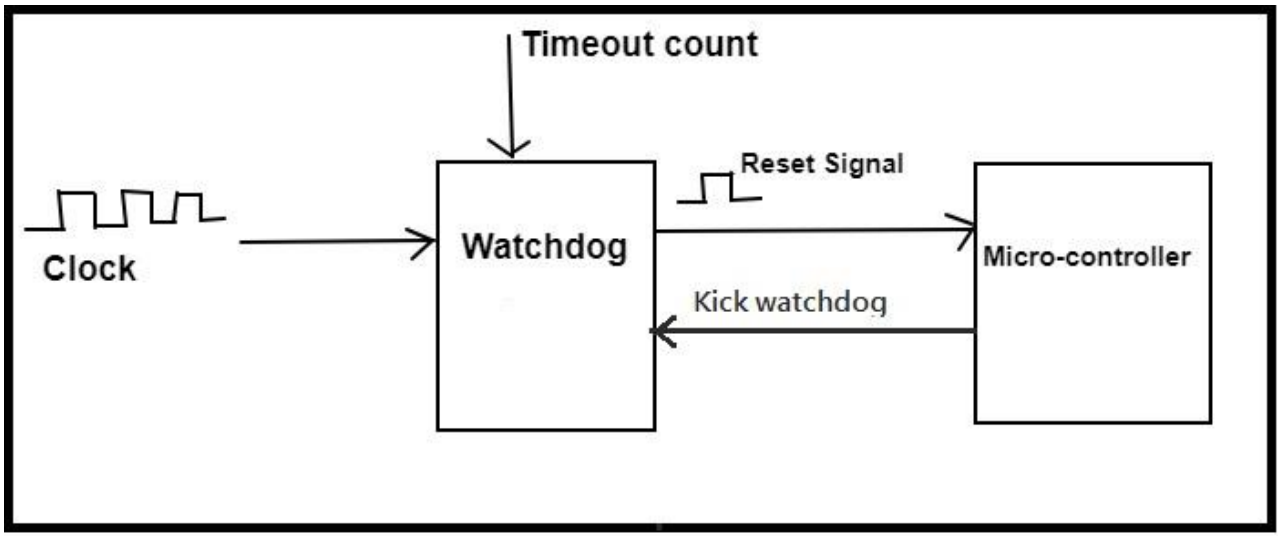


Рисунок. 2.3 – Схема алгоритму роботи сторожового таймера

Дії, що робить таймер WDT :

– перезавантаження системи – перезапускає програму або операційну систему, потенційно усуваючи проблему та дозволяючи системі відновитися;

– вхід у безпечний режим – система може перейти в безпечний режим із обмеженою функціональністю, що дозволяє усунути несправності або відновити дані;

– таймер надсилає попередження або повідомлення про помилку, щоб вказати на потенційну проблему.

Застосування сторожових таймерів досить широке. Найпоширініше їх використання у серверах і шасі блейд-серверів [26]. Також вони використовуються у вбудованих системах, таких як мікроконтролери та промислові системи керування, де несподівані збої можуть мати серйозні наслідки.

У мережевих пристроях для підтримки стабільності мережі, таких як: мережеві маршрутизатори, комутатори та брандмауери. Для критично-важливих систем безпеки, де збої можуть бути небезпечними, наприклад у

медичних приладах або системах управління авіацією, де WDT відіграють вирішальну роль у забезпеченні надійності системи.

Так у більшості серверів просте налаштування основних функцій сторожевого таймеру знаходиться у налаштуваннях системного БІОС. На риснку 2.4 показано БІОС сервера Tyan, в якому ми можемо включити або виключити функцію сторожового таймеру. Також там можна налаштувати яку дію виконувати при зависанні сервера – вимикати живлення чи перезавантажувати систему. А також можна обрати тривалість часу, яка має пройти щоб виконати цю дію автоматично – 5, 10, 15 чи 20 хвилин.

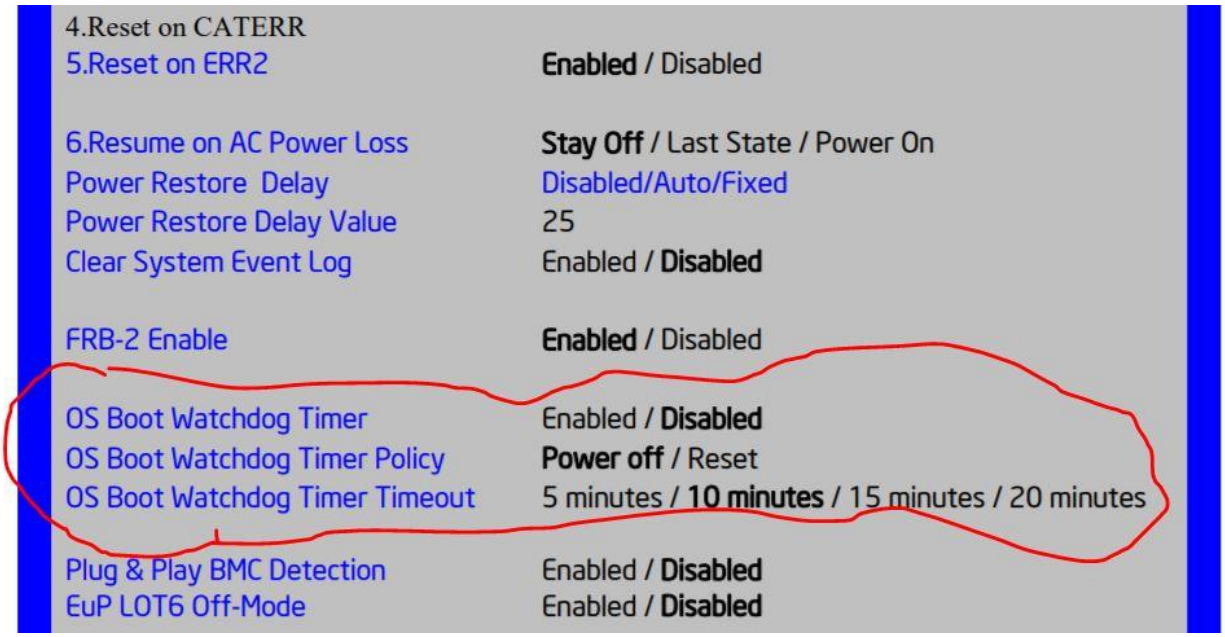


Рисунок 2.4 – Налаштування Watchdog сервера Tyan

Деякі Watchdog мають розширені функції такі як, налаштування прийнятного часу скидання таймеру у передбаченому діапазоні, що забезпечує більшу гнучкість для складних систем. Watchdog може відстежувати різні системні параметри, такі як використання ЦПУ, розподіл пам’яті, час відповіді програми та підключення до мережі. Вони можуть надсилати електронні листи, SMS або запускати візуальні чи звукові сповіщення, щоб сповістити адміністраторів про можливі проблеми.

Watchdog можна використовувати для виявлення та реагування на атаки DoS шляхом моніторингу використання системних ресурсів і перезапуску перевантажених служб.

Переваги сторожового таймеру:

– покращена стабільність системи. Вони автоматично виявляють та відновлюють несправності, що запобігає збоям системи та втраті даних;

– підвищена надійність. Таймери забезпечують працездатність критично важливих систем, автоматично перезапускаючи їх у разі збоїв;

– пвидке відновлення. Watchdog пропонують швидке відновлення порівняно з ручним втручанням після збою системи.

Крім апаратного Watchdog існують також програмні реалізації. Сторожове програмне забезпечення використовує інший підхід до захисту стабільності системи. Воно діє як програмне забезпечення, що працює в операційній системі, відстежуючи стан системи та швидкість реакції.

Сторожове програмне забезпечення (СПЗ) стежить за системними ресурсами, використанням ЦПУ, розподілом пам’яті, дисковим простором і мережевою активністю.

СПЗ відстежує скільки часу потрібно програмам, щоб відповісти на введені користувачем дані або запити. Уповільнення або зависання можуть вказувати на можливі проблеми. Користувач може визначити конкретні порогові значення для цих показників. Якщо значення перевищують ці пороги, програмне забезпечення позначає потенційну проблему.

Сторожове програмне забезпечення може запускати інструменти діагностики, щоб точно визначити причину проблеми, допомагаючи у вирішенні проблем. Прогама як і таймер може надсилати електронні листи, SMS або відображати сповіщення на екрані, щоб сповістити адміністраторів про можливі проблеми, які потребують уваги.

Переваги сторожового програмного забезпечення:

– гнучкість в порівнянні з апаратними сторожовими системами, програмне забезпечення пропонує більший функціонал у тому, що контролювати та як реагувати;

– налаштування користувачем програмного забезпечення відповідно до своїх конкретних потреб, визначивши спеціальні показники моніторингу та бажані дії;

– централізований моніторинг може контролювати декілька систем у мережі з центрального розташування, забезпечуючи цілісне уявлення про стан системи.

Архітектура побудови багаторівневого програмного сторожового таймера приведена на рисунку 2.5.

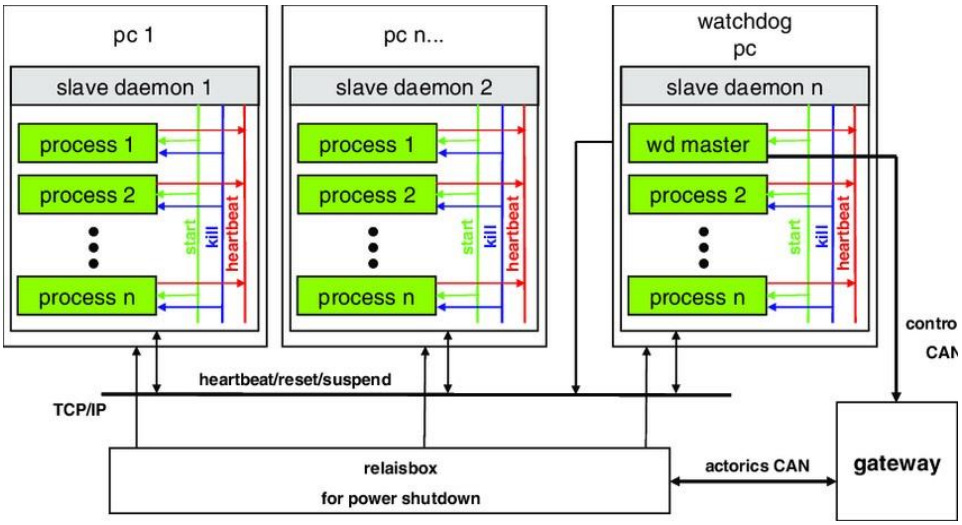


Рисунок 2.5 – Архітектура програмного сторожового таймера

Застосування програмного сторожового таймера досить широке для критично важливих серверів, на яких працюють бізнес-програми. Тут може бути корисним моніторинг програмних модулів для забезпечення стабільності та продуктивності. Також IT-спеціалісти можуть використовувати його для завчасного виявлення та вирішення проблем у мережі пристроїв. Прикладом є системи SCADA (наглядовий контроль і збір

даних) – ці системи контролюють промислові процеси. Програмне забезпечення Watchdog забезпечує безперебійну роботу, виявляючи проблеми та реагуючи на них.

Іншою системою контролю стану є IPMI (Intelligent Platform Management Interface), або інтелектуальний інтерфейс керування платформою. IPMI — це стандартизований набір специфікацій для апаратних систем керування платформами. Він пропонує можливості керування та моніторингу незалежно від центрального процесора хоста, мікропрограми (BIOS або UEFI) та операційної системи. За допомогою IPMI користувачі можуть підключатися до серверів через IP і отримувати доступ до таких функцій, як KVM через IP.

За допомогою IPMI можна віддалено керувати сервером та контролювати його роботу. Так проходить моніторинг фізичного стану обладнання, включаючи перевірку температури окремих компонентів системи, рівні напруги та швидкість обертання вентиляторів.

Відновлення працездатності сервера у автоматичному або ручному режимі, таке як віддалене перезавантаження системи, керування живленням, завантаження ISO-образів та оновлення програмного забезпечення. Можна керування периферійними пристроями, перевіряти журнал подій та зберігати інформацію про використовуване обладнання.

Більшість сучасних серверів оснащено технологією IPMI, безпосередньо вбудованою в материнську плату. Для старих серверів, у яких немає цієї функції, її можна додати як додатковий модуль.

Налаштування IPMI зазвичай передбачає доступ до налаштувань BIOS або UEFI. Для автентифікації користувачам знадобиться IP-адреса, а також ім’я користувача та пароль. Після налаштування IPMI дозволяє дистанційно контролювати та керувати серверами, навіть якщо вони офлайн із основною мережею.

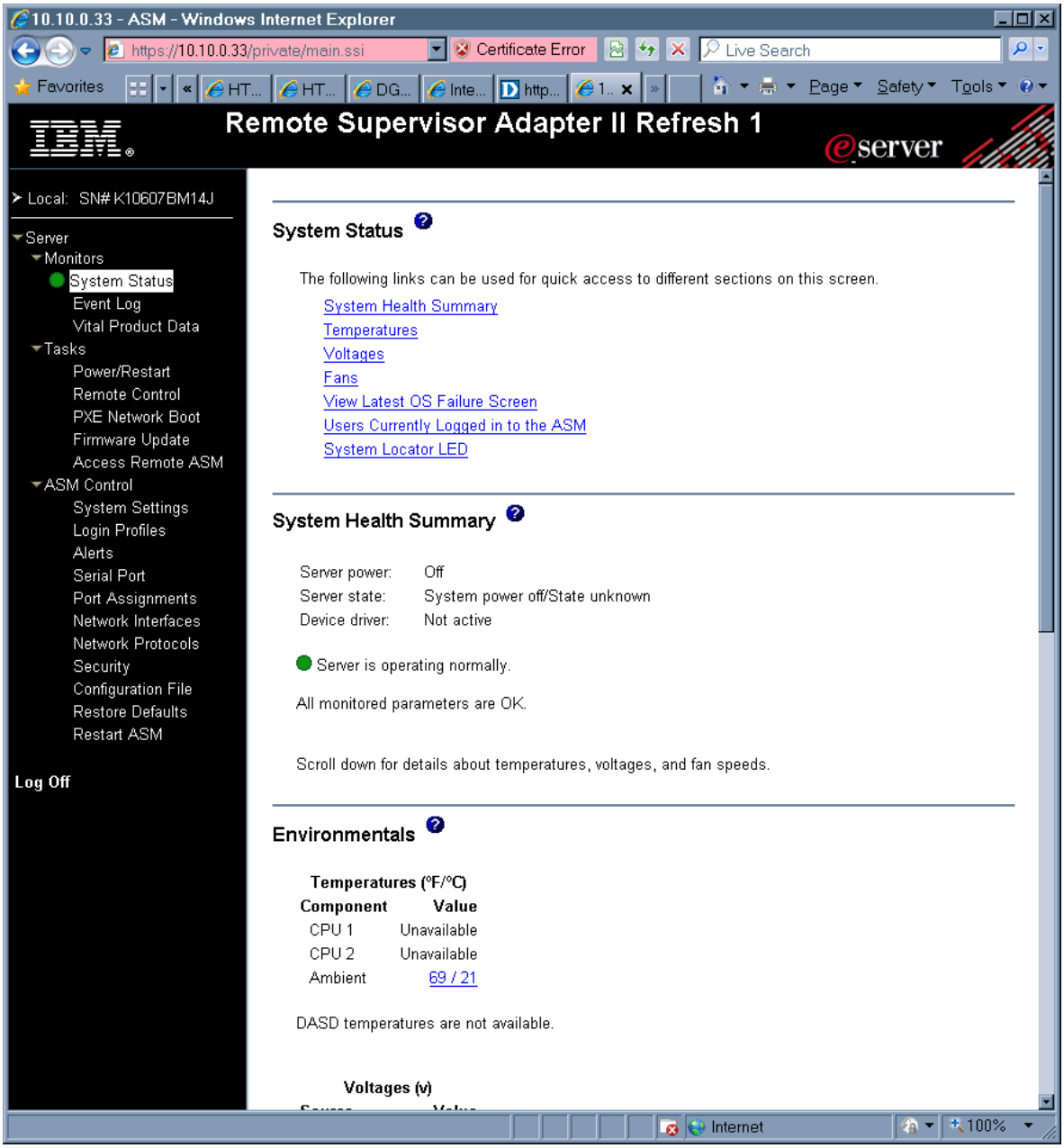


Рисунок 2.6 – Віддалений супервізор керування сервером IBM

Щоб керувати сервером через IPMI, можна використовувати різні методи. До них належать веб-доступ або спеціалізовані програми, такі як IPMI View, FreeIPMI та OpenIPMI [12]. Ці інструменти пропонують зручний інтерфейс для ефективного керування сервером.

Специфікація IPMI стандартизує інтерфейс спілкування, а не конкретну реалізацію в апаратній частині, тому IPMI не вимагає використання спеціальних запатентованих пристроїв і певних мікроконтролерів.

Виробники, дотримуючись специфікацій, розробляють власне IPMI обладнання, вбудоване в серверні платформи (таблиця 2.1 і рисунки 2.6–2.8):

Таблиця 2.1 – Виробники серверів та їх реалізації IPMI

Виробник	**Технологія на основі IPMI**
Cisco	Cisco IMC (Integrated Management Controller)
DELL	iDRAC (Integrated Dell Remote Access Card)
HP	iLO (Integrated Lights-Out)
IBM	IMM (Integrated Management Module)
Supermicro	SIM (Supermicro Intelligent Management)

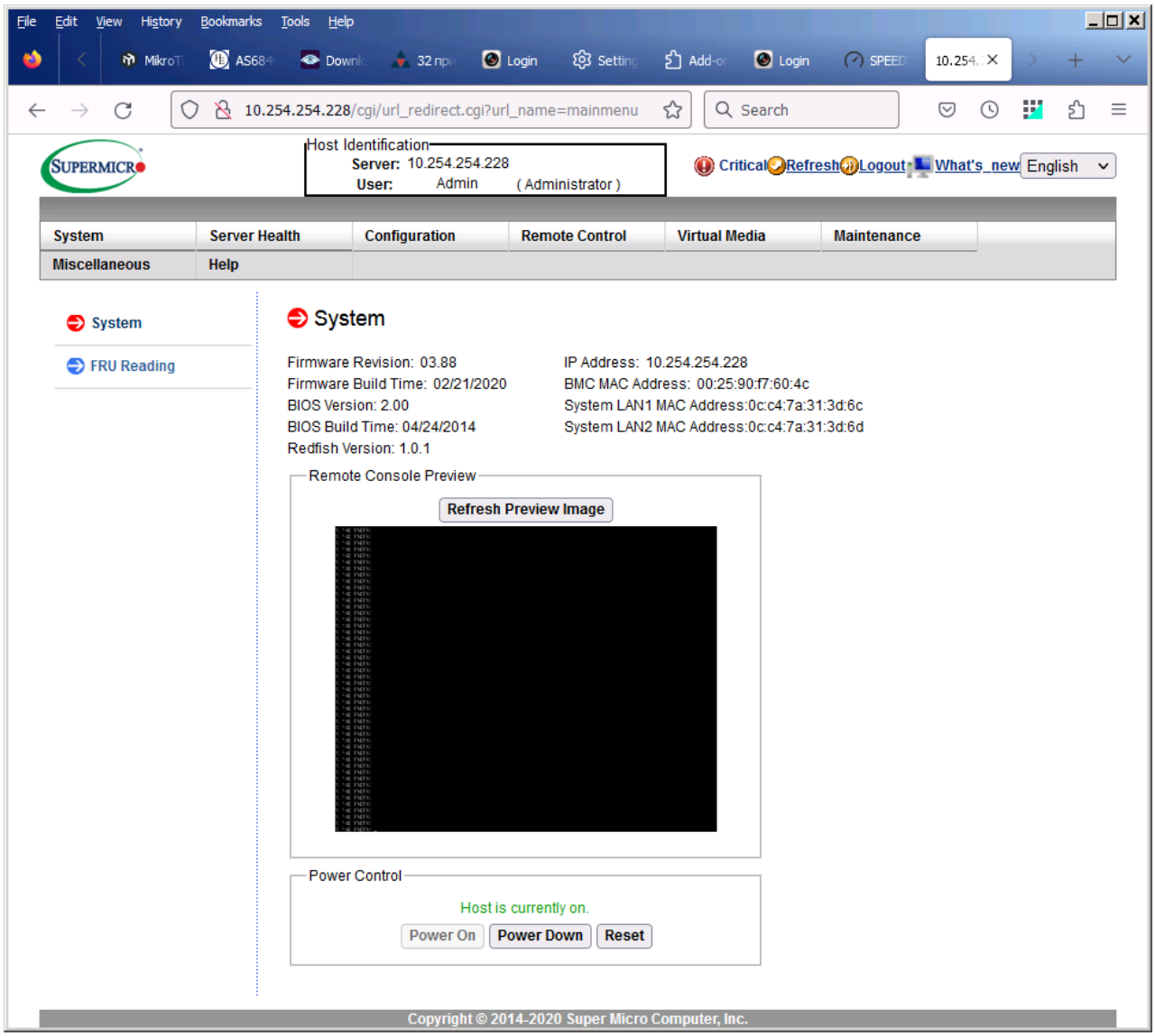


Рисунок 2.7 – Віддалене керування сервером Supermicro

Рисунок 2.8 – Віддалене керування ILO сервера HP

Рішення виробників відрізняються за такими основними характеристиками:

– ступенем візуалізації інформації про стан обладнання;

– унікальним набором програм для відновлення працездатності сервера у разі відмови комплектуючих;

– можливістю збирання статистики по всім комплектуючим сервера, включаючи ті, які підключені через карти розширення PCI, NVM і т.д;

– здатністю використовувати технологію не лише у серверному обладнанні, але і у звичайних комп'ютерах за допомогою плат розширення PCI-Express.

Основні компоненти IPMI зоюражені на рисунку 2.9. У центрі архітектури знаходиться "мозок" IPMI, який представлений мікроконтролером BMC (Baseboard Management Controller). Саме через нього здійснюється віддалене управління сервером. По суті, BMC – це окремий комп'ютер із власним програмним забезпеченням та мережним інтерфейсом. Він розміщений на материнській платі або підключається як плата розширення через шину управління PCI.

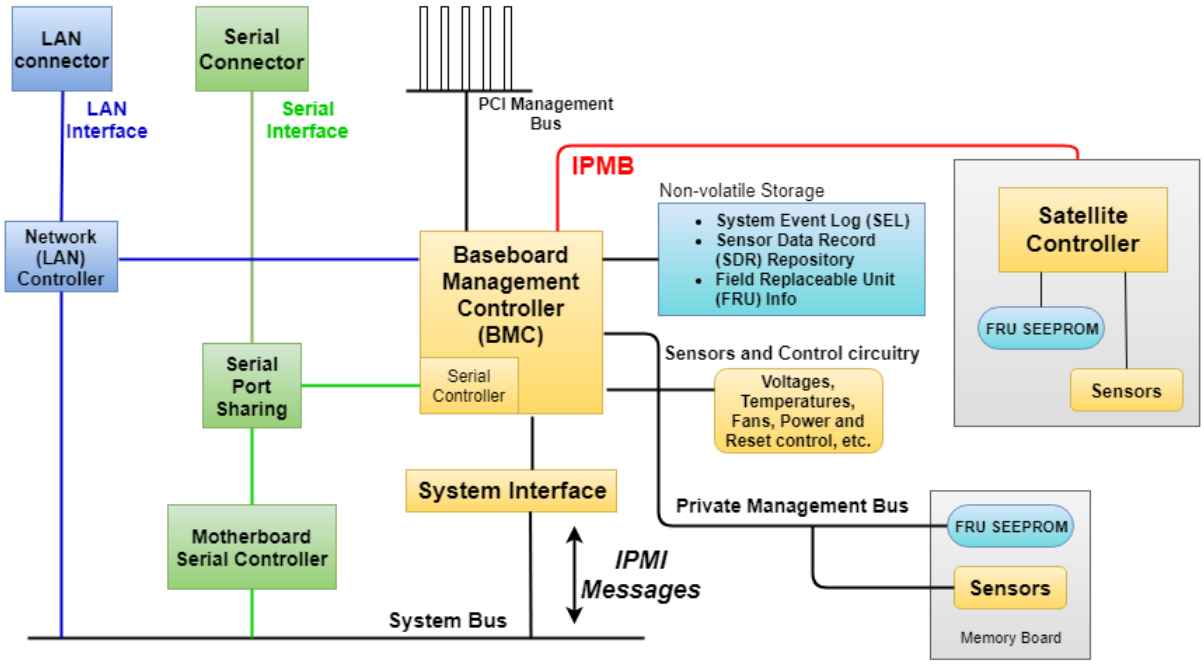


Рисунок 2.9 – Основні компоненти IPMI

BMC отримує живлення від основної напруги материнської плати, що дозволяє йому працювати незалежно від стану сервера(рисунок 2.10).

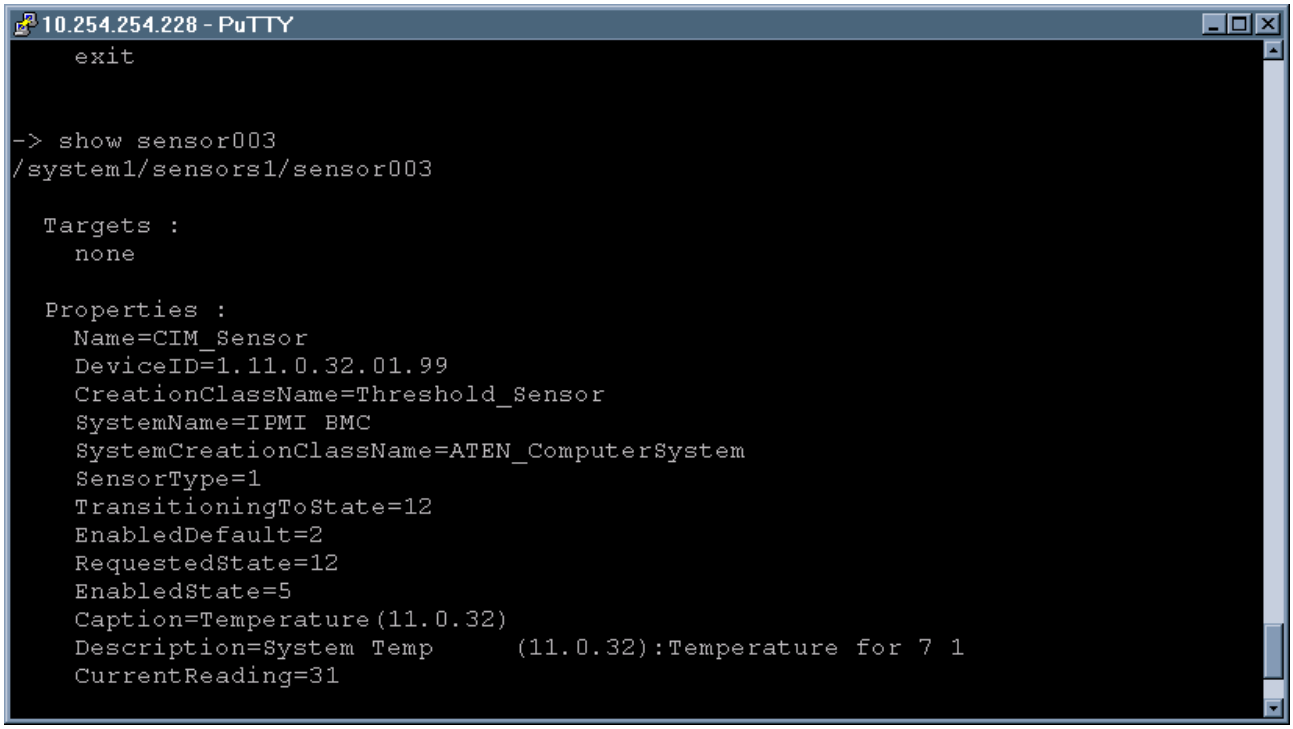


Рисунок 2.10 – Інформація про BMC

До BMC можна підключити додаткові контролери управління (Management Controllers, MCs) для того щоб розширити базові можливості управління [21]. Наприклад, в той час, коли основна система здійснює керування через BMC, то MCs призначені для моніторингу різних підсистем, таких як резервні джерела живлення, RAID-масиви, та периферійні пристрої.

MCs постачаються як самостійні плати, і вони працюють окремо від центрального BMC, тому їх часто називають супутниковими контролерами (Satellite Controllers). Кількість додаткових контролерів може бути різною, але центральний BMC завжди лише один.

Контролери підключаються до BMC через інтерфейс IPMB (Intelligent Platform Management Bus – шина інтелектуального керування платформою). IPMB базується на протоколі I2C (Inter-Integrated Circuit), за допомогою якого BMC перенаправляє команди управління до різних частин архітектури:

– Взаємодіє з додатковими контролерами (MCs).

– отримує дані від сенсорів (Sensors) (рисунок 2.11);

– звертається до енергонезалежного сховища (Non-Volatile Storage).

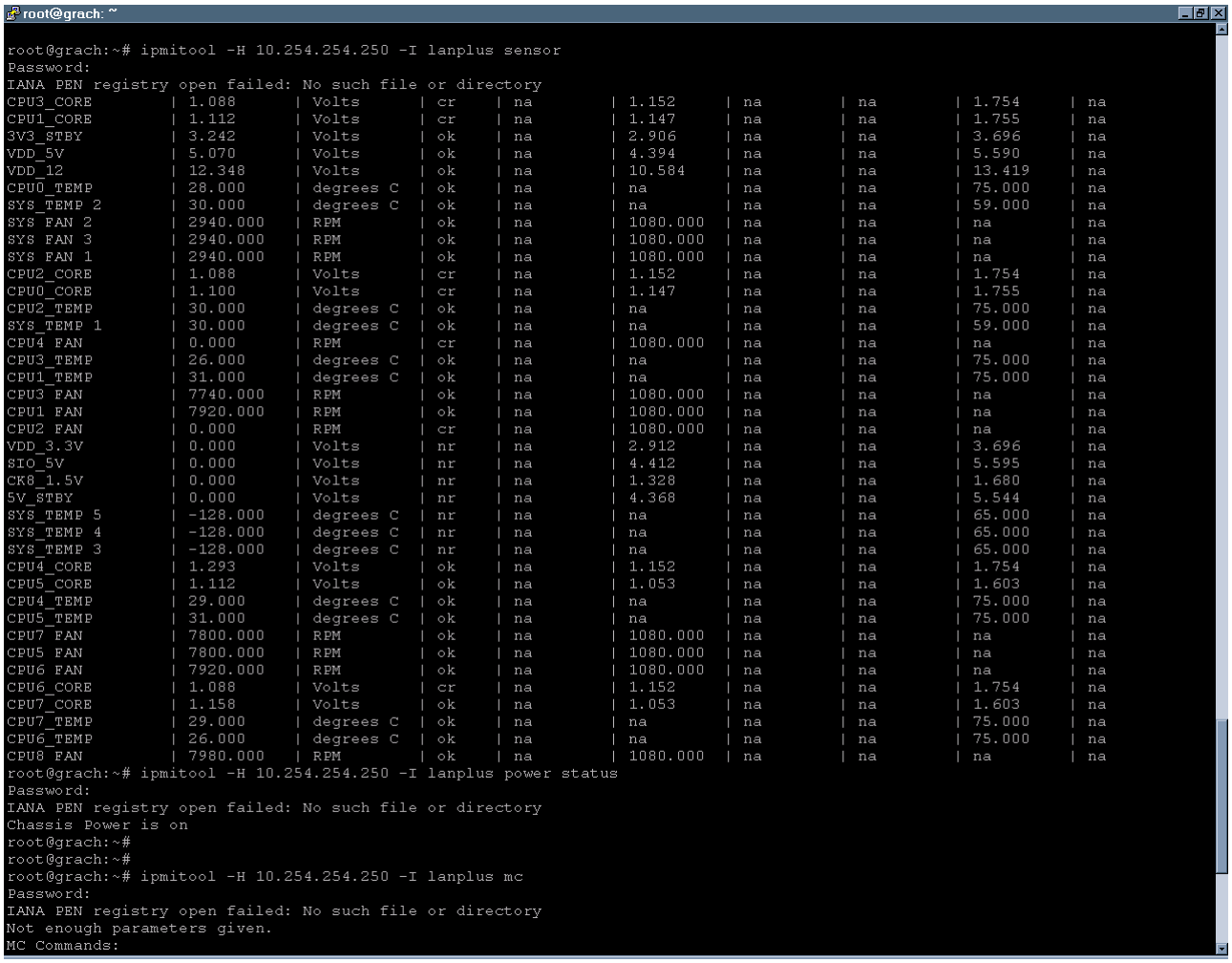


Рисунок 2.11 – Сенсори IPMI BMC

Архітектура IPMI забезпечує віддаленому адміністратору обмежений доступ до компонентів системи. Наприклад, для отримання даних з сенсорів, він відправляє команду на BMC, який, в свою чергу, взаємодіє з сенсорами.

Окрім передачі команд на BMC, можна налаштувати автоматичне виконання дій контролером за допомогою наступних механізмів: PEF (Platform Event Filtering), Watchdog Timer, Firmware Firewal.

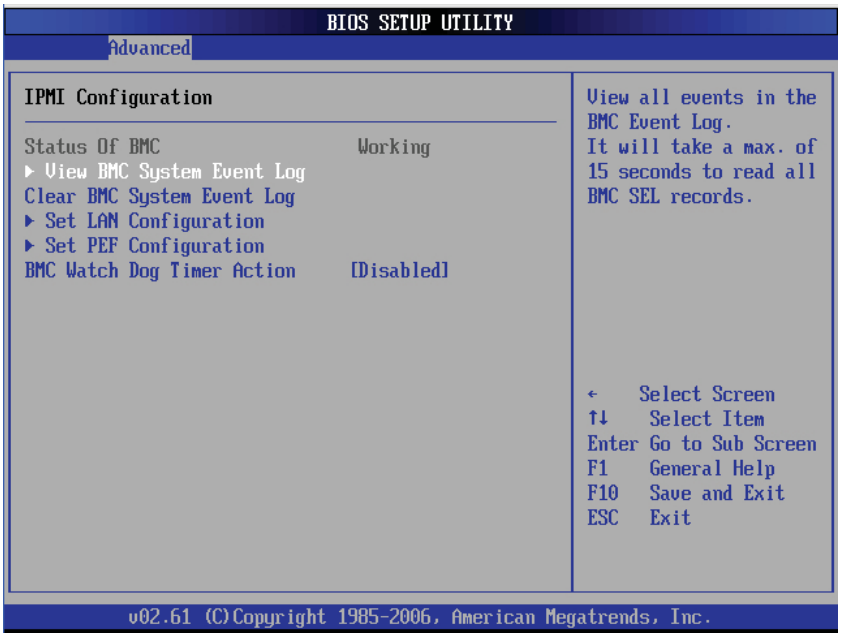


Рисунок 2.12 – Налаштування BMC у БІОС

У початковій версії IPMI віддалена консоль підключалася до модуля BMC через послідовний інтерфейс (Serial Interface). У специфікації IPMI v2.0 додано використання мережевого інтерфейсу (LAN Interface).

Мережевий інтерфейс LAN забезпечується через виділений мережевий порт BMC, що має свою IP-адресу. Під час передачі через LAN повідомлення IPMI проходять кілька етапів інкапсуляції (рисунок 2.13):

– повідомлення IPMI формуються в прикладні пакети;

– пакети IPMI Session інкапсулюються за допомогою протоколу RMCP (Remote Management Control Protocol);

– RMCP-пакети формуються в UDP datagrams;

– створюється IP-пакет;

– все поміщається в Ethernet-кадри.

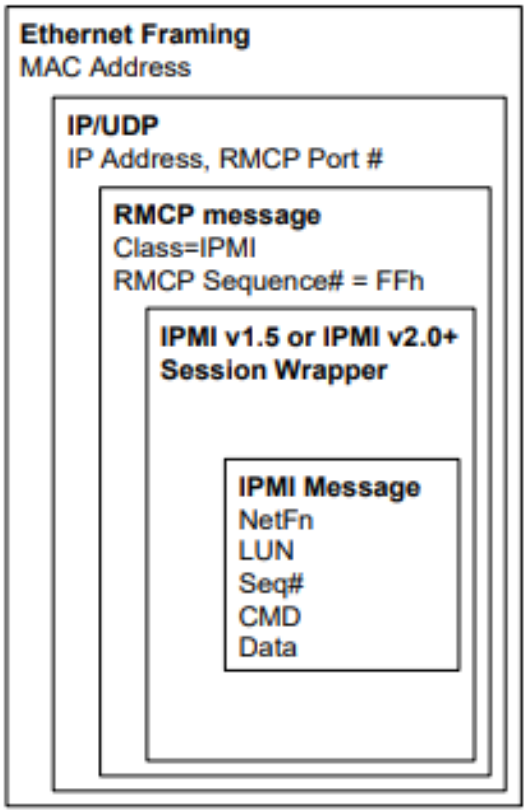


Рисунок 2.13 – Інкапсуляція прикладних пакетів IPMI

Недоліки IPMI. Збої в роботі IPMI можна поділити на чотири категорії:

– мережевий рівень: проблеми з портами, несправне обладнання, пошкоджені кабелі або недоліки у з'єднанні;

– програмний рівень: помилки програмного забезпечення, зависання модуля BMC, потреба у оновленні вбудованого програмного забезпечення модуля;

– апаратний рівень: перегрів, відмова критичних компонентів (наприклад, пам'яті, процесора), дефекти в архітектурі системи;

– енергозабезпечення: відключення живлення BMC або проблеми з блоком живлення сервера.

Наступним способом контролю стану є простий протокол керування мережею (SNMP), який є стандартним мережевим протоколом. Він використовується для керування та моніторингу пристроїв у мережах IP [5].

Він вбудований у багато різних пристроїв, таких як маршрутизатори, комутатори, концентратори, мости, повторювачі, шлюзи, сервери, брандмауери та бездротові точки доступу. Протокол надає можливість отримання доступу до них за допомогою їх IP-адреси. SNMP забезпечує стандартизований механізм для передачі керуючої інформації між пристроями у мережі LAN або WAN. SNMP входить до складу прикладного рівня моделі OSI [27].

Зазвичай, SNMP працює з використанням User Datagram Protocol (UDP). UDP є протоколом без підключення, що відрізняється від TCP тим, що не має механізму контролю передачі та відновлення з'єднання [22,23]. Він надсилає дейтаграми одержувачеві без перевірки, чи будуть вони отримані чи ні.

Інформаційні бази керування SNMP (MIB) визначають структуру даних, яку можна отримати або змінити на локальному пристрої [3,5]. Існують стандартні MIB, розроблені організаціями стандартизації, такими як IETF та ISO, а також пропрієтарні MIB, визначені окремими виробниками обладнання та постачальниками програмного забезпечення, такими як Cisco, Microsoft і Oracle.

Існує три основні версії протоколу SNMP наведені в таблиці 2.2:

Таблиця 2.2 – Основні версії протокола SNMP

SNMP версії 1 (SNMPv1	Перша версія протоколу, яка була впроваджена згідно зі специфікацією управління структурою, описаною в RFC 1157.
SNMP версії 2 (SNMPv2)	Розроблена з метою поліпшення ефективності обробки помилок. Початкова версія була випущена як RFC 1441, а пізніше була удосконалена та опублікована у RFC 1901. Часто її називають також SNMPv2c.
SNMP версії 3 (SNMPv3)	Випущена з метою підвищення рівня безпеки та конфіденційності передачі даних. Вона була представлена у RFC 3410.

Інструменти моніторингу SNMP:

- автоматично виявляє, контролює та керує мережевими пристроями;
- відстежує ключові показники продуктивності на рівні пристрою та інтерфейсу;
- SNMP дає змогу отримувати повне та детальне бачення продуктивності мережевого пристрою;
- системний адміністратор налаштовує порогові обмеження та створює сповіщення у разі аномалій.

SNMP працює, передаючи блоки даних, відомі як запити SNMP GET, на мережеві пристрої, які відповідають на ці запити [23]. Усі взаємодії відслідковуються, а інструменти моніторингу мережі використовують запити GET для збору даних через SNMP [31]. Трафік, що надходить у вашу мережу, може мати різне джерело, а SNMP спілкується з усіма пристроями у мережі.

SNMP попередньо налаштований на пристроях і після активації протоколу пристрої почнуть зберігати статистику продуктивності. Кожен мережевий сервер має кілька файлів бази інформації керування (MIB). Файли MIB пристрою запитуються для отримання моніторингових даних [9]. Робота SNMP базується на його компонентах, кожен з яких сприяє управлінню ресурсами.

Оскільки цей протокол є частиною набору TCP/IP, повідомлення SNMP об’єднуються та передаються через UDP [20,25].

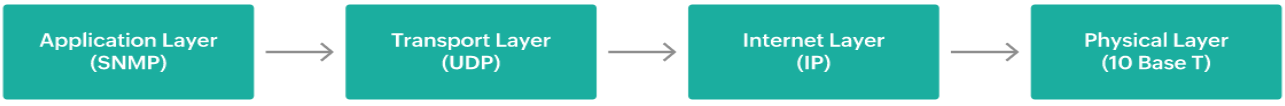


Рисунок 2.14 – Принцип передачі SNMP

Системи перевірки стану пристроїв, такі як Watchdog, IPMI та SNMP грають важливу роль у забезпеченні надійності і доступності мережевих пристроїв і серверів. Було розглянуто переваги та недоліки цих способів.

Перші обидві системи зазвичай реалізовані у дорогому мережевому або серверному обладнанні, тому в даній комп’ютерно-інтегрованій системі контролю стану та керування живленням мережевого обладнання буде використовуватись SNMP.

2.2. Вибір одноплатного комп’ютера контролером системи

Одноплатні комп'ютери (Single Board Computer, SBC) містять всі необхідні компоненти для повноцінної роботи комп'ютера. На одній платі знаходяться: процесор, графічний процесор, оперативна пам'ять та контролери вводу-виводу.

Одноплатні комп'ютери відрізняються від звичайних настільних комп'ютерів або ноутбуків, які можуть бути настільними або портативними і оновлюватися шляхом заміни процесорів і відеокарт або додаваням мікросхем оперативної пам'яті в спеціальні слоти. SBC не можна оновити так легко, як звичайні комп'ютери або повністю модульні ноутбуки.

Основні переваги одноплатних комп'ютерів:

– SBC є недороговартісними, що робить їх доступними для широкого кола користувачів;

– вони мають компактні розміри, що робить їх ідеальними для використання в обмежених просторах або вбудованих системах;

– більшість одноплатних комп’ютерів використовують енергоефективні компоненти, що робить їх зручними для використання;

– багато SBC мають велику спільноту користувачів та розробників, які надають підтримку, допомогу та розвивають велику кількість проектів та програмного забезпечення для цих платформ;

– SBC мають можливості для розширення.

Одноплатні комп'ютери можна поділити на дві основні категорії – пропріетарні і з відкритим кодом. Пропрієтарні SBC зазвичай призначені для використання в кінцевих продуктах або як макети для оцінки роботи. Вони часто виконані в індустріалізованих корпусах, які пройшли тестування, аналогічне тому, що вимагається від кінцевого продукту. Ці комп'ютери часто вбудовані в конструкцію кінцевих продуктів або є частиною сталої конфігурації.

SBC з відкритим кодом надають користувачам доступ як до апаратного дизайну, так і до компонентів, а також доступ до вихідного коду, який використовується на платі [17]. Це ідеально підходить для всіх користувачів, оскільки вони можуть легко зрозуміти, як працює програмне та апаратне забезпечення. Вони можуть змінити дизайн, щоб задовольнити власні вимоги до кінцевого продукту або просто дізнатися, як працює частина апаратного або програмного забезпечення.

Сучасні SBC постачаються з великою різноманітністю типів процесорів, більшість з них мають вбудовані графічні процесори. Ці процесори охоплюють широкий діапазон від процесорів на базі X86 традиційних ПК (AMD і Intel) до процесорів ARM, які використовуються в промисловості та у мобільних пристроях. Найпоширеніші операційні системи, що використовуються на одноплатних комп’ютерах – це Linux і його численні похідні версії, такі як Android, Ubuntu, Fedora, Debian і Arch Linux, а також FreeBSD і Windows CE.

SBC використовують як прототипи технологій, навчальні комп'ютери та вбудовані системи. Останнім часом спостерігається збільшення популярності одноплатних комп’ютерів завдяки прогресу в технологіях виготовлення. Майже кожного місяця анонсується новий SBC або нова версія існуючого, що свідчить про динамічний розвиток цього ринку.

Одноплатні комп'ютери широко використовуються в різних сферах, включаючи освіту, домашнє використання, промислові застосування та

наукові дослідження. Вони є популярними серед розробників через їх низьку вартість, доступність та можливості для розширення та налаштування.

Raspberry Pi – це лінійка одноплатних комп'ютерів, що першочергово була розроблених з метою підтримки викладання основ інформатики в школах і країнах, що розвиваються. Даний комп'ютер є розміром з кредитну карту, який можна підключити до телевізора та клавіатури (рисунок 2.15).

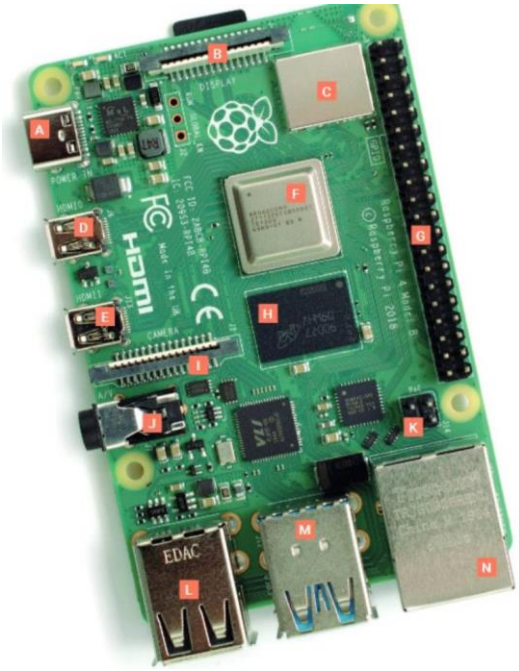


Рисунок 2.15 – Raspberry Pi 4 Model B

Raspberry Pi можна використовувати для багатьох завдань, таких як перегляд Інтернету, гра у відеоігри та редагування документів, хоча його потужності значно нижчі, ніж у звичайного настільного комп'ютера, а тому він може бути не таким ефективним для запуску важких програм. Проте на його основі можна реалізовувати різноманітні задачі, для яких звичайні ком'ютери будуть незручними.

Raspberry Pi надає широкий спектр програмного забезпечення та операційних систем (ОС), що відкриває перед користувачем безмежні можливості.

Операційна система Raspberry Pi (раніше відома як Raspbian) – це офіційна ОС, розроблена на основі Debian Linux і відома своєю зручністю, легкістю використання та підтримкою з боку Raspberry Pi Foundation [10]. Вона пропонується у трьох варіантах: Raspberry Pi OS Lite, Raspberry Pi OS з робочим столом, Raspberry Pi OS з настільним комп'ютером і рекомендованим програмним забезпеченням.

Raspberry Pi OS Lite – найпростіший варіант, що є ідеальним для досвідчених користувачів або додатків, які працюють без підключеного монітора

Raspberry Pi OS з робочим столом – варіант містить графічний інтерфейс користувача (GUI) для традиційного робочого досвіду.

Raspberry Pi OS з настільним комп'ютером і рекомендованим програмним забезпеченням – це версія містить попередньо завантажені стандартні програми, такі як веб-браузер, офісний пакет і медіаплеєри, щоб забезпечити повноцінний робочий стіл.

Варіанти апаратного виконання Raspberry Pi наведені в таблиці 2.3:

Таблиця 2.3 – Основні характеристики видів Raspberry Pi

Модель	CPU	RAM	Ціна
Raspberry Pi 4 Model B	1.5 GHz quad-core	2GB, 4GB, or 8GB	$55– $75
Raspberry Pi 3 Model B+	1.4 GHz quad-core	1GB	$35
Raspberry Pi Zero 2 W	1 GHz single-core	512MB microSD card	$15

Також великою перевагою Raspberry Pi є можливість під'єднання до нього різноманітних пристроїв, компонентів та периферійних пристроїв. До них можна віднести USB-пристрої, різні дисплеї, мережі Ethernet, бездротових мереж Wi-Fi та Bluetooth, HDMI, порти вводу-виводу загального призначення GPIO, відеокамери, аудіопристрої тощо. Повний перелік наведено у таблиці 2.4:

Таблиця 2.4 – Порти підключення периферійних пристроїв Raspberry Pi

Порт	Назва	Опис
A	Вхід живлення USB Type-C	Для живлення пристрою. USB Type-C є сучасним стандартом зарядки та передачі даних, який забезпечує швидку зарядку та підтримку різноманітних пристроїв.
B	Порт дисплея DSI	Дозволяє підключати дисплей за допомогою інтерфейсу DSI (Display Serial Interface), який забезпечує передачу відеосигналу високої якості.
C	Бездротовий / Bluetooth	Модуль бездротового зв'язку дозволяє підключатися до бездротових мереж Wi-Fi та Bluetooth для забезпечення доступу до Інтернету та підключення бездротових пристроїв.
D, E	Micro-HDMI 0 Micro-HDMI 1	Ці роз'єми призначені для підключення моніторів або телевізорів за допомогою кабелів HDMI. Вони дозволяють передавати високоякісний відеосигнал на зовнішні відображення.
F	System-on-chip	Це головний процесорний чип, який містить центральний процесор (CPU), графічний процесор (GPU) та інші важливі компоненти
G	GPIO-порти	Порти вводу-виводу загального призначення дозволяють підключати зовнішні пристрої та сенсори, а також керувати ними через програмне забезпечення.
H	RAM	Оперативна пам'ять пристрою, яка використовується для тимчасового зберігання даних та виконання програм.
I	CSI- порт для камери	Цей порт дозволяє підключати камеру до Raspberry Pi за допомогою інтерфейсу CSI (Camera Serial Interface)
J	3.5mm AV	Роз'єм, який дозволяє підключати аудіопристрої.
K	PoE	Роз'єм для підтримки технології Power over Ethernet (PoE), що дозволяє живити пристрій через мережевий кабель Ethernet.
L, M	USB 2.0, USB 3.0	Ці роз'єми призначені для підключення різноманітних периферійних пристроїв. Роз'єм USB 3.0 надає швидший обмін даними порівняно з USB 2.0.
N	Ethernet port	Для підключення пристрою до мережі.

Такий одноплатний комп’ютер повністю виконає усі поставлені на нього задачі, як основа головного контролера. Він хоч і є малопотужним, але його низька ціна і різноманіття функціоналу дозволять нам контролювати стан і керувати подачею живлення на бцдь-яке мережеве обладнання.

2.3. Проектування програмно-апаратного модуля

В основі програмно-апаратного модуля для контролю стану мережевого обладнання буде знаходитись найдешевша версія одноплатного комп’ютера Raspberry Pi Zero. До нього буде підключено через GPIO-порти кероване реле із нормально замкненими контактами (рисунок 2.16).

Даний програмно-апаратний модуль буде підключений до комп’терної мережі, де буде агентом менеджера системи моніторингу SNMP. Також він буде і частиною цієї системи, тобто виступати мініменеджером контролю стану мережевого обладнння. У разі, якщо мережевий пристрій, за яким ведеться контроль і моніторинг, починає некоректно працювати або зависає, то через реле йому відмикається електроживлення на деякий час [4]. Після включення живлення перевіряється стан мережевого пристрою.

Рисунок 2.16 – Кероване реле із нормально замкненими контактами

Кероване реле – це електронний модуль, який містить реле розмикання контактів та електронні компоненти для його керування. Кероване реле призначене для використання в різних електронних пристроях, які потребують керування різними навантаженнями за допомогою сигналів з мікроконтролерів або інших електронних пристроїв.

Керований модуль має один канал реле, що дозволяє керувати одним навантаженням. Цей модуль працює з напругою живлення 5В, тому його можна легко інтегрувати з різноманітними мікроконтролерами, такими як Arduino, Raspberry Pi та інші. Основні характеристики керованого реле описані в таблиці 2.5.

Таблиця 2.5 – Характеристики керованого реле

Тип живлення	DC
Кількість полюсів	1
Максимальний струм комутації	5A
Робоча температура	(-55°ДО 85°C)
Кількість перемикаючих контактів	1
Кількість замикаючих контактів	1
Кількість розмикальних контактів	3
Макс. напр. Комутації	220 AC
Тип контактора	1C
Макс. напр. Комутації	5DC
Номінальна напруга котушки	24В
Тип реле	DPDT

Кероване реле має два входи: вхід керування (IN) та вхід живлення (VCC), а також два виходи: вихід реле (COM) та вихід нормально замкнутого контакту (NO). Керування реле здійснюється за допомогою сигналу з мікроконтролера, що подається на вхід керування. При активації реле,

контакт COM переводиться в положення NO, і навантаження відключається від джерела живлення.

Цей модуль реле без оптоізоляції не має додаткових компонентів безпеки для захисту електронних пристроїв від можливих впливів, таких як перенапруга або перевантаження, тому може бути потрібна додаткова оптоізоляція для забезпечення надійного функціонування електронних пристроїв.

Для керування модулем реле одноплатний комп'ютер має порти вводу-виводу загального призначення (рисунок 2.17), що дозволяють підключати зовнішні пристрої та сенсори, а також керувати ними через програмне забезпечення. Такі порти на Rasberry Pi позначаються терміном GPIO (general-purpose input/output).

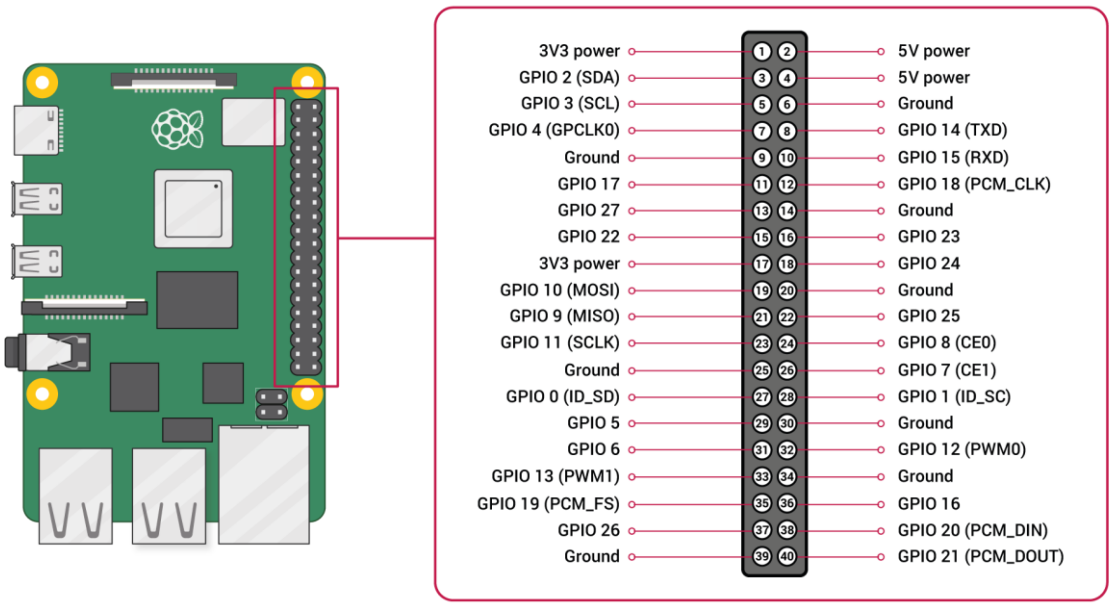


Рисунок 2.17 – Розміщення контактів GPIO на одноплатному Rasberry Pi

Загальні порти розміщені біля верхнього краю плати. 40-контактний роз'єм GPIO є на всіх поточних платах Raspberry Pi, хоча він не розпаяний на

Raspberry Pi Zero, Raspberry Pi Zero W і Raspberry Pi Zero 2 W. Роз'єми GPIO на всіх платах мають 0,1 дюйма (2,54 мм) крок шпильки.

На платі є два контакти 5 В і два контакти 3,3 В, а також вісім контактів заземлення (GND), які не можна переналаштувати. Решта штифтів є контактами загального призначення 3,3 В, тобто виходи встановлені на 3,3 В, а входи толерантні до 3,3 В. Схема нумерації контактів GPIO йде не по порядку номерів. Контакти GPIO 0 і 1 присутні на платі (фізично контакти з номерами 27 і 28), але зарезервовані для розширеного використання. Нумерація контактів наведена на рисунку 2.18.

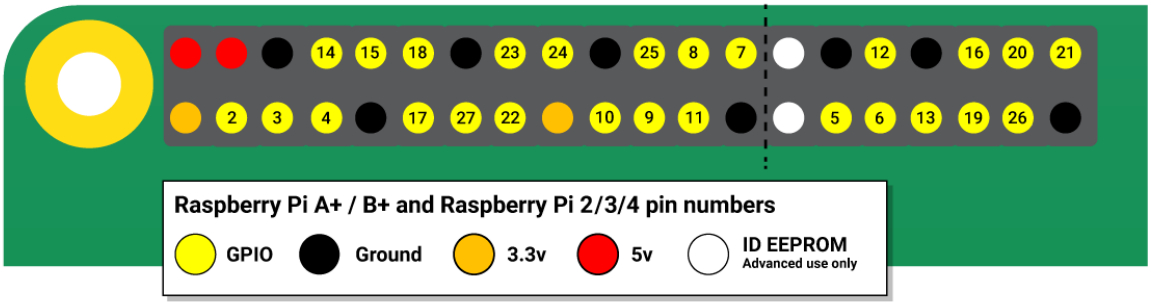


Рисунок 2.18 – Нумерація контактів GPIO

Будь-який з контактів GPIO можна позначити в програмному коді як вхідний або вихідний контакт і використовувати для широкого діапазону цілей. Для роз'єму GPIO, позначеного в програмному коді як вихідний контакт, можна встановити високий рівень сигналу (3,3 В) або низький рівень сигналу (0 В). А рівні сигналу контакту GPIO, позначеного як вхідний, можна зчитати програмним кодом як логічна одиниця (3,3 В) або логічний нуль (0 В). Це згладжується за допомогою внутрішніх резисторів підвищення або пониження. Виводи GPIO2 і GPIO3 мають фіксовані підвищуючі резистори, але для інших виводів це можна налаштувати програмно.

Окрім простих операцій вводу та виводу, контакти GPIO можна використовувати з різними альтернативними функціями, деякі доступні на всіх контактах, інші – на окремих контактах.

– PWM (Pulse-Width Modulation) або ШІМ (широтно-імпульсна модуляція)

- Програмна ШІМ доступна на всіх контактах
- Апаратна ШІМ доступна на GPIO12, GPIO13, GPIO18, GPIO19

– SPI (Serial Peripheral Interface) або ППІ (послідовний периферійний інтерфейс)

- SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
- SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)

– I2C (Inter-Integrated Circuit.) або МІС (міжінтегральна схема)

- Data: (GPIO2); Clock (GPIO3)
- EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)

– Serial або послідовний

- TX (GPIO14); RX (GPIO15)

Довідник GPIO можна отримати на Raspberry Pi, відкривши вікно терміналу та запустивши команду виводу номерів контактів “pinout”. Цей інструмент надається бібліотекою GPIO Zero Python, яка за замовчуванням встановлена в ОС Raspberry Pi.

Хоча підключення простих компонентів до контактів GPIO є абсолютно безпечним, важливо бути обережним при підключенні [11,14]. Світлодіоди повинні мати резистори для обмеження струму, що проходить через них. Не можна використовувати 5 В для компонентів 3,3 В. Не можна підключати двигуни безпосередньо до контактів GPIO, натомість необхідно використовувати схему Н-мосту або плату контролера двигуна.

Як видно із вищеописаного, незадіяними для підключення додаткових функцій є загалні порти: GPIO4, GPIO5, GPIO6 та від GPIO22 до GPIO27. Тому для керування модулем реле може бути використано один із цих 9 портів – наприклад GPIO4. Цей контакт буде з'єднано із входом керування на модулі реле IN. Так як для розмикання реле необхідна напруга 5 В, то вона буде подана на контакт VCC. Також необхідно з'єднати котакти GND на одноплатному комп'ютері та модулі реле.

Відповідно, на 40-піновому роз'ємі GPIO будуть задіяні контакти:

4 – VCC 5V

6 – GND

7 – Output control

Після програмування GPIO4 на Rasberry Pi можна буде керувати розмиканням реле [18].

3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНО-ІНТЕГРОВАНОЇ СИСТЕМИ

3.1. Функціональна схема програмного забезпечення

Основою системи контролю стану та керування живленням мережевих пристроїв буде стандартний менеджер SNMP HP OpenView. Це широко відома система моніторингу і управління корпоративними та великими комп'ютерними мережами. Для нашого котролера, реалізованого на основі одноплатго комп'ютера, було розроблено програмний модуль на мові Python, який виступає агентом SNMP і взаємодіє із менеджером SNMP, а у випадках коли немає з'єднання виконує процедуру перевірки стану мережевого пристрою і перемикає йому електроживлення при необхідності. Взаємодія між менеджером та агентами показана на рисунку 3.1.

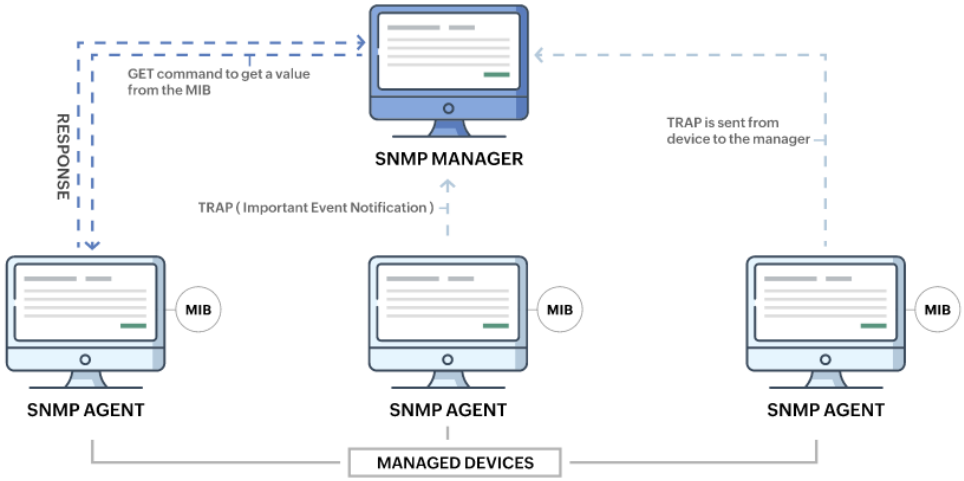


Рисунок 3.1 – Функціональна схема менеджера та агентів SNMP

OpenView є потужним рішенням з управління IT-інфраструктурою підприємства будь-якого розміру та напрямки діяльності. Побудовано на основі модульної архітектури. Надає широкі можливості з моніторингу та управління локальними обчислювальними мережами, серверними

платформами (такими як HP-UX, Solaris, AIX, Novell, Linux, весь спектр Windows платформ). Приклад мережі комутаторів зображено на рисунку 3.2.

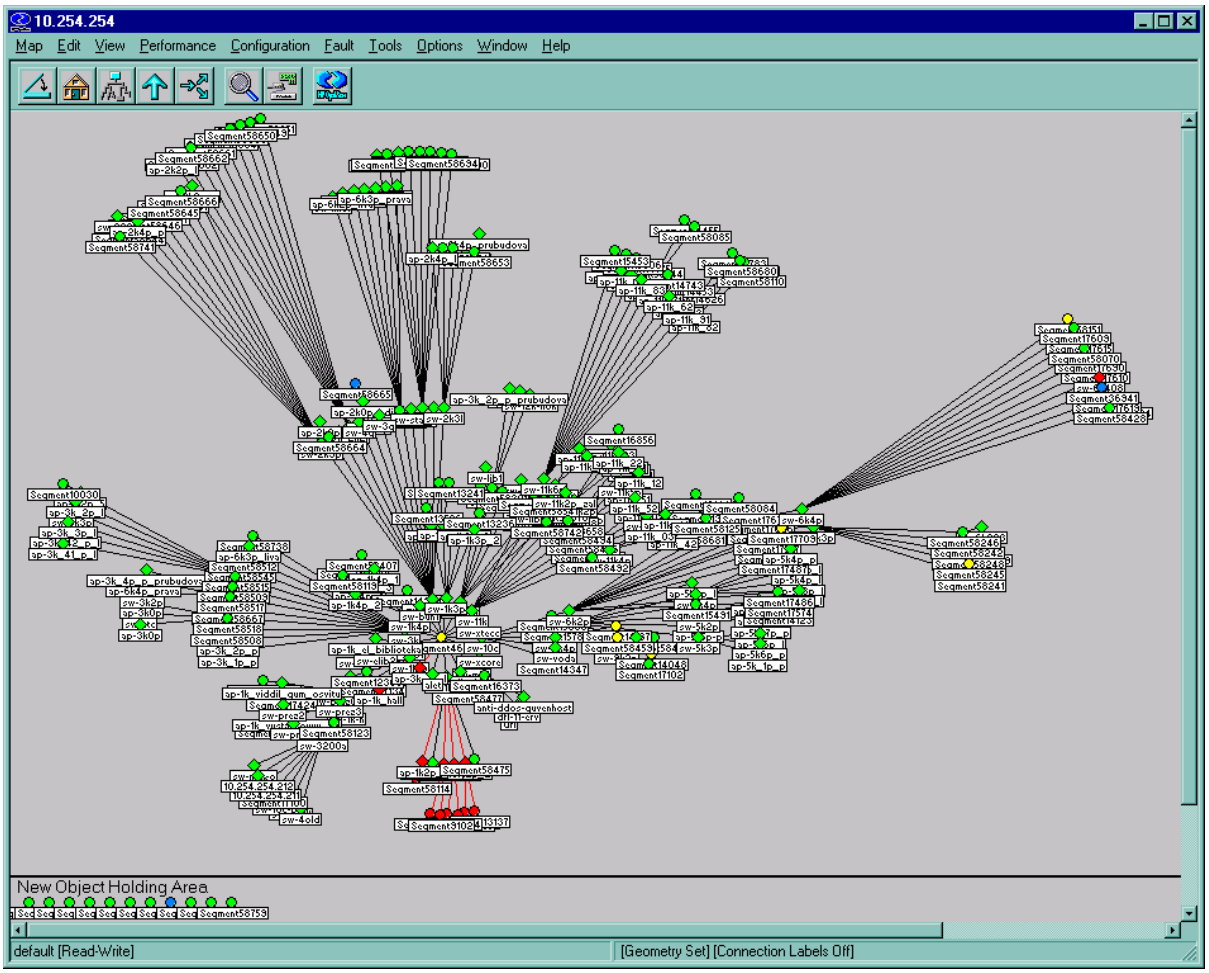


Рисунок 3.2 – Моніторинг мережі комутаторів в HP OpenView

Також цей менеджер SNMP дозволяє керувати додатками (SAP, Oracle, Sybase, MS SQL, Exchange, DB2, Informix, MS Active Directory), робочими місцями користувачів (інвентаризація, віддалена установка ОС, оновлень, програмного забезпечення, створення акаунтів користувачів, контроль за використанням ПЗ), організація диспетчерської служби, надання інструментів для вибудовування IT інфраструктури згідно процесів ITIL / ITSM. Більше 50 програмних продуктів, що вирішують найрізноманітніші завдання від резервного копіювання до моніторингу стану бізнес процесів в реальному часі.

Менеджер SNMP — це центральна система, яка використовується для моніторингу мережі SNMP. Також відомий як станція керування мережею (Network Management System, NMS), менеджер SNMP відповідає за зв’язок із мережевими пристроями, на яких реалізовано агент SNMP. Він працює на хості в мережі. Менеджер SNMP запитує агентів, отримує відповіді, встановлює змінні та підтверджує події від агентів.

Менеджер SNMP – це програмна система, яка використовує SNMP для збору даних щодо управління несправностями, моніторингу продуктивності та планування ресурсів[7].

Керований мережевий пристрій (Managed network device, MND) – це мережевий об'єкт із підтримкою SNMP яким керує менеджер SNMP. Зазвичай це маршрутизатори, комутатори, концентратори, мости, повторювачі, шлюзи, сервери, брандмауери, принтери та бездротові точки доступу.

Агент SNMP – це програмний модуль, який відіграє ключову роль у керуванні мережею. Він відповідає на запити від менеджерів SNMP, щоб надати інформацію про статус і статистику мережевого вузла. Агент SNMP розташовується локально на мережевому пристрої, від якого він збирає, зберігає та передає дані моніторингу менеджеру SNMP (рисунок 3.3)..

Мнеджер SNMP може регулярно надсилати запити до всіх пристроїв, щоб перевіряти їх стан. Але і в свою чергу можна налаштовувати агенти SNMP, щоб вони відсилали повідомлення у разі зміни їх стану. Такі повідомлення називаються пастками (traps).

Пастки SNMP надсилаються в певному форматі, показуючи час, ідентифікатор і значення. Час показує, коли сталася помилка, а ідентифікатор походить із MIB і називається OID.

Порт перехоплення SNMP – це порт, на який менеджер отримує повідомлення-пастки. Цей порт зазвичай має номер 162. Однак можна змінити цей порт, якщо потрібно. Однією із складних задач при перехопленні

SNMP є те, що вони не завжди ефективні для сповіщення про серйозні помилки.

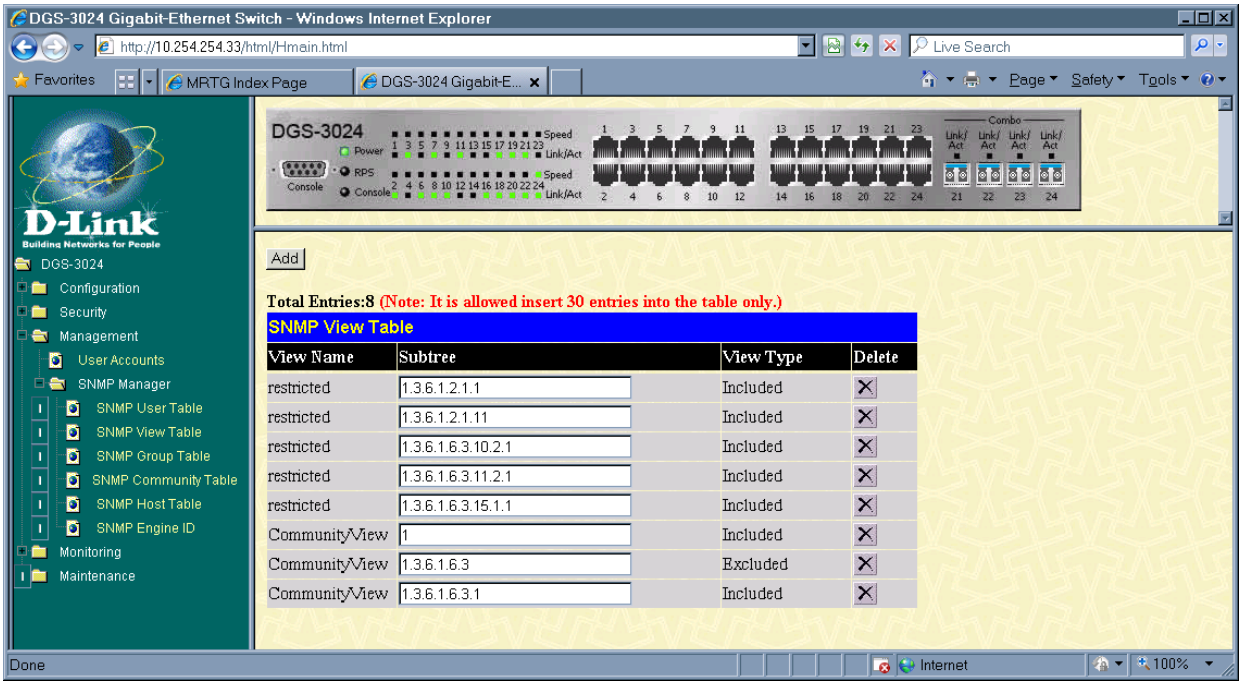


Рисунок 3.3 – Вбудований в комутатор SNMP-агент

Іноді агент мережевого пристрою надсилає перехоплення SNMP для незначної проблеми та пропускає серйозну проблему, яка може вивести з ладу всю комп'ютерну мережу. Таке відбувається, якщо на мережевому пристрої виникне фатальна проблема, яка вимикає весь пристрій, агент SNMP також більше не зможе працювати і перехоплення SNMP не надсилатиметься.

А тому, коли мережевий пристрій зависає або некоректно працює, то відповідно і не працює його агент SNMP. Саме для таких випадків, коли є критичним постійна робота такого мережевого пристрою або він далеко географічно віддалений, потрібен наш контролер стану із своїм власним агентом SNMP, який зможе перезавантажити керований ним пристрій. Також деякі мережеві пристрої не мають власного вбудованого SNMP, тому для них також підійде наш котролер.

3.2. Структура керованої інформаційної бази даних MIB

MIB є необхідною складовою моделей управління мережею. SNMP MIB визначає структуру обміну інформацією у системі SNMP [24]. Кожен агент SNMP утримує інформаційну базу даних, яка описує параметри керованого мережевого пристрою. Менеджер SNMP зберігає зібрані дані у MIB як спільну базу даних між агентом і менеджером.

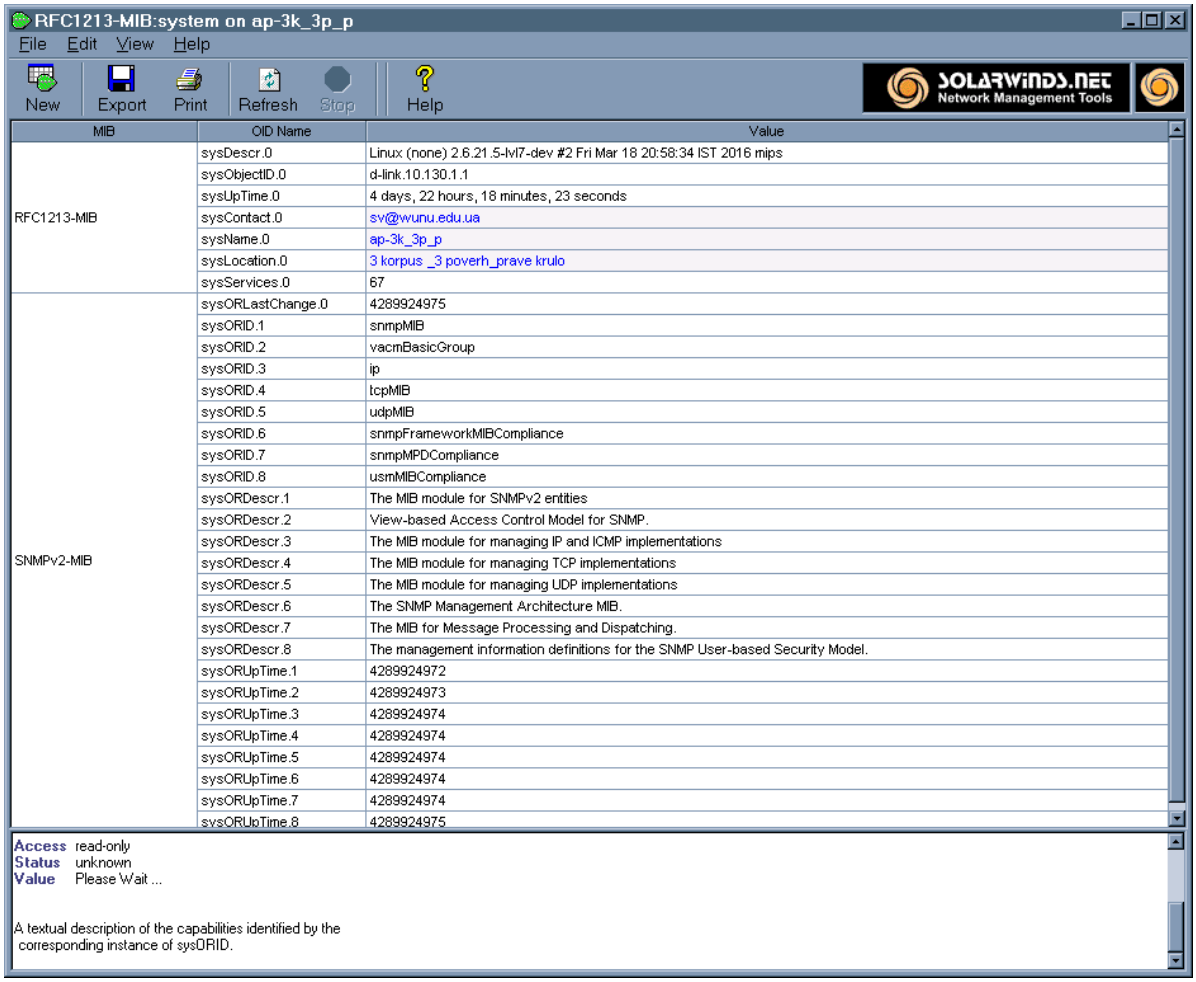


Рисунок 3.4 – Структура керованої інформаційної бази даних MIB

Файли MIB зберігаються у текстовому форматі, який розуміють редактори MIB, конструктори агентів SNMP, інструменти управління мережею та інструменти моделювання мережі, що спрощує розробку, тестування, розгортання та управління мережею. Керовані об’єкти у файлах

MIB ідентифікуються як ідентифікатори об’єктів (OID) (рисунок 3.4 та додаток 2).

MIB організовуються у вигляді дерева OID з ієрархічною структурою, що містить усі керовані функції продуктів. Кожна гілка дерева має унікальний номер і назву, а кожна точка у цьому дереві вказує на конкретний об'єкт і має свій власний унікальний ідентифікатор, який складається з повного шляху від кореня дерева до цієї точки. OID представляє елемент пристрою, який контролюється, наприклад температуру, функцію ЦП або пам’ять або навіть те, чи закінчується чорнило в принтері.

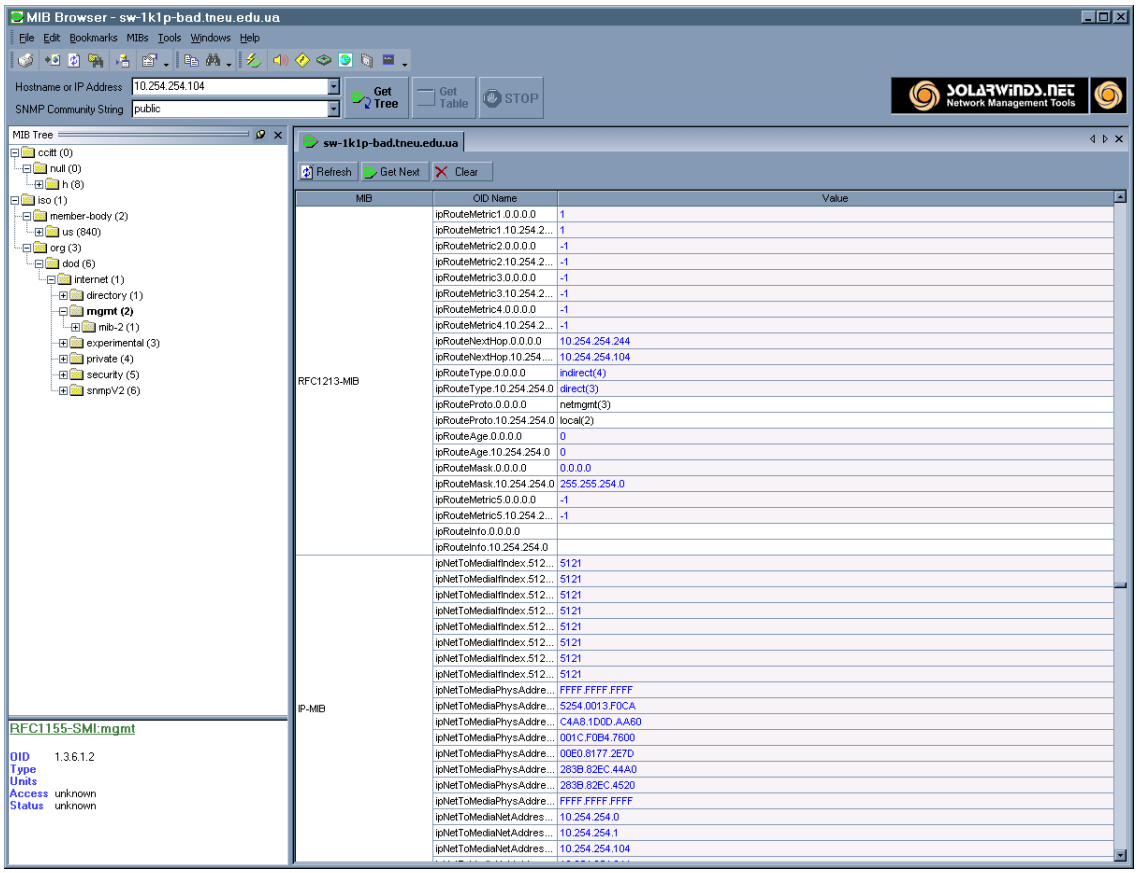


Рисунок 3.5 – Організація MIB у вигляді дерева OID

SNMP OIDs можна ідентифікувати за допомогою рядків чисел, розділених крапками. Дерево MIB зображено на рисунку 3.6.

В системі SNMP існують два типи керованих об'єктів:

– скалярні об'єкти, які характеризуються лише одним екземпляром об'єкта, тобто може існувати лише одне значення;

– табличні об'єкти, що складаються з кількох пов'язаних екземплярів об'єктів, організованих у вигляді таблиці MIB.

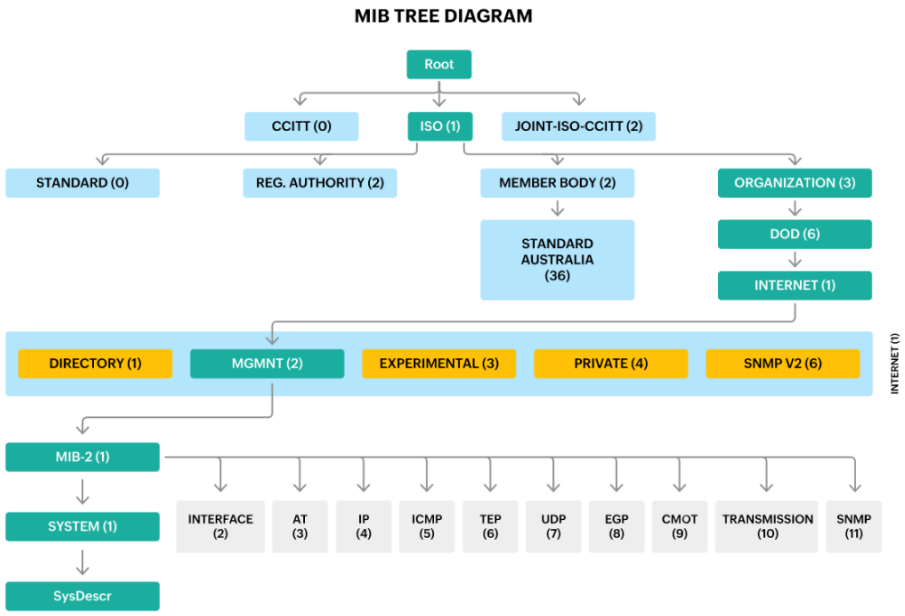


Рисунок 3.6 – Ієрархія дерева MIB

Таким чином в ієрархічному дереві MIB програмний модуль контролю стану буде використовувати , як стандартні типи керованих об’єктів так і власні.

3.3. Програмний модуль контролю стану мережевих пристроїв

Поєднавши Python та менеджер SNMP було створено програмний модуль, який запускається на одноплатному комп’ютері [1]. Мова програмування Python найкраще підходить для одноплатного комп’ютера, так як розробники Rasberry Pi розробили для неї велику кількість готових бібліотек керування різноманітними пристроями.

Цей програмний модуль сканує мережеві пристрої, визначає таблиці MAC адрес активних пристроїв у мережі, будує ієрархічне дерево MIB, взаємодіє із локальним сервісом SNMP, контролює стан, підключеного до нього мережевого обладнання, надсилає повідомлення на менеджер SNMP та викликає програму перемикання живлення керованого ним мережевого пристрою.

Перехоплення пасток-повідомлень SNMP відбувається, коли будь-яка подія, спровокована пристроєм, виявляється та надсилається приймачеві перехоплень [8]. Ці події включають зміни стану або виявлення аномалій (рисунок 3.7).

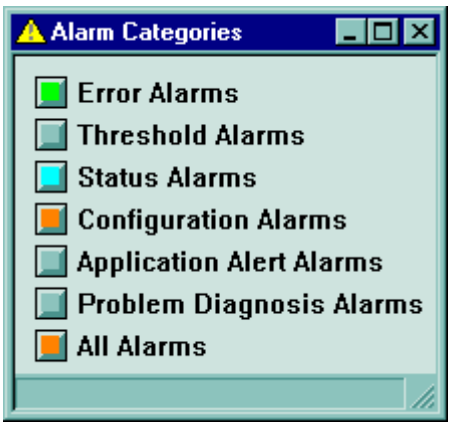


Рисунок 3.7 – Моніторинг повідомлень

Як було зазначено раніше наш програмний модуль виступає агентом менеджера SNMP HP OpenView (скор. HPOV, HP OV), яке є сімейством програмних продуктів компанії Hewlett Packard по управлінню системами та комп'ютерними мережами. Відкрита архітектура взаємодії менеджерів і агентів по протоколу SNMP дозволяє зробити даний програмний модуль Python частиною потужної системи керування мережевим обладнанням

Програмний модуль складається з наступних файлів (підмодулів):

– main.py – основний підмодуль (додаток 3);

– scan.py – сканування пристроїв мережі (додаток 4);

– mac.py – збір фізичних адрес (додаток 5);

– mib.py – формування інформаційної бази даних (додаток 6);

– snmp.py – виконання запитів SNMP (додаток 7);

– trap.py – відсилання повідомлень про стан;

– control.py – контроль мережевого пристрою (додаток 8);

– reset.py – перезавантаження мережевого пристрою (додаток 9).

Кожен з цих файлів є набором окремих процедур і функцій, які необхідні для моніторингу, контролю та керуванням мережевим пристроєм.

Так файл main.py (додаток 3) є основним підмодулем і відповідає за логіку і взаємодію між всіма рештами процедурами і функціями. Він координує роботу решти підмодулів і постійно запущений як основний процес операційної системи одноплатгого комп'ютера. Тут створюються об'єкти або класи, які аргументуються для визначення чим є кожен із них. Далі описуються методи, за якими виконуватимуться дії над об'єктами. Деякі змінні передаються у функції, для можливості використовувати їх у коді, не описуючи ці змінні заново.

Система керування мережею отримує ці повідомлення про події (рисунок 3.8), що означає, що пошук несправностей відбувається автоматично.

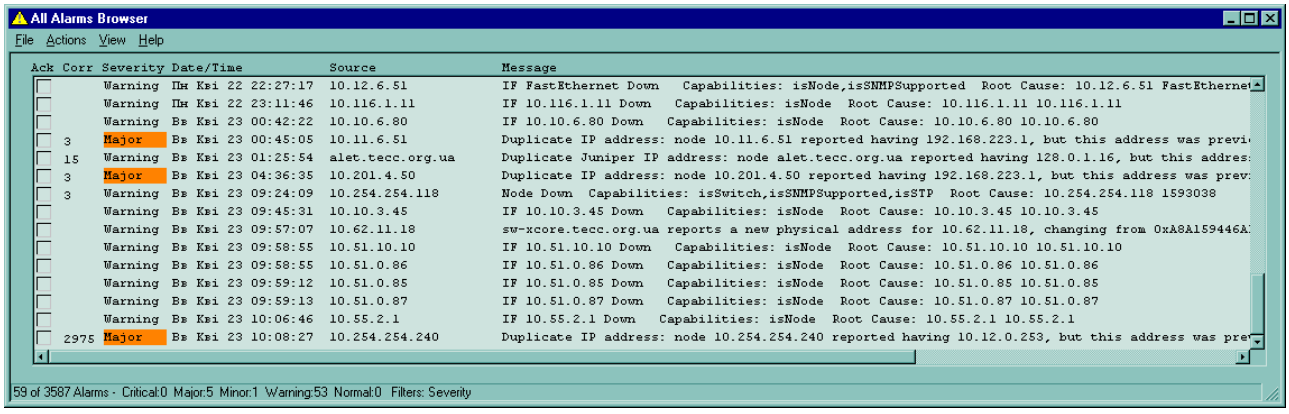


Рисунок 3.8 – Повідомлення про події

В процедурі scan.py знаходиться код, який відповідає за постійне сканування мережевих пристроїв, введеня IP адреси, створення переліку пристроїв, повідомлення про зміну стану пристроїв, прикріплення до сторінки скриптів з серверів. По суті у цьому файлі описане все, що безпосередньо бачить навколо себе програмний модуль.

Файл mac.py містить у собі рядки коду, які створюють з'єднання між програмним модулем та базою даних MAC адрес, у яку записують дані, що отримує менеджер SNMP після сканування мережі. В даній базі даних MAC адрес знаходяться фізичні адреси всіх оточуючих мережевих пристроїв, які бачить одноплатний контролер.

Підмодуль mib.py містить в собі функції, за допомогою яких, вираховується належність певних мережевих пристроїв до відповідних груп. У цьому файлі обробляються дані, що витягуються з бази даних MIB і представляються у вигляді ієрархічних здерев і в'язків. Результат також відправляється менеджеру SNMP, який виводить це на моніторингову програму мережевого оператора або адміністратора у зрозумілому для користувача вигляді.

У підмодулі snmp.py описані функції, які за допомогою SNMP команд отримують відповіді від мережевих пристроїв та допомагають будувати таблиці MAC адрес, номера VLAN по портам до яких вони належать. Якщо відповідь повертається і відображається у інтерфейсі без зміни інформації, тоді вважається, що комп'ютерна мережа знаходиться в стабільному режимі.

Функція trap.py є операторами, які відповідають за динамічне виконання запитів про текучий стан мережевого пристрою. Тобто, коли мережевий пристрій змінює стан, підмодуль відправляє повідомлення до менеджера SNMP.

Процедура control.py — це, можна сказати, основний виконуючий файл програмного модуля. В цьому файлі записані оператори, їх класи, об'єкти, методи та порядок виконання. Оператори описані таким чином, щоб

максимально визначати текучий стан мрежевого пристрою, яким керує контролер.

Якщо стан комп'ютерної мережі нормальний, то вся інформація передається менеджеру SNMP і програмний модуль виступає стандартним агентом SNMP. Якщо виникає проблема, адміністратор миттєво отримує сповіщення. Зазвичай ці повідомлення кодуються, і для їх розкодування використовується процесор перехоплень SNMP (рисунок 3.9).

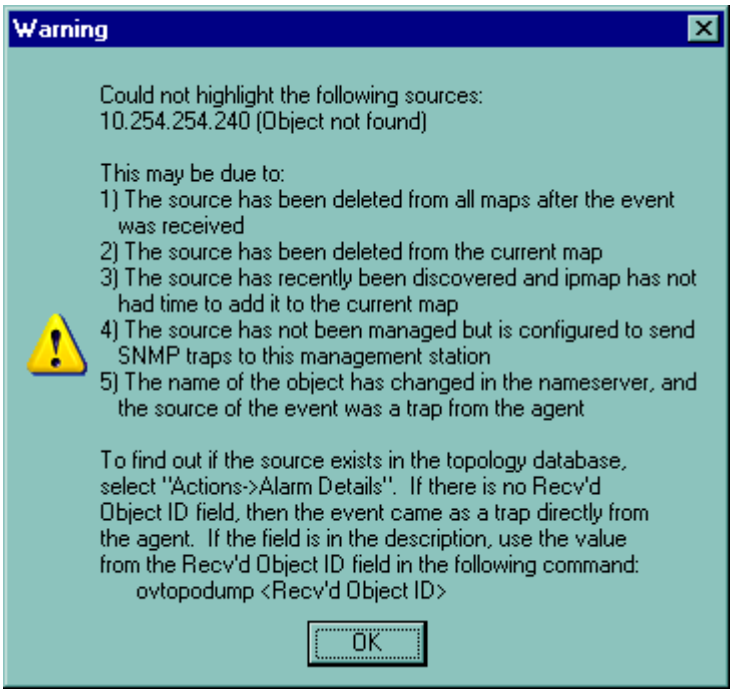


Рисунок 3.9 – Розкодування перехопленого сповіщення

У випадках, коли control.py бачить, що із мережевим пристроєм відбуваються негаразди, тоді підмодуль намагається передати цю інформацію менеджеру SNMP, який після втручання мережевого опреатора або адміністратора може перезавантажити пристрій. Якщо це неможливо, тоді запускається внутрішня процедура перевірки контролю стану мережевого обладнання.

Ця процедура включає перевірку чи відкликається мережевий пристрій на запити ICMP, перевіряє чи є фізичний адрес мережевого пристрою що моніториться у таблиці ARP. І коли не отримується ніяка

інформація про текучий стан і ніяких відповідей від мережевого пристрою, викликається функція reset.py.

Послідовність виконання файлів програми представлено схемою на рисунку 3.10.

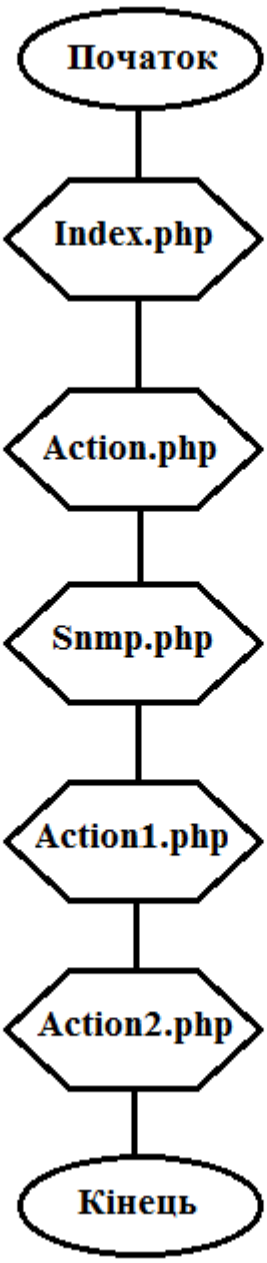


Рисунок 3.10 – Схема послідовності виконання файлів системи.

Крім того підмодуль control.py записує всю останню інформацію про мережевий пристрій, який котролює одноплатний контролер, у тимчасовий файл, який передається менеджеру SNMP після відновлення нормальної роботи комп’ютерної мережі.

Функція reset.py реалізує фізичне відключення від електроживлення на деякий час мережевого пристрою. Потім вмикає електроживлення і повідомляє про це підмодуль control.py.

ВИСНОВКИ

Дана робота показала можливість реалізації веб-інтерфейсу для керування комутаторами та маршрутизаторами по SNMP. Інтерфейс виявився зручним у використанні, так як не потребує інсталяції на комп'ютер. До розробленої програми є доступ з будь-якого хосту мережі. Функціонал продукту задовольняє усі потреби адміністрування. Швидкодія, веб варіанту програми, не перевищує загальний час виконання схожих запитів на прикладних аналогах. Крім того, в більшості існуючих продуктів, що працюють по SNMP, не реалізована можливість відобразити ієрархічну будову мережі.

Також було виявлено основні проблеми в реалізації продуктів, що працюють через SNMP (англ. Simple Network Management Protocol):

- персональні MIB;
- відносна застарілість протоколу;
- відсутність підтримки SNMP на більшості моделей активного мережевого обладнання

Для написання роботи використовувались:

- протокол SNMP;
- мови програмування PHP, JavaScript;
- підмережа з активним мережевим обладнанням, що підтримує SNMP;
- операційна система з налаштованим SNMP;
- веб-сервер Apache;
- база даних MySQL

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cisco – Україна. Cisco. URL: https://www.cisco.com/c/uk_ua/index.html (дата звернення: 18.01.2024).
2. Дипломне проектування. Методичні вказівки для студентів факультету автоматики. -Львів: Державний університет “Львівська політехніка”, 1996.-28с.
3. Douglas M., Schmidt K. Essential SNMP. 2-ге вид. OReilly, 2005. 460 с.
4. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. -К.: Держстандарт України, 1995. -37 с.
5. Gross, Kevin P., and Tom Holtzen. "Controlling and Monitoring Audio Systems with Simple Network Management Protocol (SNMP)." Audio Engineering Society Convention 105. Audio Engineering Society, 1998.
6. IEEE Xplore. IEEE Xplore. URL: http://www.ieeexplore.ieee.org (дата звернення: 08.02.2024).
7. K.O. What is IPMI?. MonoVM.com. URL: https://monovm.com/blog/what-is-ipmi/ (дата звернення: 05.06.2024).
8. Landing. D-Link. URL: https://www.dlink.com/ua/uk (дата звернення: 17.01.2024).
9. Law, David. "IEEE 802.3 Clause 30 Management, MIB, Registers and Function." IEEE P802. 3az, Energy-efficient Ethernet Task Force, Plenary Week Meeting. 2007.
10. LinuxQuestions.org. LinuxQuestions.org. URL: http://www.linuxquestions.org (дата звернення: 15.02.2024).
11. Методичні вказівки до дипломного проектування для студентів спеціальностей 7.050207 “Інформаційні системи в менеджменті”, 7.091504 “Захист інформації в комп’ютерних системах”. / Укл. І.А.Білоусов, Н.М.Васильків, Г.М.Гладій, М.П.Дивак.

12. Mauro, Douglas, and Kevin Schmidt. Essential SNMP: Help for System and Network Administrators. " O'Reilly Media, Inc.", 2005.

13. Murphy, Niall. Watchdog Timers-To keep a watchdog timer from resetting your system, you've got to kick it regularly. But that's not all there is to watchdog science. We will examine the use and testing of a. Embedded Systems Programming, 2000, 13.12: 112-126.

14. NETGEAR: Advanced WiFi & Networking. NETGEAR. URL: https://www.netgear.com/ (дата звернення: 23.11.2023).

15. Observability and IT management platform | solarwinds. Observability and IT Management Platform | SolarWinds. URL: http://www.solarwinds.com (дата звернення: 16.02.2024).

16. Olifer N., Olifer V. Computer networks: principles, technologies and protocols for network design. 3-тє вид. 2006.

17. Ortmeyer, Cliff. A Brief History of Single Board Computers. A Premier Farnell Company, Electronic Design Uncovered, USA, 2014, 06: 11.

18. PHP. PHP. URL: http://www.php.su/ (дата звернення: 07.11.2023).

19. RFC 1065. Structure and Identification of Management Information for TCP/IP-based internets.

20. RFC 1066. Management Information Base for Network Management of TCP/IP-based internets.

21. RFC 1067. A Simple Network Management Protocol.

22. RFC 1155. Structure and Identification of Management Information for TCP/IP-based internets.

23. RFC 1157. A Simple Network Management Protocol v.3.

24. RFC 1212. Concise MIB Definitions.

25. RFC 1213. Management Information Base for Network Management of TCP/IP-based internets: MIB-II.

26. Santic, John. Watchdog timer techniques. Embedded Systems Programming, 1995, 8.4: 58-69.

27. Schoenwaelder, J., and Jeffree, T. RFC 4789: Simple Network Management Protocol (SNMP) over IEEE 802 Networks. 2006.

28. Stephenson, Neal. In the beginning... was the command line. New York: Avon Books, 1999.

29. Supermicro IPMI utilities | supermicro server management utilities | supermicro. Supermicro Data Center Server, Blade, Data Storage, AI System. URL: https://www.supermicro.com/en/solutions/management-software/ipmi-utilities (дата звернення: 05.06.2024).

30. Tanenbaum A. S. Computer networks. New Jersey : Pearson Education International, 2003.

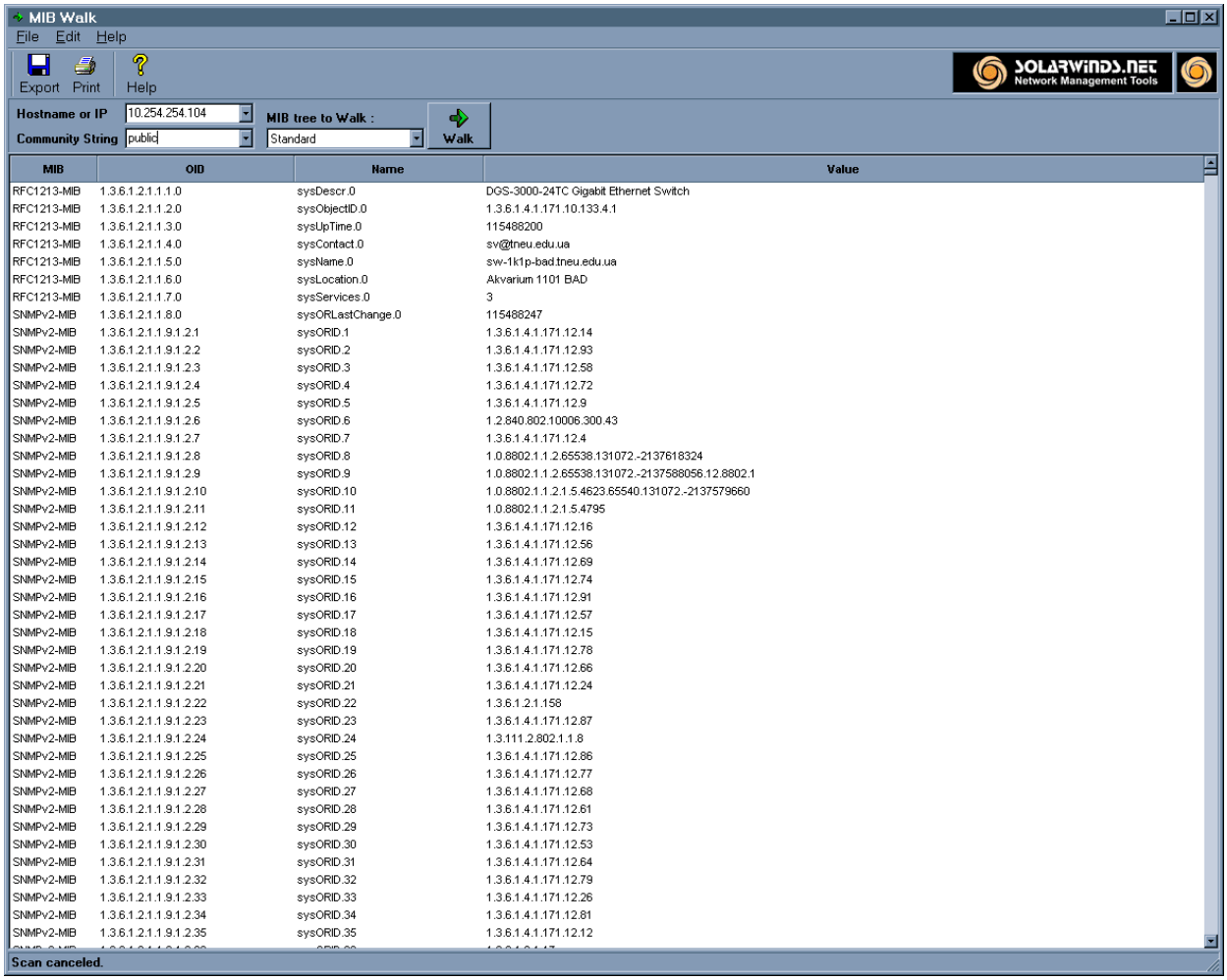
31. What is SNMP | SNMP Monitoring: Site24x7. Site24x7. URL: https://www.site24x7.com/network/what-is-snmp.html (дата звернення: 10.04.2024).

32. Служба підтримки Microsoft. Microsoft Support. URL: https://support.microsoft.com/uk-UA (дата звернення: 07.03.2024).

33. Zissis, Dimitrios; LEKKAS, Dimitrios. Addressing cloud computing security issues. Future Generation computer systems, 2012, 28.3: 583-592.

34. Zyxel. Zyxel. URL: https://www.zyxel.com/ua/uk-ua/home (дата звернення: 22.12.2023).

ДОДАТОК 2 – База даних MIB комутатора



MIB	OID	Name	Value
RFC1213-MIB	1.3.6.1.2.1.1.1.0	sysDescr.0	DGS-3000-24TC Gigabit Ethernet Switch
RFC1213-MIB	1.3.6.1.2.1.1.2.0	sysObjectID.0	1.3.6.1.4.1.171.10.133.4.1
RFC1213-MIB	1.3.6.1.2.1.1.3.0	sysUpTime.0	115488200
RFC1213-MIB	1.3.6.1.2.1.1.4.0	sysContact.0	sv@tneu.edu.ua
RFC1213-MIB	1.3.6.1.2.1.1.5.0	sysName.0	sw-1k1p-bad.tneu.edu.ua
RFC1213-MIB	1.3.6.1.2.1.1.6.0	sysLocation.0	Akvarium 1101 BAD
RFC1213-MIB	1.3.6.1.2.1.1.7.0	sysServices.0	3
SNMPv2-MIB	1.3.6.1.2.1.1.8.0	sysORLastChange.0	115488247
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.1	sysORID.1	1.3.6.1.4.1.171.12.14
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.2	sysORID.2	1.3.6.1.4.1.171.12.93
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.3	sysORID.3	1.3.6.1.4.1.171.12.58
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.4	sysORID.4	1.3.6.1.4.1.171.12.72
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.5	sysORID.5	1.3.6.1.4.1.171.12.9
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.6	sysORID.6	1.2.840.802.10006.300.43
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.7	sysORID.7	1.3.6.1.4.1.171.12.4
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.8	sysORID.8	1.0.8802.1.1.2.65538.131072.-2137618324
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.9	sysORID.9	1.0.8802.1.1.2.65538.131072.-2137588056.12.8802.1
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.10	sysORID.10	1.0.8802.1.1.2.1.5.4623.65540.131072.-2137579660
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.11	sysORID.11	1.0.8802.1.1.2.1.5.4795
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.12	sysORID.12	1.3.6.1.4.1.171.12.16
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.13	sysORID.13	1.3.6.1.4.1.171.12.56
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.14	sysORID.14	1.3.6.1.4.1.171.12.69
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.15	sysORID.15	1.3.6.1.4.1.171.12.74
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.16	sysORID.16	1.3.6.1.4.1.171.12.91
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.17	sysORID.17	1.3.6.1.4.1.171.12.57
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.18	sysORID.18	1.3.6.1.4.1.171.12.15
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.19	sysORID.19	1.3.6.1.4.1.171.12.78
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.20	sysORID.20	1.3.6.1.4.1.171.12.66
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.21	sysORID.21	1.3.6.1.4.1.171.12.24
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.22	sysORID.22	1.3.6.1.2.1.158
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.23	sysORID.23	1.3.6.1.4.1.171.12.87
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.24	sysORID.24	1.3.111.2.802.1.1.8
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.25	sysORID.25	1.3.6.1.4.1.171.12.86
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.26	sysORID.26	1.3.6.1.4.1.171.12.77
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.27	sysORID.27	1.3.6.1.4.1.171.12.68
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.28	sysORID.28	1.3.6.1.4.1.171.12.61
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.29	sysORID.29	1.3.6.1.4.1.171.12.73
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.30	sysORID.30	1.3.6.1.4.1.171.12.53
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.31	sysORID.31	1.3.6.1.4.1.171.12.64
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.32	sysORID.32	1.3.6.1.4.1.171.12.79
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.33	sysORID.33	1.3.6.1.4.1.171.12.26
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.34	sysORID.34	1.3.6.1.4.1.171.12.81
SNMPv2-MIB	1.3.6.1.2.1.1.9.1.2.35	sysORID.35	1.3.6.1.4.1.171.12.12

Рисунок 1 – Нумерація OID комутатора D-Link

ДОДАТОК 3 – Файл main.py

```
import time, sys
from time import sleep
from RPi import GPIO
from gpiozero import OutputDevice
from threading import Thread
from snmp import MIB

seg =
[MIB(27,active_high=False),MIB(25,active_high=False),MIB(24,
active_high=False),
MIB(23,active_high=False),MIB(22,active_high=False),MIB(18,a
ctive_high=False),
MIB(17,active_high=False)]
segmentPattern =
[[0,1,2,3,4,5],[1,2],[0,1,6,4,3],[0,1,2,3,6],[1,2,5,6],[0,2,
3,5,6], #0 to 5
[0,2,3,4,5,6],[0,1,2],[0,1,2,3,4,5,6],[0,1,2,5,6],[0,1,2,4,5
,6], #6 to A
[2,3,4,5,6],[0,3,4,5],[1,2,3,4,6],[0,3,4,5,6],[0,4,5,6] ] #B
to F
sensor = NetDevice(15,4)
def main() :
print("Display state of network device")
while 1:
distance = sensor.distance * 10 # distance in ms
print("distance",distance)
if distance >= 10.0:
```

```
distance = 16.0
display(int(distance))
time.sleep(0.8)
def convert_temp(gen):
for value in gen:
yield (value * 3.3 - 0.5) * 100
adc = MCP3008(channel=7)
graph = MIBGraph (26, 19, 13, 6, 5, pwm=True)
for temp in convert_temp(adc.values):
bars = temp / 35
graph.value = bars
sleep(1)
def display(number):
for i in range(0,7):
seg[i].off()
if number < 16:
for i in range(0,len(segmentPattern[number])):
seg[segmentPattern[number][i]].on()
# Main program logic:
if __name__ == '__main__':
main()
```

ДОДАТОК 4 – Файл scan.py

```
import scapy.all as scapy
def scan(ip):
    arp_request = scapy.ARP(pdst=ip)
    broadcast = scapy.Ether(dst="ff:ff:ff:ff:ff:ff")
    arp_request_broadcast = broadcast / arp_request
    answered_list = scapy.srp(arp_request_broadcast,
timeout=1, verbose=False)[0]
    results = []
    for element in answered_list:
     result = {"ip": element[1].psrc, "mac":
element[1].hwsrc}
     results.append(result)
     return results
def display_results(results):
    print("IP Address\t\tMAC Address")
    print("-----------------------------------------")
    for result in results:
        print(result["ip"] + "\t\t" + result["mac"])
target_ip = "192.168.1.1/24"
scan_results = scan(target_ip)
display_results(scan_results)
```

ДОДАТОК 5 – Файл mac.py

```
import sys
from requests import get
__version__ = "1.0"
class Macpyoui:
    def __init__(self, api):
        self.api = api
site = "https://api.macvendors.com/"
data = Macpyoui(site)
macaddress = input("Please enter the MAC address: ")
def searchmac():
    macsend = data.api + macaddress
    vendorsearch = get(macsend).text
    if "Not Found" in vendorsearch:
        print("MAC address not found.")
    elif len(sys.argv) == 1:
        print("No MAC address entered.")
    else:
        print(vendorsearch)
if __name__ == "__main__":
    searchmac()
```

ДОДАТОК 6 – Файл mib.py

```
import sys
from requests import get


 {
      "ifMIB": {
          "name": "ifMIB",
          "oid": "1.3.6.1.2.1.31",
          "class": "moduleidentity",
          "revisions": [
            "2007-02-15 00:00",
            "1996-02-28 21:55",
            "1993-11-08 21:55"
          ]
        },
      ...
      "ifTestTable": {
        "name": "ifTestTable",
        "oid": "1.3.6.1.2.1.31.1.3",
        "nodetype": "table",
        "class": "objecttype",
        "maxaccess": "not-accessible"
      },
      "ifTestEntry": {
        "name": "ifTestEntry",
        "oid": "1.3.6.1.2.1.31.1.3.1",
        "nodetype": "row",
        "class": "objecttype",
```

```
      "maxaccess": "not-accessible",
      "augmention": {
        "name": "ifTestEntry",
        "module": "IF-MIB",
        "object": "ifEntry"
      }
    },
    "ifTestId": {
      "name": "ifTestId",
      "oid": "1.3.6.1.2.1.31.1.3.1.1",
      "nodetype": "column",
      "class": "objecttype",
      "syntax": {
        "type": "TestAndIncr",
        "class": "type"
      },
      "maxaccess": "read-write"
    },

  }
```

ДОДАТОК 7 – Файл snmp.py

```
import re
import json
from sys import argv
from pysnmp.entity.rfc3413.oneliner import cmdgen
import logging
logging.basicConfig(
    level=logging.DEBUG, filename="./log.txt",
    format='%(asctime)s %(name)s.%(funcName)s +%(lineno)s:
%(levelname)-8s [%(process)d] %(message)s',
)
logger = logging.getLogger("./log.txt")
class Device:
    def __init__(self, ipswitch, ro_community, oid_mt,
port=161):
        self.ip = ipswitch
        self.ro = ro_community
        self.oid = oid_mt
        self.port = port
        self.if_oids = ['ifAdminStatus', 'ifOperStatus',
'ifInOctets', 'ifOutOctets']
        self.types_response = {'7': 'ifAdminStatus',
                               '8': 'ifOperStatus',
                               '10': 'ifInOctets',
                               '16': 'ifOutOctets'
                               }
```

```
        self.re_part =
re.compile("(\d\.\d\.\d\.\d\.\d\.\d\.)(?P<part_mt>.*?)$",
re.MULTILINE | re.DOTALL)
        self.part_mt_oid =
self.re_part.search(self.oid).group('part_mt')
        self.re_mt =
re.compile(f'\S+({self.part_mt_oid})\.(?P<port>\d{1,
2})\.(?P<sign>\d+)',
                                   re.MULTILINE | re.DOTALL)
        self.re_if =
re.compile("\S+\:\:\S+2\.2\.1\.(?P<key>\d+)\.(?P<port>\d{1,2
})$",
                                   re.MULTILINE | re.DOTALL)
        self.result = {}

    def get_ifwalk(self) -> dict:
        """
        Получение ответов коммутатора на ifAdminStatus,
ifOperStatus, ifInOctets, ifOutOctets и переданный
медиатайп.
        :return: self.result: dict
        """

        oids_form = [(oid_if,) for oid_if in self.if_oids]
        oids_form.extend((self.oid,))

        try:
            cmdGen = cmdgen.CommandGenerator()
```

```
            errorIndication, errorStatus, errorIndex,
varBindTable = cmdGen.nextCmd(
                cmdgen.CommunityData(self.ro, mpModel=1),
                cmdgen.UdpTransportTarget((self.ip,
self.port)),
                *oids_form)

            if errorIndication:
                raise BaseException(f"errorIndication:
{errorIndication}")
            if errorStatus:
                raise BaseException(f"errorStatus: "

f"{errorStatus.prettyPrint(), errorIndex and varBindTable[-
1][int(errorIndex) - 1] or '?'}")

            # если нет ошибок в полученном ответе -
записываем все параметры в словарь
            for varBindTableRow in varBindTable:
                for name, val in varBindTableRow:

                    founds_mt_responce =
self.re_mt.search(name.prettyPrint())
                    if founds_mt_responce is not None:
                        port =
founds_mt_responce.group("port")
```

```
                        self.result.setdefault('sign',
{})[port] = founds_mt_responce.group("sign")
                        self.result.setdefault('link',
{})[port] = val.prettyPrint()

                    found_if_responce =
self.re_if.search(name.prettyPrint())
                    if found_if_responce is not None:
                        port =
found_if_responce.group('port')
                        type_response =
self.types_response.get(found_if_responce.group('key'))
                        if (type_response in
['ifAdminStatus', 'ifOperStatus']) and (val.prettyPrint() ==
'1'):
                            status = 'up' if
val.prettyPrint() == '1' else 'down'

self.result.setdefault(type_response, {})[port] = status
                            continue

self.result.setdefault(type_response, {})[port] =
val.prettyPrint()

        except BaseException as bex:
            logger.error(bex)
        return self.result
```

```
if __name__ == "__main__":
    name_script, ip, ro, oid = argv
    device = Device(ip, ro, oid)
    print(json.dumps(device.get_ifwalk()))
```

ДОДАТОК 8 – Файл control.py

```
import time, sys
from gpiozero import NetDevice, OutputDevice
from threading import Thread
sensor = NetDevice(echo = 16, trigger = 20)
IN1_m1 = OutputDevice(17)
IN2_m1 = OutputDevice(18)
IN3_m1 = OutputDevice(21)
IN4_m1 = OutputDevice(22)
IcmpPing_m1 = [IN1_m1,IN2_m1,IN3_m1,IN4_m1] # SNMP v1 ping
IN4_m2 = OutputDevice(19)
IN3_m2 = OutputDevice(13)
IN2_m2 = OutputDevice(5)
IN1_m2 = OutputDevice(6)
IcmpPing_m2 = [IN1_m2,IN2_m2,IN3_m2,IN4_m2] # SNMP v2 ping
Seq = [[1,0,0,1], # Defi ne icmp sequence
[1,0,0,0], # as shown in manufacturer’s datasheet
[1,1,0,0],
[0,1,0,0],
[0,1,1,0],
[0,0,1,0],
[0,0,1,1],
[0,0,0,1]]

Ping = len(Seq)
all_clear = True
running = True
def bump_watch(): # thread to watch for obstacles
global all_clear
```

```
while running:
value = sensor.distance
if value < 0.1: # trigger if obstacle
all_clear = False
else:
all_clear = True
def move_bump(snmp_request='F', seqsize=1, numicmps=2052):
counter = 0 # 2052 icmps = 1 revolution for icmp size of 2
IcmpDir = seqsize # Set to 1 or 2 for fwd, -1 or -2 for back
if snmp_request == 'T':
IcmpDir = IcmpDir * -1
WaitTime = 10/fl oat(1000) # adjust this to change speed
Pinger = 0
while all_clear and counter < numicmps: # if no obstacles
for pin in range(0, 4):
Lpin = IcmpPing_m1[pin]
Rpin = IcmpPing_m2[pin]
if Seq[Pinger][pin]!=0: # F=fwd, B=back, L=left, R=right
if snmp_request == 'L' or snmp_request == 'B' or
snmp_request == 'F':
Lpin.on() # Left device
if snmp_request == 'R' or snmp_request == 'B' or
snmp_request == 'F':
Rpin.on() # Right device
else:
Lpin.off()
Rpin.off()
Pinger += IcmpDir
```

```
if (Pinger>=Ping): # Repeat sequence
Pinger = 0
if (Pinger<0):
Pinger = Ping+IcmpDir
time.sleep(WaitTime) # pause
counter+=1

t1 = Thread(target=bump_watch) # run as separate thread
t1.start() # start bump watch thread
for i in range(4): # right-handed device
move_bump('F',-2,4104)
move_bump('R',-2,2052)
running = False
```

ДОДАТОК 9 – Файл reset.py

```
from RPi import GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(4, GPIO.IN, GPIO.PUD_UP)
while GPIO.input(4):
pass
print("Device powered off !")
if Rele == 'Off' or Rele == 'false' or Rele == '0':
Rpin.on() # Rele power off
else:
Lpin.off()
Rpin.off()
Pinger += IcmpDir
Return MAIN()
```