

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і проєкційної інженерії
(назва факультету)
Кафедра комп'ютерних систем та мереж
(повне найменування кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проєкту (роботи)

на тему: Засоби моніторингу та управління елементами IoT
ша 61 стро
(освітній ступінь (освітньо-кваліфікаційний рівень))

Виконав: студент (ка) 6 курсу, групи СІІІ-61
спеціальності (напряму підготовки)

123 "Комп'ютерна інженерія"
(шифр і назва спеціальності (напряму підготовки))

	<u>[Signature]</u> (підпис)	<u>Цюрик І. І.</u> (прізвище та ініціали)
Керівник	<u>[Signature]</u> (підпис)	<u>Ліній О. П.</u> (прізвище та ініціали)
Нормоконтроль	<u>[Signature]</u> (підпис)	<u>Жуков С. В.</u> (прізвище та ініціали)
Рецензент	<u>[Signature]</u> (підпис)	<u>Крамар О. О.</u> (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)

Факультет Інформаційні системи та програмно-інженерія
Кафедра Комп'ютерні системи та мережі
Освітній ступінь магістр
Напрямок підготовки _____
Спеціальність 123 «Комп'ютерна інженерія»
(шифр і назва)
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри КС
Осудівська Г.М.
«30» 04 2019 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ**

Цюпріка Богдан Зіорович
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Засоби моніторингу та управління елементом IoT

Керівник проекту (роботи) Денис Олег Петрович доктор технічних наук, професор
(прізвище, ім'я, по батькові, науковий ступінь, місце роботи)

Затверджені наказом по університету від «24» 08 2019 року № 4/7-854

2. Термін подання студентом проекту (роботи) 26.12.2019

3. Вихідні дані до проекту (роботи) Засоби моніторингу та управління елементом IoT, пріоритети передачі даних по короткій відстані, засоби передачі даних по короткій відстані

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Вступ, 2. Огляд архітектури мережі інтернету речей, 3. Опис і аналіз параметрів мережі та налаштування засобів мережі та управління елементом IoT, 4. Реалізація програмного забезпечення для засобів моніторингу та управління елементом IoT 5. Обґрунтування економічної частини ефективності в короткій мережі та безпечності в надзвичайних ситуаціях, 7. Експлуатація, 8. Висновки.

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

1. Засоби моніторингу та управління елементом IoT, 2. Мережа об'єктів, 3. Порівняння результатів, 4. Порівняння ефективності передачі даних по мережі, 5. Порівняння ефективності передачі даних по мережі, 6. Налаштування мережі BWP/DO, 7. Конфігурація мережі, 8. Порівняння роботи мережі, 9. Приклад записів, 10. Висновки

АНОТАЦІЯ

Засоби моніторингу та управління елементами IoT // Дипломна робота // Цюприк Богдан Ігорович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, група СІм-61 // Тернопіль, 2019 // с. – 120, рис. – 51, табл. – 8, аркушів А1 – 10, додат. – 2, бібліогра. – 38.

Ключові слова: ІНТЕРНЕТ РЕЧЕЙ, ДАВАЧІ, СЕНСОРИ, INTERNET OF THINGS, RASPBERRY PI, DOMOTICZ, МОНІТОРИНГ ЕЛЕМЕНТІВ, JAVA.

Мета роботи полягає у дослідженні та розробці системи керування засобами моніторингу та управління елементами IoT.

У дипломній роботі досліджено та налаштовано апаратні засоби та розроблено програмне забезпечення, яке використовується для моніторингу та керування елементами інтернету речей.

Проаналізовано протоколи та методи, що використовуються для передавання інформації в мережі інтернету речей, що дало змогу виявити позитивні та негативні якості кожного з протоколів, та обрати оптимальні для створення мережі.

На основі системи Domoticz, з допомогою мови програмування Java та фреймворку Spring Boot, розроблено систему, що дозволяє з декількох віддалених один від одного місць віддалено спостерігати та керувати елементами інтернету речей.

Продемонстровано приклад роботи системи.

ANNOTATION

IoT components monitoring and control facilities // Master diploma thesis // Tsiupryk Bohdan // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information System and Software Engineering, group CIM-61 // Ternopil, 2019 // p. – 120, fig. – 51, tab. – 8, sheets A1. – 10, addit. – 1, bibliography – 38.

Key words: INTERNET OF THINGS, SENSORS RASPBERRY PI, DOMOTICZ, ELEMENT MONITORING, JAVA.

The purpose of the work is to research and develop an IoT monitoring and control system.

The thesis investigates and configures the hardware and develops software that is used to monitor and control the elements of the Internet of Things.

The protocols and methods used to transmit information on the Internet of Things were analyzed, which revealed the positive and negative qualities of each protocol and selected the best for networking.

Using the Java programming language and Spring Boot framework, Domoticz has developed a system that allows you to remotely monitor and control elements of the Internet of Things from several remote locations.

An example of how the system works is demonstrated.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 ОГЛЯД АРХІТЕКТУРИ МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ	12
1.1 Екосистема інтернету речей	13
1.2 Огляд архітектури інтернету речей	15
1.3 Огляд компонентів архітектури інтернету речей	18
1.4 Огляд технологій та протоколів передачі даних у мережі інтернету речей	23
1.5 Протоколи для передачі повідомлень в мережі інтернету речей	31
1.6 Огляд існуючих засобів моніторингу та управління елементами інтернету речей	38
1.7 Висновок до розділу	41
РОЗДІЛ 2 ОПИС І АНАЛІЗ АПАРАТНОЇ ЧАСТИНИ ТА НАЛАШТУВАННЯ ЗАСОБІВ МОНІТОРИНГУ ТА УПРАВЛІННЯ ЕЛЕМЕНТАМИ ІОТ	42
2.1 Опис вибраної платформи Raspberry Pi	42
2.2 Установка та налаштування Raspberry Pi	45
2.3 Підключення зовнішнього диска та модуля Wi-Fi	47
2.4 Підключення до Raspberry Pi з допомогою ssh. Налаштування з'єднання та встановлення Domoticz	50
2.5 Аналіз Domoticz API, як способу управління	57
2.6 Висновок до розділу	58
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАСОБІВ МОНІТОРИНГУ ТА УПРАВЛІННЯ ЕЛЕМЕНТАМИ ІОТ	59
3.1 Опис архітектури та інструментів розробки	59

	7
3.2 Реалізація системи автоматизації збору даних	63
3.3 Налаштування системи	68
3.4 Демонстрація роботи	70
3.5 Висновки до розділу	74

**РОЗДІЛ 4 ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ Помилка!
Закладку не визначено.**

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР	75
4.2 Визначення витрат на оплату праці та відрахуван на соціальні заходи	81
4.3 Розрахунок витрат на електроенергію	81
4.4 Розрахунок витрат на матеріали	81
4.5 Розрахунок суми амортизаційних відрахувань	82
4.6 Обчислення накладних витрат	83
4.7 Складання кошторису витрат та визначення собівартості НДР	83
4.8 Розрахунок ціни НДР	84
4.9 Визначення економічної ефективності і терміну окупності капітальних вкладень	85
4.10 Висновки до розділу	89

РОЗДІЛ 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКИ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1 Охорона праці	87
5.2 Основи фізіологічної праці й комфортних умов життєдіяльності. Класифікація основних форм діяльності людини. Особливості фізичної та розумової праці.	90
5.3 Оцінка можливої радіаційної обстановки при аварії на АЕС	93

	8
5.4 Висновки до розділу	96
РОЗДІЛ 6 ЕКОЛОГІЯ	97
6.1 Відчуження земель під лінії електропередач	97
6.2 Етапи та техніка збору та обробки екологічної інформації	99
6.3 Висновки до розділу	102
ВИСНОВКИ	104
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	105
Додаток А Тези конференцій	115
Додаток Б Схема шини GPIO плати Raspberry Pi	121

ВСТУП

Актуальність теми. Інтернет речей (Internet of Things) – по своїй сутті є мережею мереж, що складається з об'єктів (речей), які мають індивідуальні ідентифікатори та можуть взаємодіяти між собою без втручання ззовні.

На сьогоднішній день до пристроїв IoT відносяться мільярди пристроїв по всьому світу, які підключенні до мережі інтернет. Слід зазначити, що термін інтернет речей, давно перейшов в складніше явище ніж просто набір давачі, які виводили інформацію на екран приймального пристрою. На даний час, давачі об'єднуються в єдину мережу де відбувається обмін даними, обробка, аналітика та керування системою, таким чином формується самостійна система, що потребує мінімального втручання людини, або і взагалі не потребує останнього. Система здатна сама приймати рішення та керувати об'єктом, або системою об'єктів, в залежності від того для чого була розроблена мережа.

Мета і завдання дослідження. Дослідження та розробка системи керування засобами моніторингу та управління елементами IoT.

Для досягнення поставленої цілі, необхідно розв'язати наступні задачі:

- дослідити сучасні системи управління елементами інтернету речей;
- проаналізувати архітектуру мережі інтернету речей, технології та протоколи передачі даних;
- дослідити апаратне забезпечення та провести його налаштування;
- реалізувати програмне забезпечення з метою покращення управління елементами інтернету речей;
- здійснити оцінку якості розробленого програмного забезпечення.

Об'єкт дослідження - система моніторингу та керування елементами інтернету речей.

Предметом дослідження є засоби та протоколи передачі даних в мережі інтернету речей та системи моніторингу та управління її елементами;

Методи дослідження. Для виконання задачі дипломної роботи використано наступні методи: теоретико-емпіричний, системного аналізу, теорії проектування комп'ютерних мереж.

Наукова новизна одержаних результатів:

- проведено компаративний аналіз сучасних протоколів передачі даних та засобів моніторингу та керування елементами інтернету речей, що дало змогу обґрунтувати на основі кількісних та якісних показників найбільш ефективні з них та дозволило виявити недоліки та переваги, що потрібно вирішити при розробці своєї системи;

- серед множини апаратних засобів для розгортання системи моніторингу та управління елементами інтернету речей обґрунтовано переваги використання одноплатних комп'ютерів, як найбільш ефективних для систем моніторингу та керування елементами інтернету речей;

- з використанням інженерії прикладного програмного забезпечення, а саме із застосуванням структурного, компонентного та об'єктно-орієнтованого методів, розроблено систему для моніторингу та керування елементами інтернету речей.

Практичне значення отриманих результатів. Налаштовано апаратні засоби та розроблено програмне забезпечення, яке використовується для моніторингу та керування елементами інтернету речей.

Публікації. Результати роботи апробовано на VIII Міжнародній науково-технічній конференції молодих учених та студентів «Актуальні задачі сучасних технологій» м. Тернопіль 27-28 листопада 2019 року та VII

науковотехнічна конференція «Інформаційні моделі, системи та технології» м. Тернопіль 11-12 грудня 2019 року.

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається із вступу, шести розділів, висновків, списку використаних джерел та додатків. Обсяг роботи: пояснювальна записка – 120 аркушів формату А4, графічна частина – 10 аркушів формату А1.

РОЗДІЛ 1

ОГЛЯД АРХІТЕКТУРИ МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ

Термін «інтернет речей» завдячує своєю появою Кевіну Ештону, який у 1997 р, застосував технологію радіочастотної ідентифікації (RFID) для керування системою поставок. Завдяки цій роботі його запросили в Масачусетський технологічний інститут, де він з групою колег організував дослідницький консорціум Auto-ID Center. З того часу інтернет речей перейшов від простих радіо-міток до екосистеми.

Першим пристроєм інтернету речей можна вважати автомат з газованою водою, підключений до інтернету в університеті Карнегі-Меллон у 1982 році. Поступово до IoT почали підключати все більше речей. У 1991 році компанія HP представила перший мережевий принтер, згодом у 1993 році до інтернету підключено першу камера у Кембриджському університеті. Поява організації Bluetooth SIG у 1998 році сильно вплинула на розвиток IoT, оскільки технологія Bluetooth споживала набагато менше енергії, що допомогло досягти ліпшої автономності пристроїв. 2000 рік – перші прояви розробленої компанією HP концепції всепроникної комп'ютеризації (Cooltown): HP Labs, система обчислювальних і комунікаційних технологій, які у поєднанні один з одним створюють підключення до Інтернету для людей, місць і об'єктів. У 2005 сформовано звіт міжнародним союзом електрозв'язку (спеціалізована установа ООН), у якому вперше сформульовано прогнози розвитку інтернету речей. Успішна розробка напівпровідникових світлодіодних ламп у 2008 призвела до розвитку концепції «розумного» освітлення, та у свою чергу, «розумного» дому.

Прогнозується, що в найближчі 5-7 років, Інтернет речей захопить кожен сегмент в сфері промисловості, бізнесу, охорони здоров'я і споживчих товарів.

1.1 Екосистема інтернету речей

До екосистеми інтернету речей відносять засоби, сервіси і технології, які застосовують в інтернеті речей. Їх можна поділити на такі категорії:

- Сенсори, розумні датчики, давачі: вбудовані системи, операційні системи реального часу, джерела безперебійного живлення, мікро-електромеханічні системи;

- Системи зв'язку з датчиками: система охоплення бездротових персональних мереж становить від 0 см до 100 метрів. Для обміну даними між датчиками застосовують низькошвидкісні малопотужні інформаційні канали, які часто побудовано не на протоколі IP;

- Локальні обчислювальні мережі (LAN): зазвичай, це системи обміну даними на основі протоколу IP, наприклад, 802.11 Wi-Fi-мережу для швидкої радіозв'язку, часто це пірінгові або зіркоподібні мережі;

- Агрегатори, маршрутизатори (routers), шлюзи (gateways), пограничні пристрої (Edge Device): постачальники вбудованих систем, самі бюджетні складові (процесори, динамічна оперативна пам'ять і система зберігання даних), виробники модулів, виробники пасивних компонентів, виробники тонких клієнтів, виробники стільникових і бездротових радіосистем, постачальники міжплатформового програмного забезпечення, розробники інфраструктури туманних обчислень, інструментарій для граничної аналітики, безпеку граничних пристроїв, системи управління сертифікатами;

- Глобальна обчислювальна мережа: оператори стільникового зв'язку, оператори супутникового зв'язку, оператори малопотужних глобальних мереж (Low- Power Wide-Area Network, LPWAN).

Зазвичай застосовуються транспортні протоколи Інтернету для IoT і мережевих пристроїв (MQTT, CoAP і навіть HTTP);

- Хмара: інфраструктура в якості постачальника послуг, платформа в якості постачальника послуг, розробники баз даних, постачальники послуг потокової і пакетної обробки даних, інструменти для аналізу даних, програмне забезпечення в якості постачальника послуг, постачальники озер даних, оператори програмно-визначених мереж / програмно-визначених параметрів, сервіси машинного навчання;

- Сервіси аналізу даних: величезні масиви інформації передаються в хмару. Робота з великими обсягами даних і отримання з них користі - це завдання, що вимагає комплексної обробки подій, аналітики і прийомів машинного навчання;

- Безпека (security): при зведенні всіх елементів архітектури воедино постають питання кібербезпеки. Безпека стосується кожного компонента: від датчиків фізичних величин до ЦПУ і цифрового апаратного забезпечення, систем радіозв'язку і самих протоколів передачі даних. На кожному рівні необхідно забезпечити безпеку, достовірність і цілісність. У цьому ланцюзі не повинно бути слабких ланок, оскільки Інтернет речей стане головною мішенню для атак хакерів в світі.

Загалом, елемент інтернету речей можна поділити на чотири категорії[1-2]. Це сервіси (Services), апаратна частина (Hardware), набір правил (Rules) та програмна частина (Software). Ці категорії також можна розділити на підгрупи за призначенням (див. рис. 1.1).

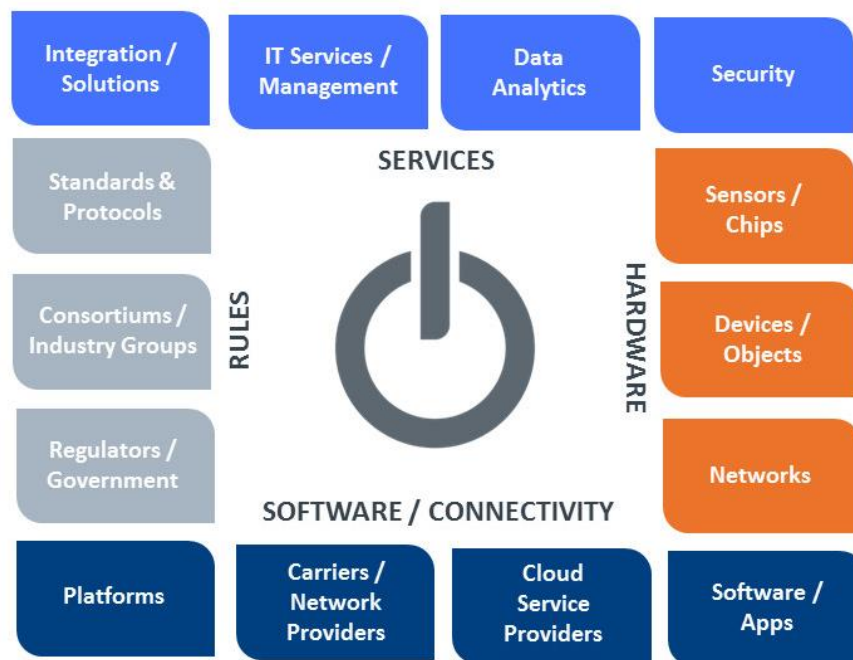


Рис. 1.1. зв'язок між елементами інтернету речей

1.2 Огляд архітектури інтернету речей

Архітектура IoT може відрізнитись в залежності від призначення та реалізації. Та загалом вона схожа на типову АСУТП (Автоматизована Система Управління Технологічними Процесами) (рис 1.2.).

Взаємодія з середовищем відбувається через датчики (sensors) та виконавчі механізми (actuators), аналогічно, як це робиться в АСУТП для будь-якого об'єкту керування. Ці датчики разом з усією інфраструктурою для інтеграції з рівнем обробки подій через мережу Internet формують так звану граничну область (Edge).

Події (дані), котрі надходять із граничної області, зберігають і опрацьовують відповідно до задачі (рівень обробки подій і аналітики, event processing, Platform).

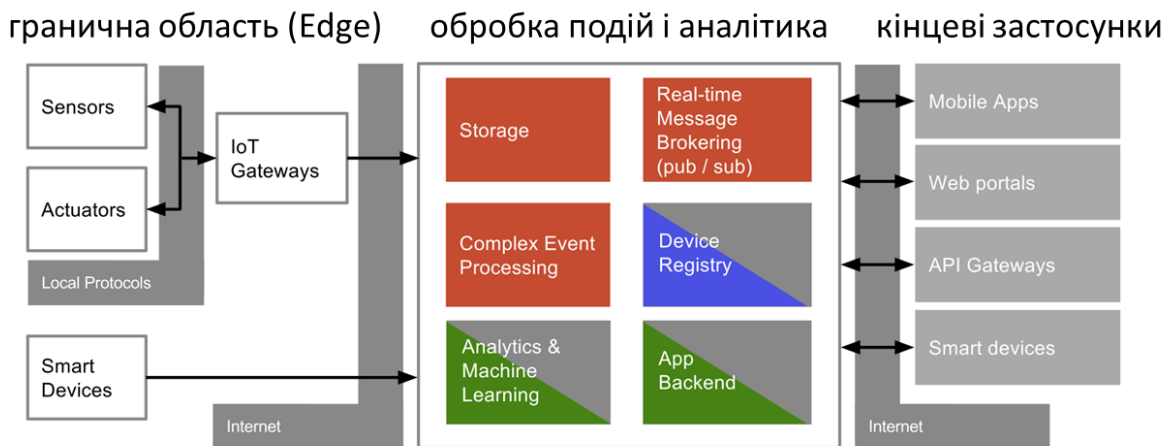


Рис. 1.2. Приклад архітектури IoT

Дані що поступають з граничної області зберігаються і обробляються відповідно до задачі. На цьому рівні дані зберігають (storage), опрацьовують (Event Processing), надсилають потрібним додаткам (Real-Time Message Brokering, Stream Processing). Додатково на цьому рівні здійснюють адміністрування та керування пристроями з граничної області (Device Registry, Edge Device Management). На основі подій (даних) проводять машинне навчання (Machine Learning), що дозволяє зробити певні висновки про об'єкт з використанням аналітичних сервісів (Analytics). Цей рівень, як правило реалізовано, з використанням хмарних (Cloud) або туманних (Fog) обчислень. Якщо провести аналогію с АСУТП, то це рівень контролерів та SCADA (за виключенням функцій НМІ). Отримання результатів, контроль, віддалене керування та адміністрування системи здійснюють через кінцеві застосунки з використанням мережі Internet.

На рис. 1.3 показано архітектуру, схожу до наведеної вище, однак у вигляді сервісів. Тут у вигляді датчиків представлено область Edge, також до неї віднесено Device Hub/Gateway (збір та маршрутизація даних) та Device Management (керування пристроями), котрі частково виконуються як хмарні обчислення так і на граничних пристроях. Усі функції збереження

та первинного опрацювання подій (даних) зведено до Data Management. Усі інші функції обробки, в тому числі аналітичні, показано як додатки PaaS, котрі взаємодіють з сервісами керування даних через API (Application Program Interface).

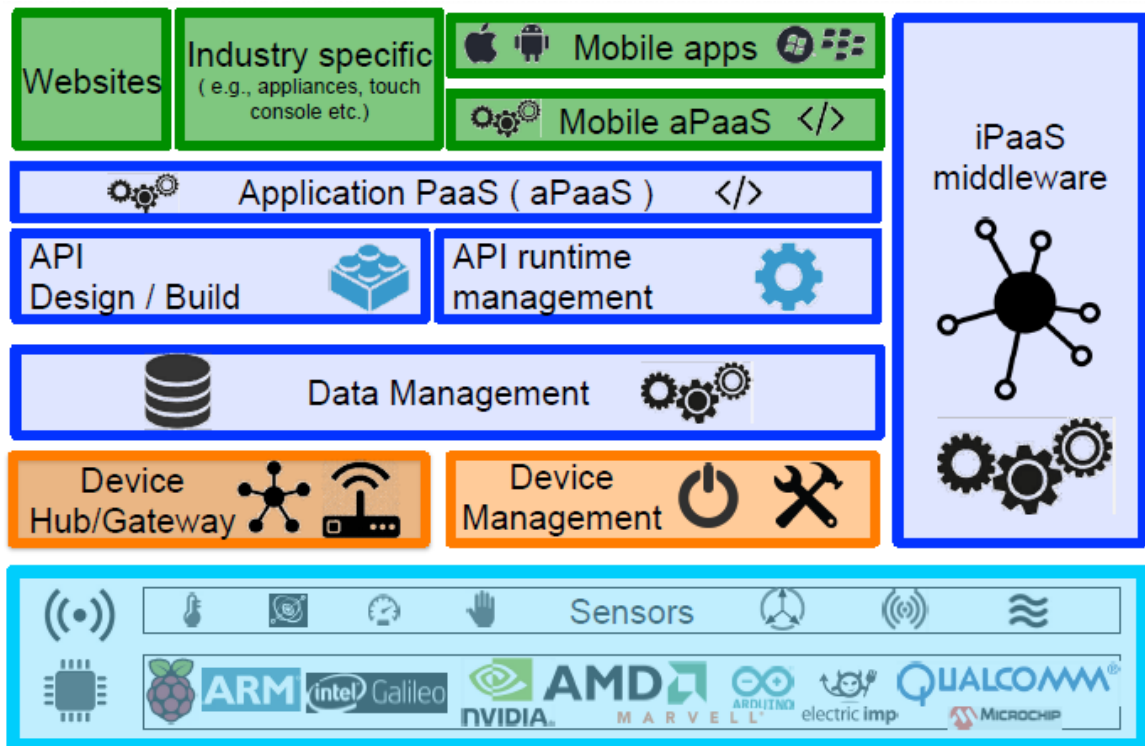


Рис. 1.3. Архітектура інтернету речей представлена у вигляді сервісів

Ще один приклад архітектури інтернету речей, вже на рівні модулів, показано на рис. 1.4. Як видно з цього рисунка, усі наведені архітектури мають спільні риси: наявність трьох рівнів, подібні функції, наявність хмарних обчислень, використання Інтернету як інтеграційного рівня.

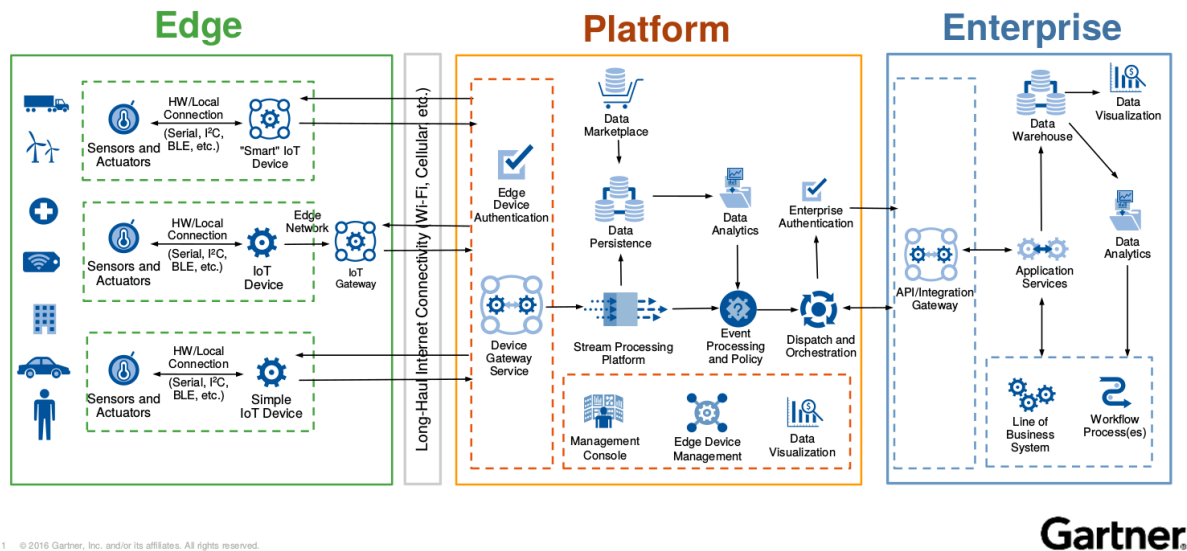


Рис. 1.4. Архітектура інтернету речей на рівні модулів

1.3 Огляд компонентів архітектури інтернету речей

1.3.1. Датчики та живлення. Інтернет речей починається або закінчується однією подією: простий рух, зміна температури або, можливо, важіль замикає замок. На відміну від багатьох існуючих ІТ-пристроїв, Інтернет речей здебільшого пов'язаний з фізичною дією або подією. Він формує реакцію на якийсь чинник реального світу. Іноді при цьому один-єдиний датчик може згенерувати величезний обсяг даних, наприклад, акустичний датчик для профілактичного огляду обладнання. В інших випадках, всього одного біта даних достатньо, щоб передати життєво важливі відомості про стан здоров'я пацієнта (рис. 1.5.). Якою б не була ситуація, системи датчиків еволюціонували і, відповідно до закону Мура, зменшилися до субнанометрових розмірів і стали значно дешевшими. Саме до цього апелюють ті, хто прогнозує, що до Інтернету речей будуть підключені мільярди пристроїв, і саме тому ці прогнози виправдаються.

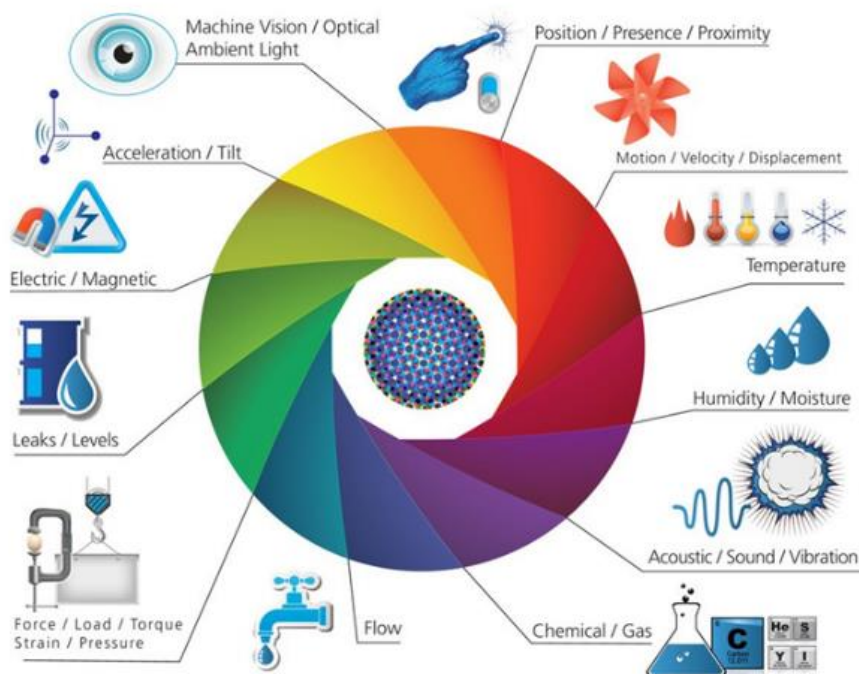


Рис. 1.5. Схематичне зображення сил та явищ, для обробки яких може використовуватись інтернет речей

Тому, розглядаючи інтернет речей, необхідно розглядати мікроелектромеханічні системи, датчики і інші типи недорогих граничних пристроїв та їх електрофізичних властивостей. Також це стосується силових і енергетичних систем, необхідних для живлення цих граничних пристроїв. Не можна вважати, що граничні пристрої забезпечуються енергією за замовчуванням. Мільярди маленьких датчиків все одно потребують великої кількості енергії. З питанням живлення також пов'язані питання організації хмарних сервісів IoT[1].

1.3.2. Передача даних. Велику увагу при розробці IoT приділяють встановленню з'єднання і роботі мереж. Інтернету речей не існувало б без надійних технологій передачі даних з найвіддаленіших і несприятливих областей в найбільші центри збору даних компаній Google, Amazon, Microsoft та IBM. Словосполучення «Інтернет речей» містить слово «Інтернет», тому необхідно вивчати питання, що стосуються мережних

технологій, обміну даними та навіть теорії сигналів. Базова опора Інтернету речей - не датчики і не програми, а можливість встановити з'єднання.

Передача даних і встановлення мережевого з'єднання базуються на системах зв'язку ближньої дії - персональних мереж (PAN), зазвичай побудованих без дотримання правил IP-протоколу. Це можуть бути як дротові, так і бездротові мережі. До бездротових IoT-мереж/протоколів, як правило, відносять протоколи Bluetooth, mesh-мережі, Zigbee, Z-Wave (рис. 1.6), для IIoT (Industrial Internet of Things) - також Wireless Hart та ISA100[2].

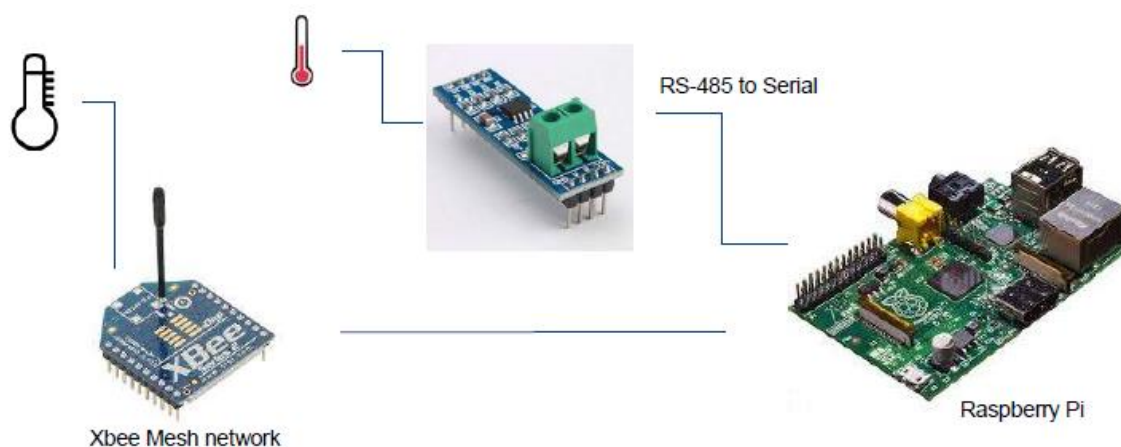


Рис. 1.6. Приклад пристроїв передачі даних в мережі PAN

Крім PAN, застосовують бездротові локальні мережі та системи зв'язку на основі IP-протоколу, включаючи широкий діапазон Wi-Fi-мереж на основі стандартів IEEE 802.11, 6LoWPAN і технології Thread. Нерідко користуються телекомунікаціями на основі стільникових стандартів (3G, 4G LTE) і новими стандарти, котрі забезпечують роботу Інтернету речей і міжмашинну взаємодію, такими як Cat-1 і Cat-NB, а також пропрієтарними протоколами LoRaWAN і Sigfox, котрі застосовують саме для IoT (рис 1.7).

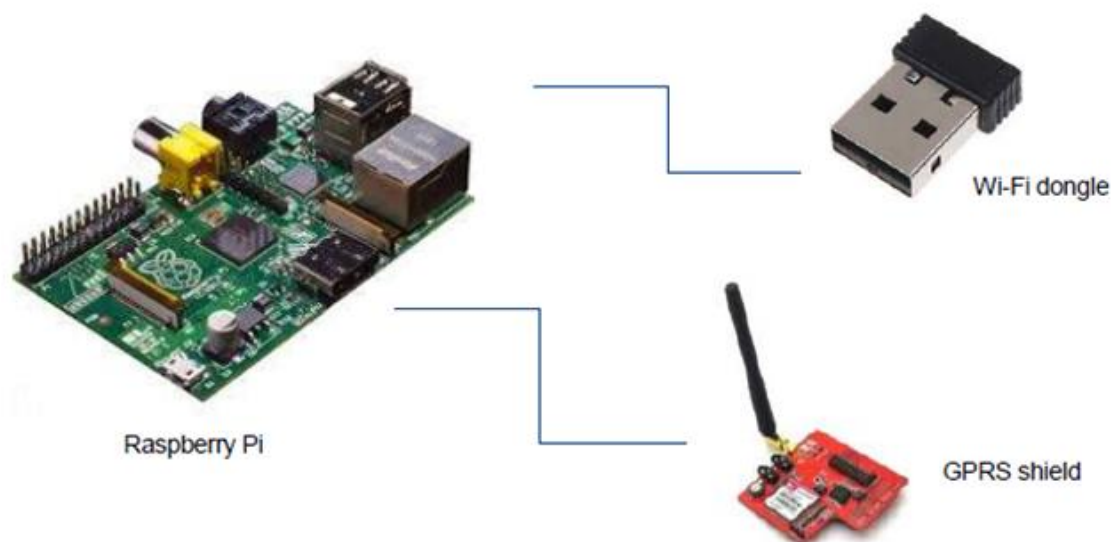


Рис. 1.7. Приклади пристрої для передачі даних в мережі LAN

1.3.3. Маршрутизація. Для передачі даних від датчиків в Інтернет-простір необхідно дві технології: маршрутизатор-шлюз і опорні інтернет-протоколи, що забезпечують ефективність обміну даними. Маршрутизатор особливо важливий в таких аспектах, як безпека, управління і напрям даних. Граничні маршрутизатори (Edge routers) керують і стежать за станом відповідних mesh-мереж, а також вирівнюють і підтримують якість даних. Також велике значення надають конфіденційності та безпеці даних. Маршрутизатор відіграє важливу роль у створенні віртуальних приватних мереж, віртуальних локальних мереж і програмно-визначених глобальних мереж. Вони можуть містити тисячі вузлів, котрі обслуговує єдиний граничний маршрутизатор, що служить розширенням для хмари (edge device)[2].

На цьому рівні застосовують низку протоколів, необхідних для обміну даними між вузлами, маршрутизаторами і хмарними сервісами у межах IoT-системи. Інтернет речей відкрив дорогу новим IoT-протоколам, які виходять на один рівень з традиційними протоколами HTTP і SNMP, які застосовують уже декілька десятків років. Для передачі IoT-даних потрібно

мати ефективні, енергозберігаючі протоколи з малою затримкою, здатні легко і безпечно відправляти дані у хмару і з неї. Зокрема, тут застосовують такі протоколи, як MQTT, AMQP і CoAP.

1.3.4. Туманні і граничні обчислення, аналітика і машинне навчання. На цьому етапі необхідно вирішити, що робити з потоком даних, що надходять у хмарний сервіс з граничного вузла (Edge Device). Щоб навчитися правильно оцінювати, як система буде розвиватися і рости, необхідно розібратися у всіх нюансах архітектури хмарних систем, а також, як впливає запізнювання на IoT-систему. Крім того, не все треба відправляти у хмару. Пересилання всіх IoT-даних обходиться значно дорожче, ніж їх обробка на кордоні мережі (граничні обчислення, Edge Computing) або включення граничного маршрутизатора в зону, яку обслуговує хмарний сервіс (туманні обчислення, Fog computing). Туманні обчислення також стандартизують, зокрема, існує стандарт туманних обчислень, приміром, архітектура OpenFog.

Дані, котрі отримано шляхом перетворення аналогового фізичного впливу у цифровий сигнал, можуть мати велику вагу. Саме тут у гру вступають засоби аналітики і процесори правил IoT-системи. Ступінь складності введення в дію IoT-системи залежить від того, яке рішення проектують. У деяких ситуаціях все досить просто: наприклад, коли на граничний маршрутизатор, який контролює декілька датчиків, потрібно встановити простий процесор правил, що відслідковує аномальні скачки температури. Інша ситуація - величезну кількість структурованих і неструктурованих даних у режимі реального часу передають у хмарне озеро даних, що вимагає високої швидкості обробки (для прогнозної аналітики) і довгострокового прогнозування на базі високотехнологічних моделей машинного навчання, таких як рекурентна нейронна мережа у пакеті аналізу сигналів з кореляцією в часі. Тут є певні проблеми і складнощі аналітики, які вирішують різними підходами і методами, наприклад,

складними обробниками подій, байесівськими мережами і формування нейронних мереж.

1.3.5. Загрози і безпека в IoT. Багато IoT-систем не обмежуватимуться безпечним простором будинку або офісу. Вони розташовуватимуться в громадських місцях, у дуже віддалених областях, у рухомих транспортних засобах або навіть всередині людини. Інтернет речей - величезна єдина ціль для будь-яких видів хакерських атак. Вже виявлено величезну кількість направлених на IoT-пристрої навчальних атак, добре організованих зломів і навіть вразливостей у системі безпеки національного масштабу. Розробник IoT рішень повинен знати особливості таких вразливостей і способи їх усунення, стандартні заходи, спрямовані на захист Інтернету речей або будь-якого компонента мережі.

1.4 Огляд технологій та протоколів передачі даних у мережі інтернету речей

1.4.1 Технології та протоколи передачі на малі відстані. Оскільки парадигма інтернету речей передбачає, що елементи знаходитимуться на невеликій відстані один від одного, та можуть бути не підключені до постійного джерела енергії. Отже, для таких пристроїв потрібно і специфічні технології, і протоколи передачі даних. Ось декілька популярних тепер технологій:

- Z-Wave - радіо технологія, що працює на частоті до 1ГГц та підходить для передавання простих команд з низькою затримкою;
- NFC - дозволяє передавати дані, застосовують на відстані до 20 см;
- RFID - технологія ідентифікації з допомогою радіосигналів;
- BLE - варіація технології Bluetooth з низьким енергоспоживанням;

- Wi-Fi HaLow - працює на частоті 900 МГц, має електроспоживання на рівні Bluetooth, та вищу, в порівнянні з Bluetooth, швидкість передачі даних.

Далі розглянемо технології детальніше, та визначимо, для чого їх потрібно.

1.4.1.1 Z-Wave. Бездротова радіо технологія з низьким енергоспоживанням. Z-Wave працює у діапазоні частот до 1 ГГц. Її оптимізовано для передавання простих команд управління з досить малими затримками (перемикання каналів, вимикання приладів, т. ін)[3].

Такі частоти вибрано не випадково, а для того, щоб зменшити кількість перешкод від інших технологій, якими вже користуються багато людей. Приміром, Wi-Fi працює на частотах 2,4 ГГц і 5 ГГц.

Хоча Z-Wave - складна система, нею дуже проста користуватися, а також вона ще є енергозберігаючою та економить наш дорогоцінний час. Система, що працює за допомогою дистанційного керування і користується радіохвилями малої потужності, є також і надійною майже на 100%, оскільки, ця мережа покриває всі області будинки, радіохвилі можуть проходити через стіни, поверхи та меблі,.

1.4.1.2 NFC. Near Field Communication, або система зв'язку на невеликих відстанях, призначена для обміну різною інформацією, наприклад, картинками, музичними файлами, номерами телефонів, або ключами цифрової авторизації між двома розташованими близько один до одного пристроями з підтримкою NFC. Це можуть бути смарт-картки, будь-які портативні пристрої, а також зчитувальні пристрої RFID. Такою технологією можна користуватися як ключем доступу до даних або служб[4].

Технологія також дозволяє передавати дані не тільки від активного пристрою до пасивного, але і між двома активними пристроями. NFC застосовують для взаємодії з пристроями радіочастотної ідентифікації RFID. Для забезпечення сумісності між картами RFID та мобільним

телефоном різних виробників перевіряють цифровий протокол і вимірюють усіх важливих властивості радіочастотного сигналу: тимчасові характеристики, чутливості та амплітуди приймача в активному режимі, частоти несучої амплітуди сигналу.

При передачі інформації від активного пристрою до пасивного пристрою застосовують амплітудну маніпуляцію ASK. При обміні обидва пристрої рівноправні. Кожен пристрій має власне джерело живлення, тому сигнал несучої відключають одразу після закінчення передачі (рис 1.8.).



Рис. 1.8. Принцип роботи NFC.

Внаслідок індуктивного зв'язку між опитуваним і прослуховуючим пристроями, пасивний пристрій впливає на активний. Зміна опору прослуховуючого пристрою викликає зміну амплітуди або фази напруги на антені опитуваного пристрою. Після цього, відбувається з'єднання двох пристроїв та передача даних. Як тільки ми роз'єднуємо два пристрої більш, ніж на 20 см, розривається електромагнітний зв'язок, і дані перестають передаватися автоматично.

Насправді, NFC можна вважати продовженням вже досить відомої технології радіочастотної ідентифікації RFID [18]. RFID знайдемо всюди у безконтактних картках і мітках. Однак, технологія NFC може не тільки зчитувати інформацію з будь-яких пасивних електронних міток, але, здатна забезпечувати двосторонній бездротовий зв'язок між пристроями.

1.4.1.3 RFID. Radio Frequency IDentification - метод автоматичної ідентифікації об'єктів, у якому за допомогою радіосигналів зчитують або записують дані, котрі зберігають у транспондерах, або RFID - мітках [19]. Будь-яка RFID-система складається із зчитувального пристрою (зчитувач, рідер) та транспондери (він же RFID-мітка, іноді також називають RFID-тег)[5].

Більшість RFID-міток складається з двох частин. Перша - інтегральна схема для обробки та зберігання інформації, демодулювання та модулювання радіочастотного сигналу і деяких інших функцій. Друга - антена для прийому і передачі сигналу. А також, для роботи цих міток потрібно програмне забезпечення — програми, за допомогою яких збирають інформацію та аналізують, отриману із RFID-міток.

Мітки бувають двох видів - активні та пасивні:

- Активні мітки мають власне джерело живлення, тому вони можуть самі посилати сигнал і зчитуватися з досить великої відстані;
- Пасивні мітки не мають власного джерела енергії і активізуються тільки після того, коли надходить сигнал до пристрою зчитування, і тоді передають записану в них інформацію.

RFID-мітки (рис. 1.9) можуть керувати товарними запасами або відстежувати часу на спортивних змаганнях [5]. Магнітні мітки не замінюють штрих-коди, а доповнюють їх можливістю дистанційного зчитування. Мітками можуть маркувати велику рогату худобу для запису інформації про проходження ветеринарного огляду. Рішення для транспорту допомагають ідентифікувати автомобіль, навіть якщо він рухається з великою швидкістю. Деякі авіалінії користуються мітками для ефективного відстеження великих потоків багажу. Також RFID вбудовують у біометричні паспорти, кредитні картки для безпечного доступу у певні області.

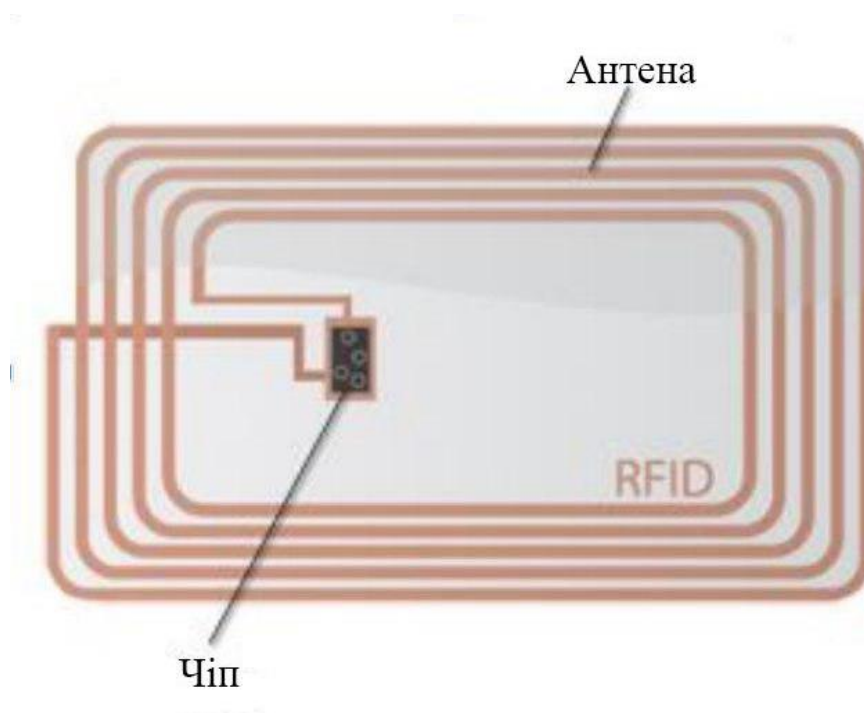


Рис. 1.9. Будова пасивної RFID мітки.

1.4.1.4 BLE, Bluetooth Low Energy. Бездротова технологія з низьким енергоспоживанням Bluetooth (BLE) – частина специфікації Bluetooth, котра починається з покоління Bluetooth 4.0 і тепер закінчується Bluetooth 5.0 [6].

Пристрої, котрі користуються BLE, споживають менше енергії, ніж інші версії Bluetooth-пристрої попередніх поколінь. У багатьох випадках пристрої зможуть працювати більше року на одній невеличкій батарейці типу таблетка без підзарядки. Завдяки цьому, можна користуватися датчиками невеликих розмірів, котрі постійно працюватимуть та взаємодіятимуть із іншими пристроями.

BLE призначено для тих пристроїв, котрі мають невеликі розміри, тобто для пристроїв, у яких важлива компактність і куди не можна встановити повноцінний акумулятор або батарею великого об'єму [6].

Bluetooth LE споживає в 10-20 разів менше енергії і здатний передавати дані в 50 і більше разів швидше та на відстані більше 100 метрів, ніж класичні Bluetooth рішення. Крім перерахованих вище переваг, BLE має високу безпеку, надійність, низьку затримку при підключенні та низьку споживчу потужність.

Ще одна важлива особливість стандарту полягає в адаптивності переналаштування частоти, тобто, відбувається захист від помилок при передачі сигналу, BLE швидко змінює свою робочу частоту, вибираючи найоптимальнішу для усунення перешкод, проблем переповнення і для зниження інтерференції.

Специфікацію Bluetooth 5.0 створено з орієнтацією на Інтернет речей. Це остаточно показало, що стандарт прагне "захопити" ринок пристроїв.

Порівняно із попередньою версією 4.0, підвищено швидкість передачі даних майже до швидкостей HSPA і LTE ранніх версій, тоді як енергоспоживання залишилося у колишніх показниках. Важливим показником для побудови мереж Інтернету речей є енергоефективність. В даний момент така специфікація є мало поширеною через те, що вона з'явилася нещодавно. Bluetooth 5, як і всі попередні версії, має зворотну сумісність. Цілком можливо, що через декілька років кожний мобільний пристрій підтримуватиме п'яту версію цього стандарту, що є найважливішою перевагою такої технології над іншими.

1.4.1.5 Wi-Fi HaLow. Протокол бездротової мережі, котрий опубліковано у 2017 році як доповнення до стандарту бездротової мережі IEEE_802.11 [7]. Протокол працює на частоті 900 МГц, котру не потрібно ліцензувати і забезпечує розширений діапазон Wi-Fi мереж, порівняно із звичайними мережами Wi-Fi, котрі працюють у діапазонах 2,4 ГГц і 5 ГГц. Його низьке енергоспоживання також є перевагою, котра дозволяє створювати великі групи станцій або датчиків, які взаємодіють і поширюють сигнали, підтримуючи концепцію інтернету-речей (Internet of

Things, IoT). Низьке енергоспоживання протоколу конкурує з Bluetooth і має додаткову перевагу - вищі швидкості передачі даних і більш широкий діапазон покриття (рис 1.10).

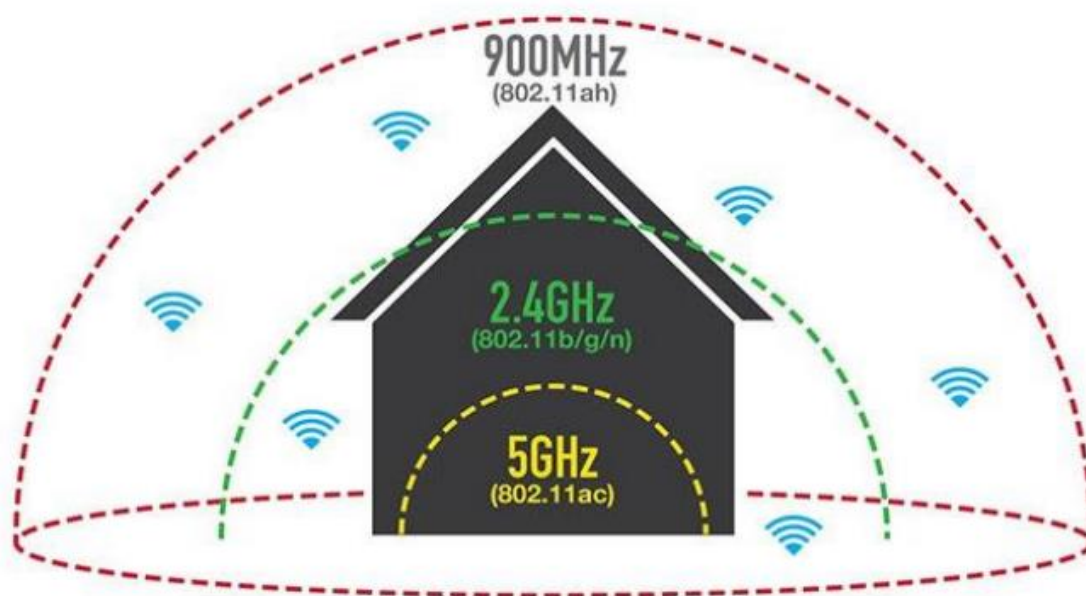


Рис. 1.10. Порівняння різних протоколів Wi-Fi за зоною покриття

Wi-Fi HaLow розширює Wi-Fi в діапазоні 900 МГц, надаючи можливість з'єднувати пристрої малої потужності, такі, як датчики та портативні комп'ютери. Wi-Fi HaLow успадковує позитивні якості попередніх протоколів, такі як надійний захист інформації, широку сумісність обладнання та простоту встановлення. Пристрої з підтримкою Wi-Fi HaLow також працюватимуть у діапазонах 2,4 та 5 ГГц, що надає можливість інтегрувати їх в екосистему, яка тепер налічує більше 7 млрд пристроїв. Також Wi-Fi HaLow матиме підтримку підключення по IP. Це дозволить працювати з хмарами, що дуже важливо для IoT. Також можна підключати до однієї точки доступу близько 1000 пристроїв.

1.4.1.6. Порівняльна характеристика технологій та протоколів передачі даних на малі відстані в IoT. Порівняльна характеристика мереж ближньої дії наведена у табл. 1.1.

Таблиця 1.1

Порівняння основних технічних характеристик мереж з низькою дальністю дії

Характеристики	RFID	NFC	BLE	Z-Wave	Wi-Fi HaLow
Смуга частот	6/13.5/433/863-870/902-928 МГц 2.4/5-27 ГГц	13.56 МГц	2.4 ГГц	868/915 МГц	Піддіапазон 1 ГГц
Швидкість передачі даних	500 кбіт/с	106/212/424/848 кбіт/с	1 Мбіт/с	9.6,40 та 100 кбіт/с	До 4 Мбіт/с
Радіус дії	0.1 - 5 м	0.1 м	70 м	100 м	100 – 1000 м
Пропускна здатність на канал	10 МГц для 6 МГц 14 МГц для 13.5 МГц 1.74 МГц для 433 МГц 7 МГц для 800 МГц 8 МГц для 2.4 ГГц 5 – 27 ГГц сегментований	Змінна	40 каналів з шириною в 2 МГц	300,400 кГц	1/2/4/8/16 МГц
Модуляція	-	ASK, BPSK	GFSK	FSK/GFSK	BPSK, QPSK, 16-/64-/256- QAM, OFDM
Топологія	Point to Point Point to Multipoint	Peer-to-peer	Single-hop	Mesh	Star
Безпека	Шифрування	Шифрування	AES-128	AES-128	WPA

1.5 Протоколи для передачі повідомлень в мережі інтернету речей

Обсяг інформації, що формується одним сенсорним вузлом, порівняно невеликий, однак більшість сервісів Інтернету речей побудовано на принципі обробки інформації від великої кількості клієнтів або давачів, що принципово відрізняється від архітектур, прийнятих в класичних мережах.

Таким чином, стикаємося з новою архітектурою: багато джерел - багато одержувачів, крім того, обсяг трафіку від сенсорного вузла може бути як дуже маленьким, так і дуже великим. У такому випадку звичними прикладними протоколами для передачі повідомлень користуватися не можна.

Базова топологія для передачі повідомлень у мережі інтернету речей має назву “видавець-підписник” (Publisher-Subscriber, або pub/sub) (див. рис. 1.11). У такій схемі введено поняття видавця – джерела інформації та передплатника – одержувача інформації. Термін передплата пов'язаний з певною операцією, котру виконують учасники, з метою отримати інформацію передплатником від конкретного видавця, а також упорядкувати збір інформації – параметри періодичності отримання та аналогічних (в залежності від реалізації) показників.

У даному випадку розглядають ситуацію, коли сенсорний вузол (Node) об'єднує інформацію від багатьох датчиків (наприклад, дані вологості повітря) і направляють її згідно з параметрами передплати або за запитом, або самостійно через певний інтервал часу. Зазвичай самі датчики досить примітивні, їх завдання зводять до постійної передачі інформації про контрольовані параметри. Тому з'являється необхідність об'єднувати датчики у вузли, оснащені мікроконтролерами, які відповідатимуть за зчитування вимірюваних даних і відправку їх за заздальгідь визначеними алгоритмами далі на сервер.

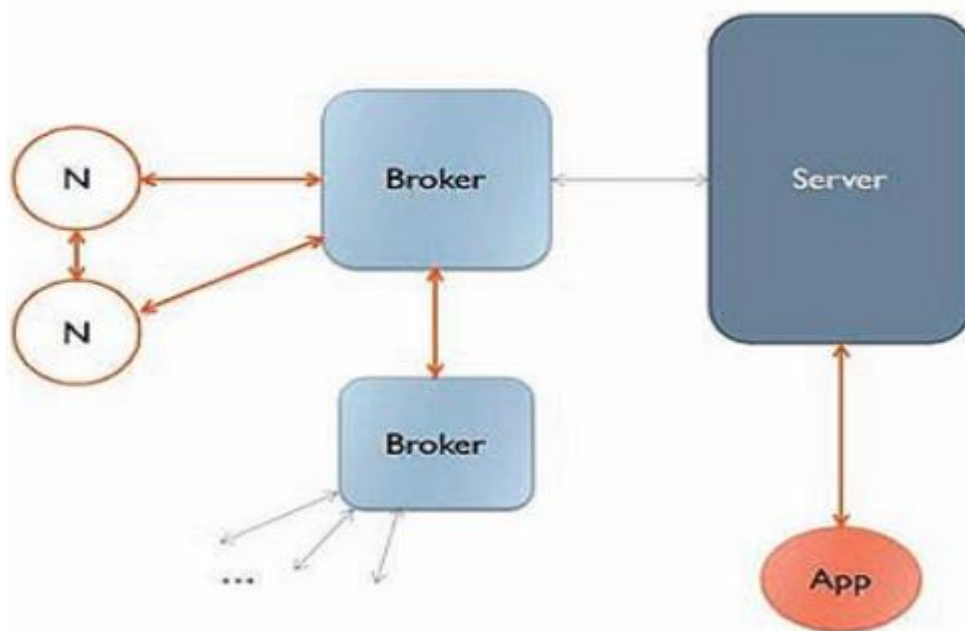


Рис. 1.11 Топологія системи, яка виконує передачу даних між різними вузлами мережі IoT

Також найчастіше для взаємодії клієнта з системою необхідно ще мати клієнтську програму (Application), встановлену на персональному пристрої, яка наочно представлятиме інформації, одержану від датчиків, або вже опрацьовану сервером управління системою. Така топологія також розрахована на включення брокера (Broker). Брокер – сервер, котрий приймає інформацію від видавців і передає її відповідним передплатникам. У складних системах брокер може виконувати також різні операції, пов'язані з аналізом та опрацюванням даних, котрі надійшли на сервер. Брокер може встановлювати пріоритети сполученням і формувати черги для передачі повідомлень. Таким чином, брокер організовує пересилання повідомлень, їх зберігання та фільтрацію. Черга повідомлень – контейнер, або блок, в якому зберігають повідомлення у процесі їх пересилання. При недостатньому ресурсі каналу зв'язку, або якщо одержувач недоступний під

час того, як надсилають повідомлення, черга зберігає повідомлення до того часу, поки його не доставлять до одержувача.

Протоколи, котрі застосовують у мережі інтернету речей:

- DDS – протокол прикладного рівня M2M для систем реального часу;
- XMPP – протокол для миттєвого обміну повідомленнями та інформацією про присутність у режимі, близькому до режиму реального часу;
- CoAP – створений для мереж і пристроїв з обмеженими ресурсами.
- MQTT – легкий, компактний і відкритий протокол обміну даними, створений для передачі даних на віддалених локаціях;
- STOMP – організовує зв'язок з брокером за методом "запит-відповідь";
- SOAP – протокол обміну структурованими та довільними повідомленнями формату XML у розподіленому обчислювальному середовищі.

Далі розглянемо найпопулярніші протоколи.

1.5.1 DDS (Data Distribution Service). Протокол прикладного рівня M2M для систем реального часу, котрий базується на моделі "видавець-передплатник". Основна функція протоколу полягає в тому, щоб здійснити з'єднання пристроїв з іншими пристроями за допомогою шини обміну повідомленнями (див рис. 1.12). Протокол DDS може ефективно та синхронно доставляти мільйони повідомлень в секунду.

Пристрої дають запит на дані інакше, ніж в IT мережах. По-перше, пристрої працюють швидко. Масштаб реального часу часто вимірюють у долях мікросекунд. Пристроєм потрібно здійснювати зв'язок з іншими пристроями, використовуючи складні шляхи, тому прості і надійні двоточкові TCP потоки даних обмежують можливості для такої передачі.

Натомість DDS забезпечує деталізований контроль якості обслуговування (QoS), багатоадресну передачу, переналаштовану надійність і всеосяжну надмірність. Крім того, сильною стороною DDS є розгалуження даних. Протокол DDS забезпечує потужні способи фільтрації та відбору даних за адресами призначення, причому число синхронних одержувачів даних може обчислюватися тисячами. Деякі пристрої досить компактні, тому існують полегшені версії протоколу DDS, які працюють в умовах обмеженого обсягу.

Для того, щоб отримувати дані від пристроїв, зіркоподібна мережа зовсім не підходить. Замість цього, DDS реалізує пряму шинний зв'язок між пристроями на базі реляційної моделі даних. Її називають шиною даних (DataBus), оскільки це мережевий аналог бази даних (database).

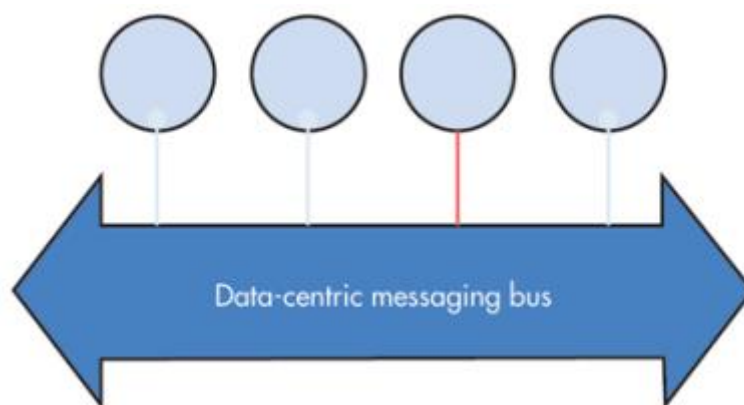


Рис. 1.12 Принцип з'єднання пристроїв за допомогою протоколу DDS

Протокол DDS застосовують у високопродуктивних пристроях. Це єдина технологія, яка забезпечує гнучкість, надійність та швидкість, необхідні для побудови складних додатків реального часу. До таких додатків належать військові системи, вітроелектростанції, інтегровані системи лікарень, системи діагностичної візуалізації, системи супроводження ресурсів, автомобільні системи випробувань і забезпечення безпеки.

1.5.2 MQTT (Message Queue Telemetry Transport) – легкий, компактний і відкритий протокол обміну даними, створений для передачі даних на віддалених локаціях, де потрібно невеликий розмір коду і пропускна здатність каналу обмежена. Перераховані вище вимоги дозволяють застосовувати його в системах M2M (машина-машина)[9-10].

Також існує версія протоколу MQTT-SN (MQTT для мереж датчиків), раніше відома як MQTT-S, призначена для вбудованих бездротових пристроїв без підтримки TCP/IP мереж.

Спрощений процес роботи протоколу MQTT:

Видавець передає повідомлення з певними даними (наприклад, інформація з датчиків вологості) на брокера, вказуючи при цьому тему (Topic), до якої ці дані відносяться (наприклад, "вологість") [].

Брокер аналізує, які із передплатників мають підписку на певні теми, в даному випадку – на тему "вологість".

Передплатникам, які підписані на тему "вологість", брокером буде відправлено повідомлення з інформацією від датчиків вологості.

Таким чином, велика кількість передплатників може підписатися на різноманітні теми і, залежно від таких передплат, отримувати необхідну їм інформацію, не спілкуючись з видавцем безпосередньо.

1.5.3 STOMP (Simple Text Oriented Message Protocol). Простий протокол обміну повідомленнями, що передбачає широку взаємодію з багатьма мовами, платформами та брокерами []. Такий протокол підходить під шаблон "видавець-передплатник" (див рис. 1.13) і за допомогою повідомлень SEND (відправити), SUBSCRIBE (підписатися), UNSUBSCRIBE (відписатися), BEGIN (почати), ABORT (переривати), ACK (підтвердити), NACK (не підтвердити), DISCONNECT (відключити) організовує зв'язок з брокером за методом "запит-відповідь".

Протокол схожий на HTTP, використовує транспорт TCP, є простим текстовим протоколом, що дозволяє клієнтам STOMP спілкуватися з будь

яким брокером повідомлень, котрі підтримують цей протокол. Таким чином, цей спосіб взаємодії, розроблений для обміну повідомленнями між платформою, написаною одною мовою програмування, і клієнтом, програмне забезпечення якого створене іншою мовою. Підтримує велику кількість сумісних клієнтських бібліотек.

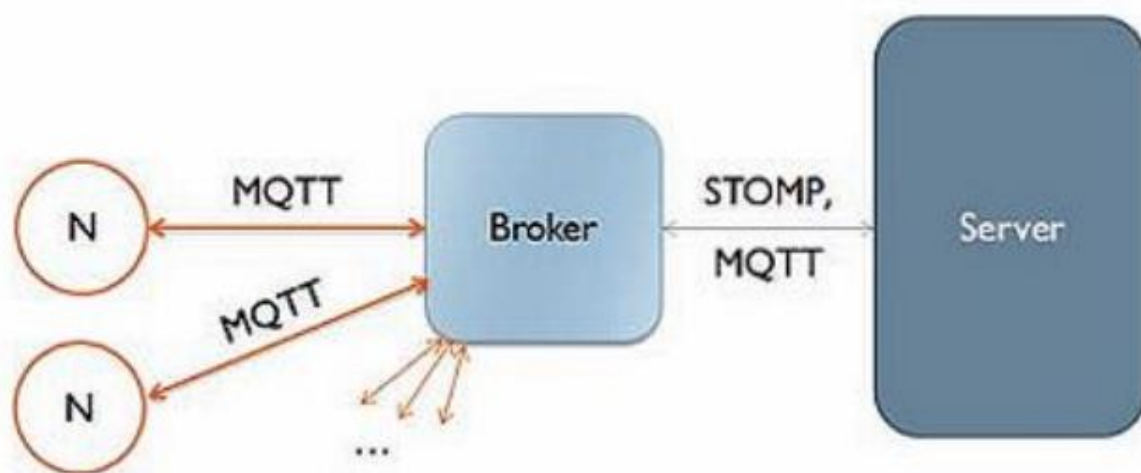


Рис. 1.13. Сегмент мережі, де застосовують протоколи MQTT та STOMP

1.5.4. Огляд операції, що можуть здійснюватись протоколами та їх призначення. З огляду згаданих вище пристроїв виходить, що кожен протокол має свою сферу застосувань. DDS призначено для комунікації між пристроями, де потрібно високу швидкість обміну великою кількістю даних, гнучкість, масштабованість. MQTT застосовують для дуже завантажених мереж, де є велика кількість пристроїв та брокерів, також якщо пропускна здатність каналу обмежена та кількість коду невелика. У свою чергу STOMP потрібно тоді, коли необхідно спростити системи обміну повідомленнями, та уніфікувати повідомлень від різних протоколів.

Далі у табл. 1.2 наведено призначення та операції, котрі можуть виконуватись даними протоколами.

Таблиця 1.2

**Призначення та операції, котрі виконуються аналізованими
протоколами**

Протокол	Призначення	Операції
DDS	Для мереж, що потребують розподіленого навантаження	Процедури отримання та відправки даних
MQTT	Для завантажених мереж з великою кількістю пристроїв та брокером	Процедури обробки публікацій/передплати
STOMP	Для мереж, в яких є можливість застосовувати декілька комбінацій різних протоколів, що потребують простий протокол передачі повідомлень через брокера	Процедури публікацій/передплати. Операції з транзакціями

1.5.5 Висновки після огляду технологій та протоколів передачі даних. Для комунікації в мережі інтернету речей, користуються різними технологіями та протоколами передачі даних, залежно від обставин та середовища, в яких будуватимуть мережу.

Для мережі у межах одного будинку, найліпше підходять технологія передачі даних Wi-Fi HaLow та протокол MQTT. Для мережі PAN найліпше створювати на основі BLE та RFID, технологій, що дозволяють пристроям, котрі знаходяться на невеликій відстані, спілкуватися між собою, та протоколи MQTT та DDS для передачі команд та даних.

Протокол STOMP дозволяє підключатися до різних пристроїв та абстрагуватися від протоколів нижчого рівня, що дозволяє уніфікувати доступу до сервісів.

1.6 Огляд існуючих засобів моніторингу та управління елементами інтернету речей

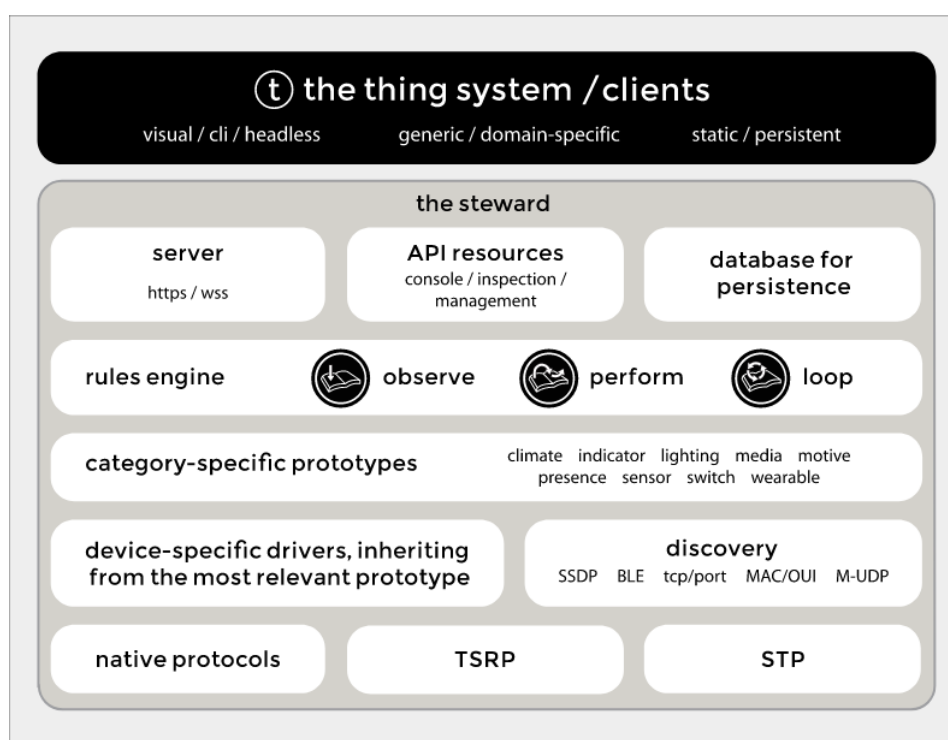
1.6.1 The Thing System. Система моніторингу екосистеми інтернету речей, розроблена групою американських інженерів, яку створено разом із Денні Гудманом, автором книг із JavaScript і HTML.

Мета проекту - об'єднати гаджети/елементи інтернету речей, що знаходяться у системах розумного будинку з централізованим управлінням.

Пристрої різних виробників часто не можуть взаємодіяти один з одним і працюють не синхронізовано. Щоб розв'язати таку задачу, автори створили ПЗ, яке може працювати з різними мережевими протоколами, гаджетами і клієнтським додатками.

Список пристроїв, котрі підтримує проект, можна знайти на сайті проекту.

Рис. 1.14. Діаграма модулів додатку Steward



Програмне забезпечення Steward, основне ПЗ the thing system, написано на node.js, тому воно портативне та легко розширюється. Воно працюватиме, як на ноутбучі, так на віддаленому сервері або на одноплатному комп'ютері Raspberry Pi.

Переваги такого ПЗ:

- Невеликі розміри;
- Портативність та мультиплатформеність;
- Підходить для різних пристроїв мережі інтернету речей;
- Легко налаштовувати та встановлювати.

Недоліки ПЗ:

- Систему розраховано на пристрої мережі розумного будинку і при розширенні на більш серйозні мережі виникатимуть проблеми з конфігурацією;
- Проблеми з продуктивності при розширенні системи, оскільки платформу написано на node.js;
- Система підтримує незначну кількість пристроїв.

Отже, система ідеально підходить для розумного будинку та простих локальних мереж. Однак при масштабуванні або передачі великої кількості даних можуть виникнути проблеми з продуктивністю.

1.6.2 Domoticz. Система автоматизації розумного будинку з відкритим кодом, котру створено для контролю різних пристроїв та опрацювання вхідних сигналів з різноманітних сенсорів. Система підтримує велику кількість пристроїв різних виробників. Також система має відкритий код, основна частина якого написана на C++, який поширюється під ліцензією “GNU GENERAL PUBLIC LICENSE”. Користувацький

інтерфейс є HTML5 веб-інтерфейсом, що автоматично адаптується під різні пристрої[15].

Така система працюватиме на Windows, Apple, Cubieboard, Raspberry Pi та на всіх Unix-подібних операційних системах. Також, система має додатки для мобільних платформ, що дозволяють керувати нею віддалено.

Система дозволяє користувачеві програмувати обробку подій та повідомлень, що надходять з пристроїв та давачів. Програмувати потрібну користувачу логіку можна у візуальному додатку Blockly (див. рис. 1.15).

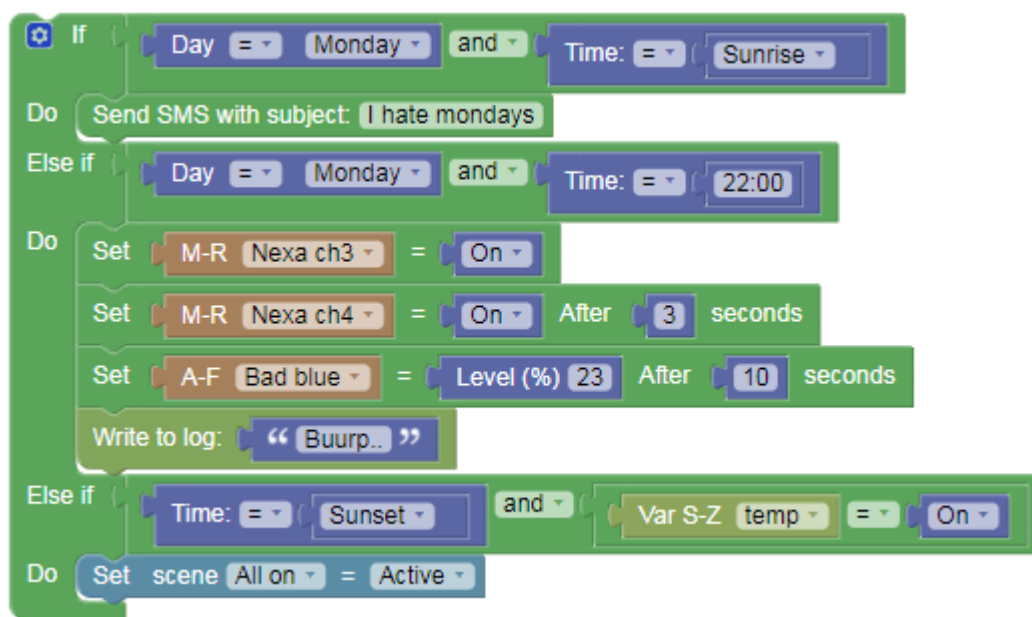


Рис. 1.15. Приклад візуального програмування з допомогою веб-додатку “Blockly”

Переваги системи:

- Мультиплатформність;
- Зручний веб-інтерфейс;
- Підтримка великої кількості вже готових пристроїв;
- Можливість писати драйвери\скрипти для своїх пристроїв;
- Легко встановлювати та налаштувати;
- Легко масштабувати системи для різних завдань;

- Можливість гнучко налаштувати поведінку пристроїв завдяки програмування з допомогою логічних блоків;
- Відкритість коду системи, що дозволяє самостійно правити і додавати функціонал.

Недоліки системи:

- Оскільки інтерфейс написано на HTML5, то не підтримуються браузери старих версій;
- При встановленні на платформи з малим об'ємом оперативної пам'яті можуть виникнути проблеми з продуктивністю.

Особливістю системи є можливість керувати нею з допомогою асинхронних запитів, що у собі містять JSON. Система містить детальну документацію стосовно усіх запитів, якими можна керувати системою.

1.7 Висновок до розділу

У розділі розглянуто та проаналізовано архітектуру мережі Інтернету речей. Наведено основні протоколи, котрі застосовують для побудови таких мереж. На малій дистанції для комунікації між пристроями найкраще використовувати технології BLE та NFC, оскільки вони надають найкраще співвідношення швидкості передачі даних та енергоспоживання. Для передачі даних на середніх дистанціях найкращим вибором буде Wi-Fi HaLow оскільки при споживанні енергії на рівні Bluetooth має вищу швидкість передачі даних та більший радіус дії.

Також розглянуто існуючі засоби моніторингу та управління елементами інтернету речей. Серед яких можна виділити Domoticz, оскільки він має відкриту кодову базу, підтримує велику кількість пристроїв та має власний програмний інтерфейс, який дозволяє керувати системою віддалено.

РОЗДІЛ 2

ОПИС І АНАЛІЗ АПАРАТНОЇ ЧАСТИНИ ТА НАЛАШТУВАННЯ ЗАСОБІВ МОНІТОРИНГУ ТА УПРАВЛІННЯ ЕЛЕМЕНТАМИ ІОТ

2.1 Опис вибраної платформи Raspberry Pi

Raspberry Pi model B - одноплатний комп'ютер, що на початку свого існування призначався для вивчення у школах базових принципів комп'ютерних наук (рис. 2.1). З розвитком мобільних процесорів, став однією з найулюбленіших платформ для ентузіастів, що розробляють додатки для інтернету речей. Основні переваги платформи: ціна, універсальність та відкритість[11].



Рис. 2.1. Зовнішній вигляд плати RaspberryPi

Відмінні особливості платформи:

- Комп'ютер може працювати на базі Unix подібних систем;

- Може відображати Full HD відео;
- Об'єм ОЗУ від 512 МБ;
- Розміри плати 85,6 * 54 * 17 мм;
- Є підтримка композитного відеовиходу RCA і HDMI для підключення монітору, а також 3.5-міліметровий роз'єм для підключення пристроїв відтворення звуку.

Живлення плати може відбуватися декількома способами. Перший це з допомогою роз'єма micro-USB, який приймає 5В. Другий це через універсальні піни. Третій, через адаптер живлення на 5В і 2А. Перші два способи не гарантують стабільне живлення при підключенні великої кількості периферійних пристроїв та датчиків. Третій спосіб найстабільніший, оскільки надає хорошу енергоємність для пристроїв що будуть підключатись[12].

Плата підтримує велику кількість інтерфейсів(рис 2.2):

- UART
- Слот для карт SD, MMC, SDIO
- Роз'єм HDMI
- 3.5мм стерео
- Композитний відеовихід
- 2x USB 2.0
- 10/100Mb RJ45 Ethernet
- Wi-Fi 802.11n
- Bluetooth 4.1
- 16 портів input/output 3.3В
- Інтерфейси I2C і SPI
- Інтерфейс ARM JTAG
- Інтерфейс DSI
- Інтерфейс MIPI CSI-2

Інтерфейси Wi-Fi 802.11n і Bluetooth 4.1 забезпечуються окремою мікросхемою Broadcom BCM43438, яка доставляється на плату через інтерфейс Shield або окремими модулями, що підключаються як периферія.

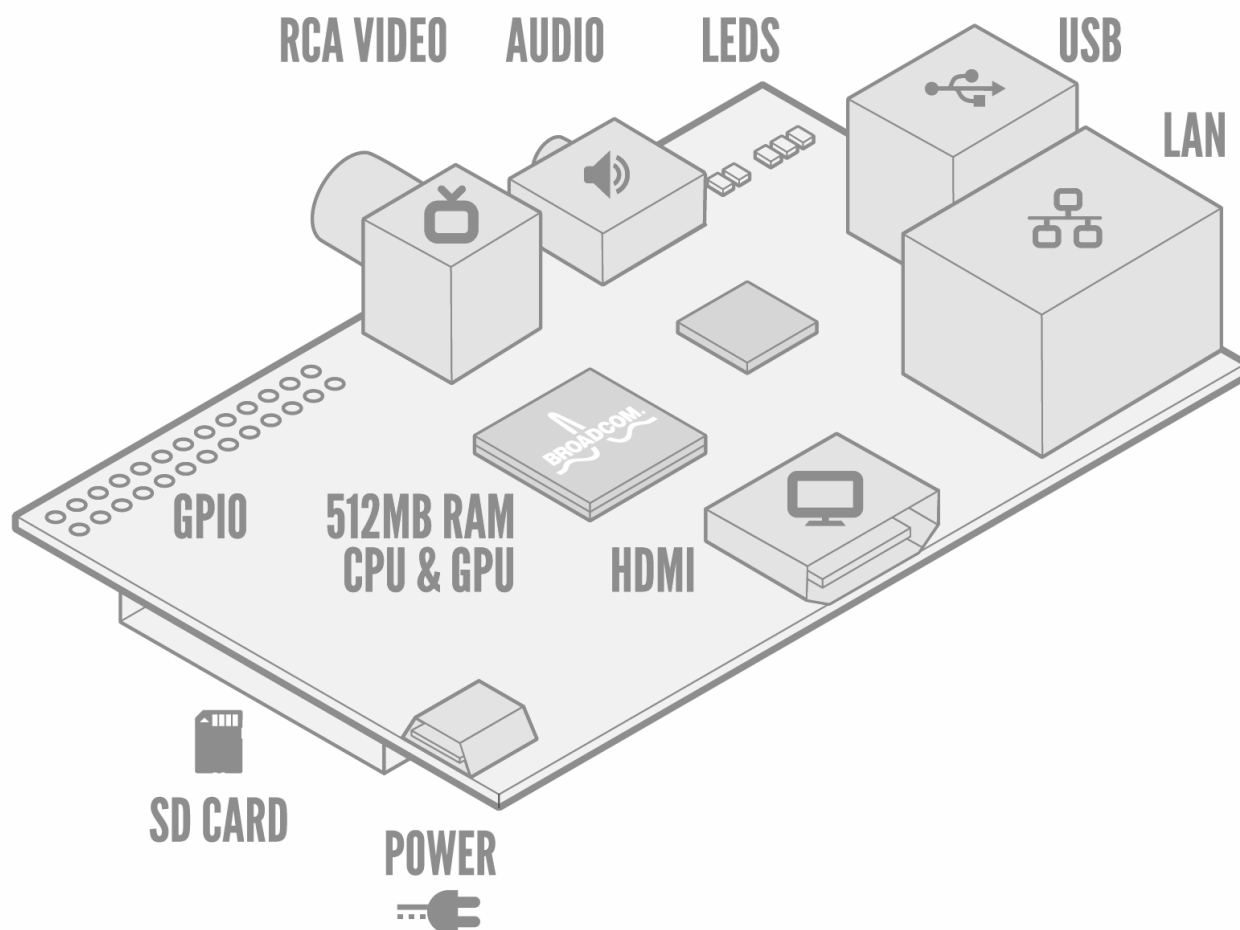


Рис. 2.2. Схема розміщення інтерфейсів на платі

Процесор Raspberry Pi B використовує 32-розрядний процесор ARM Cortex-A7 900 МГц. Він має кеш першого рівня (L1) 16 Кб, а кеш другого рівня (L2) 128 Кб. Кеш-пам'ять другого рівня використовується насамперед графічним процесором. SoC розміщується під мікросхемою оперативної пам'яті, тому його не видно.

Продуктивність. Під час роботи на частоті 700 МГц, Raspberry Pi забезпечила реальну продуктивність, приблизно еквівалентну 0,041

GFLOPS. На рівні центрального процесора продуктивність аналогічна Pentium II 300 МГц 1997-99. GPU забезпечує 1 Gpixel/s або 1.5 Gtexel/s обробки графіки або 24 GFLOPS загальної обчислювальної продуктивності.

Програмне забезпечення. Raspberry Pi працює в основному на операційних системах, заснованих на Linux ядрі. Також можлива установка Windows 10 IoT. Більш того, існують Raspberry з ліцензійною Windows 10 IoT. ARM11 заснований на 6 версії ARM, який завжди підтримує усі різновиди Linux. Для установки операційних систем існує інструмент NOOBS, або інші аналоги для запису образів ОС на носії, що дозволяють записувати на карти пам'яті операційні системи сумісні з Raspberry Pi.

2.2 Установка та налаштування Raspberry Pi

2.2.1 Підготовка карти пам'яті. Так як у Raspberry Pi немає вбудованої постійної пам'яті, для роботи комп'ютера попередньо необхідно підготувати карту пам'яті - розпакувати на неї образ бажаної операційної системи[12-13]. Для підготовки карти пам'яті потрібно:

1. SD (MMC/SDIO) карта пам'яті об'ємом від 8 до 32Гб, з високим класом швидкості читання, не нижче 8;
2. Картридер для підключення карти до комп'ютера;
3. Програма Win32DiskImager;
4. Образ операційної системи, буде використовуватись Raspbian.

Далі карту потрібно підключити до комп'ютера з допомогою картридера, та за допомогою Win32DiskImager записати на карту образ операційної системи.

2.2.2 Перше підключення Raspberry Pi, та первинне налаштування. Для першого запуску та налаштування окрім самої плати та карти пам'яті з операційною системою, знадобиться USB-клавіатура, монітор з HDMI

виходом або TV тюнер з RSA роз'ємом і кабель для підключення, та блок живлення 5В та мінімум 0.5А[13].

Після підключення всіх елементів та ввімкнення живлення, зразу почнеться загрузка операційної системи (рис 2.3).

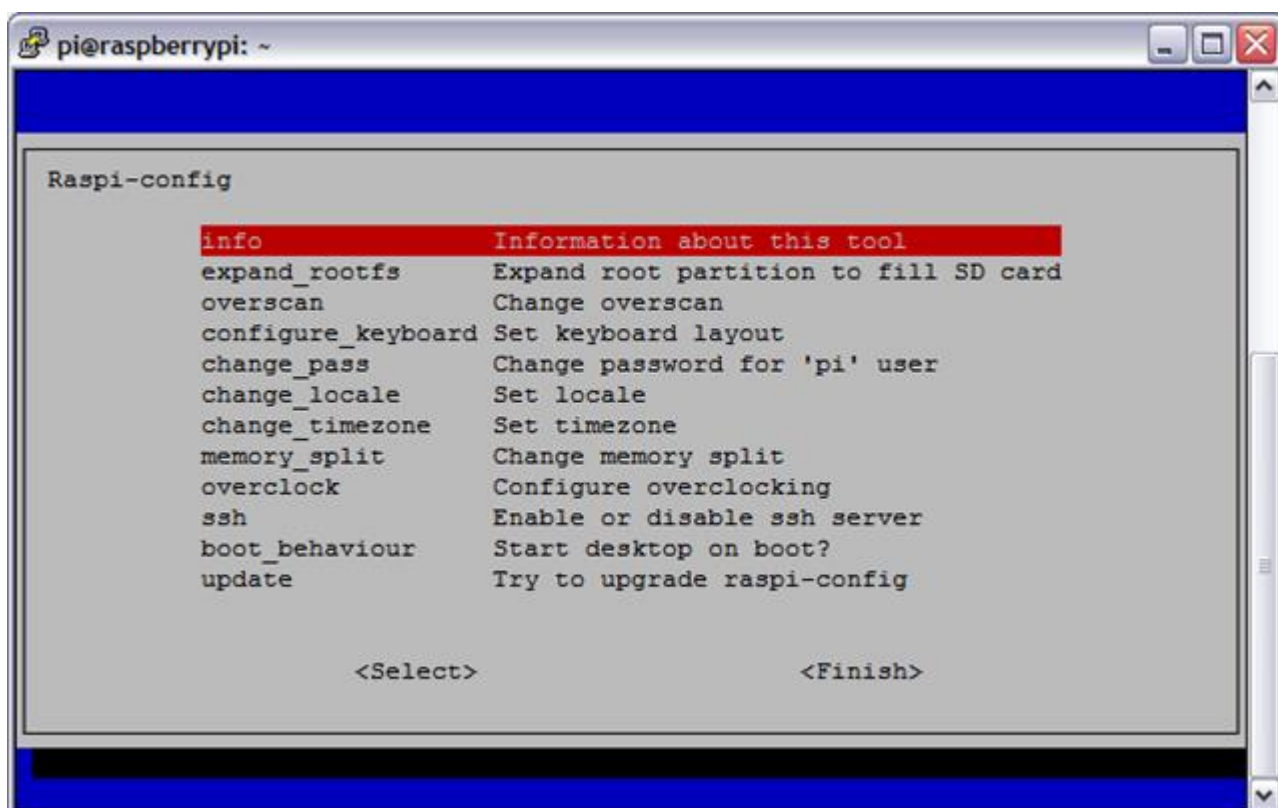


Рис. 2.3 меню первинної конфігурації Raspberry Pi

Розглянемо пункти меню, що використовуються для налаштування:

- expand_rootfs, тут можна збільшити root розмір на весь розмір карти пам'яті;
- configure_keyboard, в цьому пункті ви можете вибрати драйвер клавіатури, якщо варіант за замовчуванням вас не влаштовує;
- change_pass, заміна паролю користувача "pi", при вводі пароль не відображається, та ввести пароль потрібно двічі;
- change_locale, зміна мови системи;

- `change_timezone`, зміна часового поясу, плата немає свої часових поясів, тому підгружає їх з інтернету;
- `memory_split`, розподіл пам'яті;
- `overclock`, розгін процесора;
- `ssh`, ввімкнення або вимкнення SSH сервера, рекомендується включити, якщо в майбутньому буде використовуватись віддалене керування;
- `boot_behaviour`, завантаження візуальної оболонки при завантаженні системи.

Після закінчення налаштувань, потрібно перезавантажити систему, для цього потрібно натиснути комбінацію клавіш `Ctrl+F`, та підтвердити вибір. Після чого плата перезавантажиться.

Далі після загрузки системи потрібно встановити пароль для користувача "root". Для цього в командному рядку потрібно ввести команду "sudo passwd root", та ввести пароль двічі. Для перегляду процесів, що наразі запущені в системі потрібно набрати команду "top" (рис 2.4).

2.3 Підключення зовнішнього диска та модуля Wi-Fi

Для безпроводового доступу використовуватиметься Wi-Fi адаптер для Raspberry Pi `Auspi Wireless Adaptor– 802.11 B/G/N`. Його перевагою є робота у діапазонах b, g, n, сумісність з USB роз'ємом та те що він не потребує окремого живлення[13].

Зовнішній диск може бути будь якого формату, основне щоб була можливість його підключення через USB.

```

pi@raspberrypi: ~
top - 13:32:51 up 18:08, 1 user, load average: 2,05, 2,33, 1,83
Tasks: 73 total, 1 running, 72 sleeping, 0 stopped, 0 zombie
%Cpu(s): 20,4 us, 2,3 sy, 0,0 ni, 63,9 id, 12,4 wa, 0,0 hi, 1,0 si, 0,0 st
KiB Mem: 189100 total, 175640 used, 13460 free, 2960 buffers
KiB Swap: 102396 total, 22272 used, 80124 free, 56084 cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 23225 eggdrop  20   0 13380 3468 1304  S   15,8   1,8   1:43.20 eggdrop
  2516 debian-t  20   0 46588  13m 1244  S    4,6   7,3  38:55.79 transmission-da
  7958 pi        20   0  6348 1368 1040  R    1,0   0,7   0:00.14 top
    32 root     20   0     0     0     0  D    0,7   0,0  30:01.47 mmcqd/0
    33 root     20   0     0     0     0  S    0,3   0,0   0:03.88 jbd2/mmcblk0p2-
   355 root     20   0     0     0     0  S    0,3   0,0   0:09.13 flush-179:0
  1722 root     20   0 32044  660  100  S    0,3   0,3   0:06.63 php5-fpm
  2252 mysql    20   0 309m  22m  196  S    0,3  12,0   4:21.85 mysqld
  3202 root     20   0 19936  748  360  S    0,3   0,4   0:01.43 smbd
  3623 pi        20   0  8640  944  520  S    0,3   0,5   0:33.93 eggdrop
    1 root     20   0  2136   88   68  S    0,0   0,0   0:03.91 init
    2 root     20   0     0     0     0  S    0,0   0,0   0:00.04 kthreadd
    3 root     20   0     0     0     0  S    0,0   0,0   0:00.32 ksoftirqd/0
    6 root      0 -20     0     0     0  S    0,0   0,0   0:00.00 khelper
    7 root     20   0     0     0     0  S    0,0   0,0   0:00.01 kdevtmpfs
    8 root      0 -20     0     0     0  S    0,0   0,0   0:00.00 netns
    9 root     20   0     0     0     0  S    0,0   0,0   0:00.58 sync_supers

```

Рис. 2.4 Виконання команди top

Отже підключаємо та найперше потрібно перевірити чи вони є наявні в системі, для цього потрібно виконати команду “lsusb”, оскільки обидва пристрої підключені через USB. Якщо пристрої успішно підключені, потрібно внести конфігурацію мережі wlan0 у файл /etc/network/interfaces. для цього потрібно відкрити файл з допомогою команди “sudo nano /etc/network/interfaces”, та внести зміни у файл (рис 2.5).

Рис 2.5 виконання команди lsusb та внесення змін у конфігурацію

```

pi@raspberrypi ~ $ lsusb
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 0b05:1786 ASUSTek Computer, Inc. USB-N10 802.11n Network Adapter [Realtek RTL8188SU]
Bus 001 Device 005: ID 1005:a102 Apacer Technology, Inc.
pi@raspberrypi ~ $ sudo nano /etc/network/interfaces
GNU nano 2.2.6  файл: /etc/network/interfaces  Изменён

auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
auto wlan0
iface wlan0 inet dhcp
    wpa-ssid имя_сети
    wpa-psk пароль_сети
iface default inet dhcp

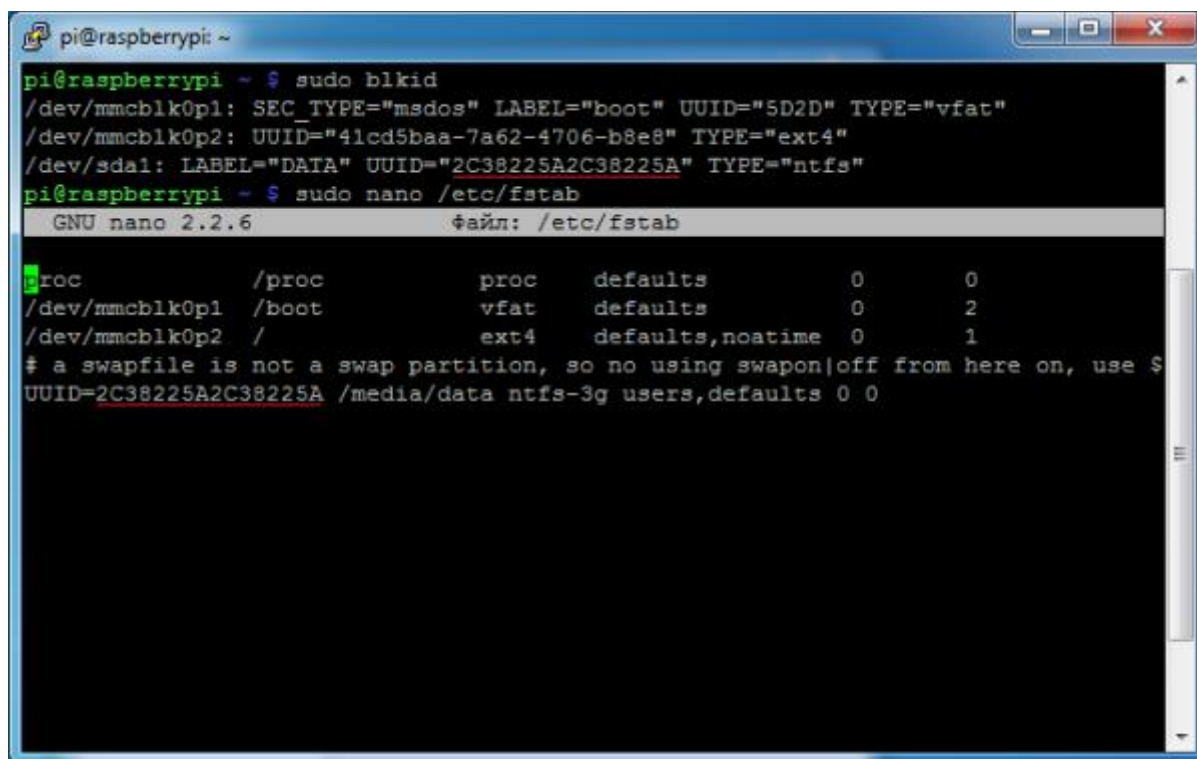
```

мережі.

Далі потрібно перезапустити мережеві служби, що можна зробити з допомогою команди “service networking restart”, та оновити систему виконавши по черзі “apt-get update” та “apt-get upgrade”.

Наступним підключемо зовнішній жорсткий диск. Для початку потрібно в каталозі “/media” створити папку “data”, куди будуть монтуватись розділи з файлами.

В “/etc/fstab” необхідно додати UUID розділу та шлях до дерективи, як показано на рис №. Далі зберігаємо файл конфігурації, монтуємо розділ “sudo mount /media/data”, для кожного наступного розділу потрібно повторити операцію заново. Перезагружаємо Raspberry та відключаємо клавіатуру з монітором, оскільки надалі будемо працювати з нею по ssh.



```

pi@raspberrypi ~ $ sudo blkid
/dev/mmcblk0p1: SEC_TYPE="msdos" LABEL="boot" UUID="5D2D" TYPE="vfat"
/dev/mmcblk0p2: UUID="41cd5baa-7a62-4706-b8e8" TYPE="ext4"
/dev/sda1: LABEL="DATA" UUID="2C38225A2C38225A" TYPE="ntfs"
pi@raspberrypi ~ $ sudo nano /etc/fstab
GNU nano 2.2.6          файл: /etc/fstab

proc                /proc              proc               defaults           0                 0
/dev/mmcblk0p1     /boot              vfat              defaults           0                 2
/dev/mmcblk0p2     /                  ext4              defaults,noatime  0                 1
# a swapfile is not a swap partition, so no using swapon|off from here on, use $
UUID=2C38225A2C38225A /media/data ntfs-3g users,defaults 0 0

```

Рис. 2.6 Пошук зовнішнього диску та конфігурування файлу /etc/fstab

Далі в налаштуваннях роутера потрібно присвоїти Raspberry Pi статичну IP-адресу і прописати параметри провайдера DDNS. На цьому

налаштування плати завершене, і її можна використовувати як сервер, та подальші дії проводити з допомогою ssh з'єднання.

2.4 Підключення до Raspberry Pi з допомогою ssh. Налаштування з'єднання та встановлення Domoticz

2.4.1 Підключення до Raspberry з допомогою ssh з'єднання. Для створення ssh підключення з ОС Windows потрібна безплатна утиліта PuTTY[14]. Для успішної роботи потрібно завантажити з сайту PuTTY такі файли:

- PuTTY.exe основний клієнт Secure Shell;
- PuTTYgen.exe генератор публічного та особистих ключів SSH;
- Pagent.exe для збереження ключів, агент аутентифікації;
- PSCP.exe захищене копіювання через консоль;
- PSFTP.exe захищене копіювання через протокол FTP.

Для коректної роботи у середовищі операційної системи Windows, потрібно встановити змінні середовища (рис 2.7):

Variable name - PATH

Variable value - C:\BIN; %PATH%

Вони потрібні для того, щоб утиліта знайшла всі системні файли, що потрібні їй для роботи.

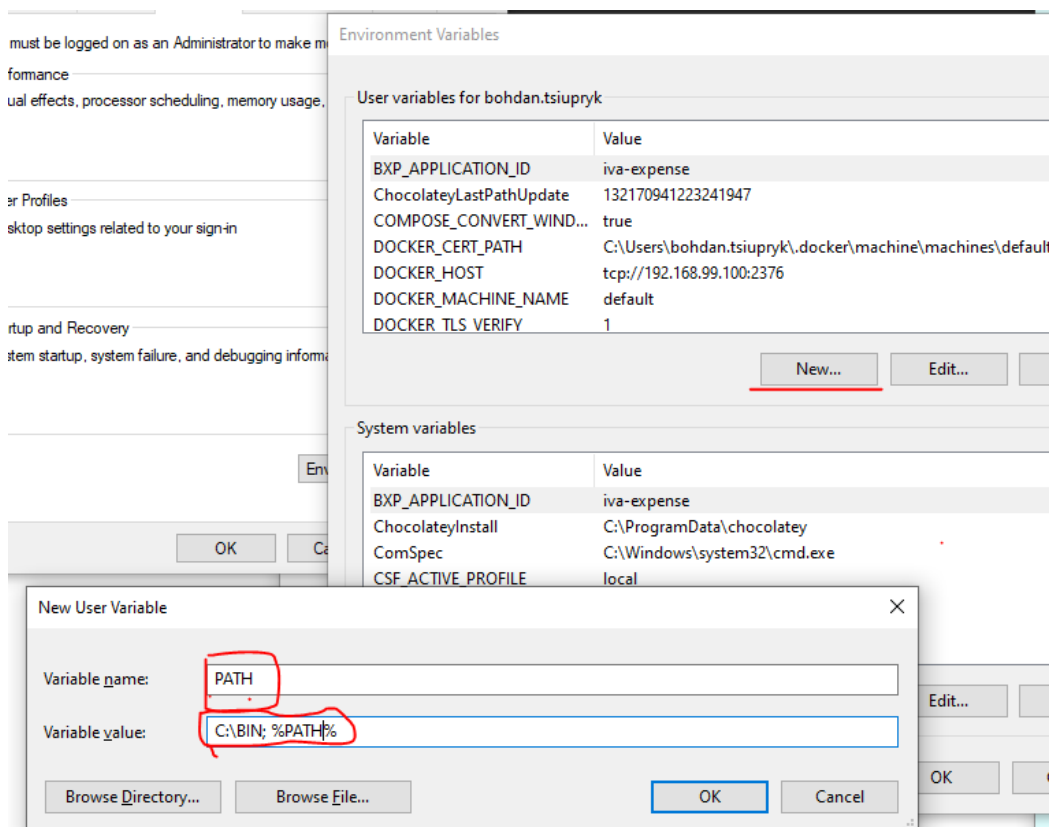


Рис. 2.7 Встановлення змінних середовища, для коректної роботи PuTTY

Далі розглянемо саму програму PuTTY, її налаштування, та процес підключення[14]. На рисунку 2.8 відображено налаштування сесії для підключення.

Розглянемо пункти налаштування підключення:

1. Host name (or IP address) - ім'я хоста до якого будемо підключатись, або його IP-адреса;
2. Port - для підключення по ssh використовується 22 порт;
3. Connection type - для використання з'єднання з допомогою security shell, потрібно обрати SSH;
4. Saved session - дозволяє зберігати сесії, щоб не потрібно було вводити дані кожного разу при запуску програми, для зберігання потрібно ввести назву сесії, та натиснути кнопку Save.

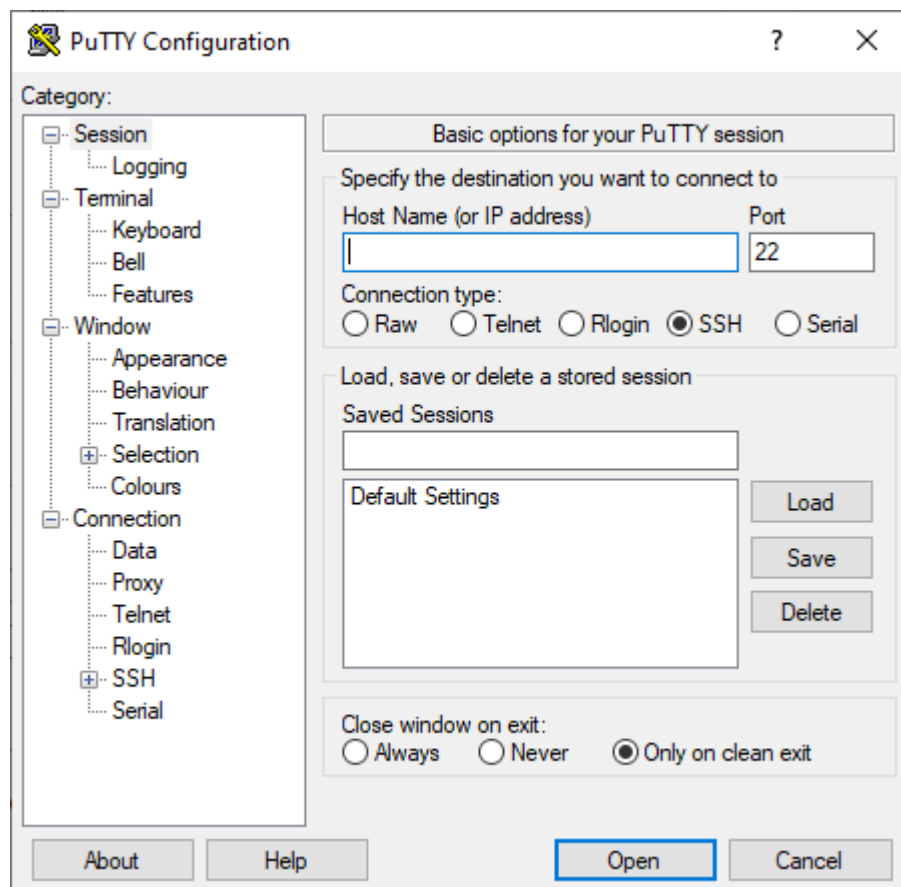


Рис. 2.8 Програма PuTTY. Вкладка session

Після введення всіх даних потрібно натиснути кнопку Open, після чого відкриється термінал, де потрібно ввести логін та пароль, які ми задавали на початку конфігурації плати Raspberry Pi. Після введення даних, буде відкрите з'єднання з платою.

2.4.2 Встановлення та конфігурація Domoticz. Для встановлення потрібно лише перейти в домашню директорію з допомогою команди “cd /home/”, далі скачати та запустити з допомогою “curl -L https://install.domoticz.com | bash”. [15]Після закінчення встановлення, потрібно перейти у браузер та перейти по шляху “IP-адреса-Raspberry-Pi:8080”, відкриється початкова сторінка domoticz (рис. 2.9). Оскільки поки немає підключених пристроїв, вона буде пустою.

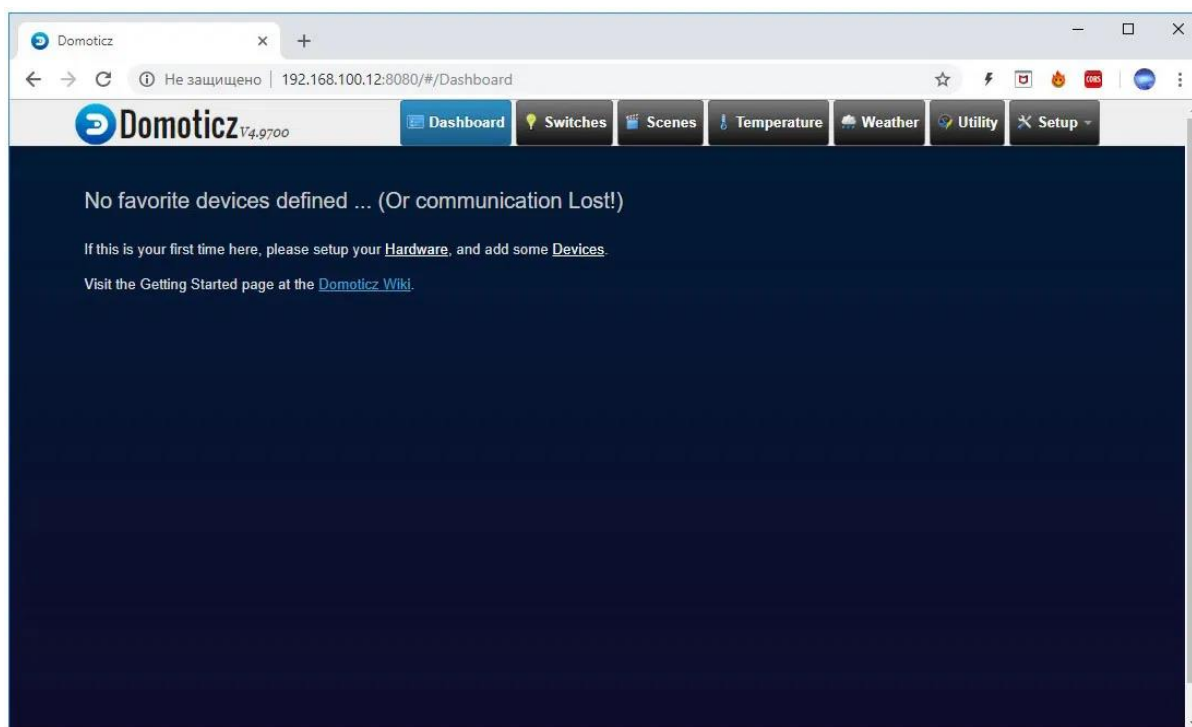


Рис. 2.9 Домашня сторінка програми Domoticz

Для налаштування потрібно перейти по шляху Setup - Setting (рис 2.10), та налаштувати такі параметри, як мова, тема оформлення, локація, захист сайту та інші. Для збереження налаштувань потрібно натиснути Apply Setting.

На цьому початкове налаштування завершене, і можна додати декілька датчиків для тестування.

Далі підключимо два датчика температури DS18B20 з допомогою інтерфейсу 1-wire, датчик температури та атмосферного тиску BPM180 та датчик освітленості TSL2561 з інтерфейсом I2C. Схема шини GPIO плати Raspberry Pi наведена у додатку Б.

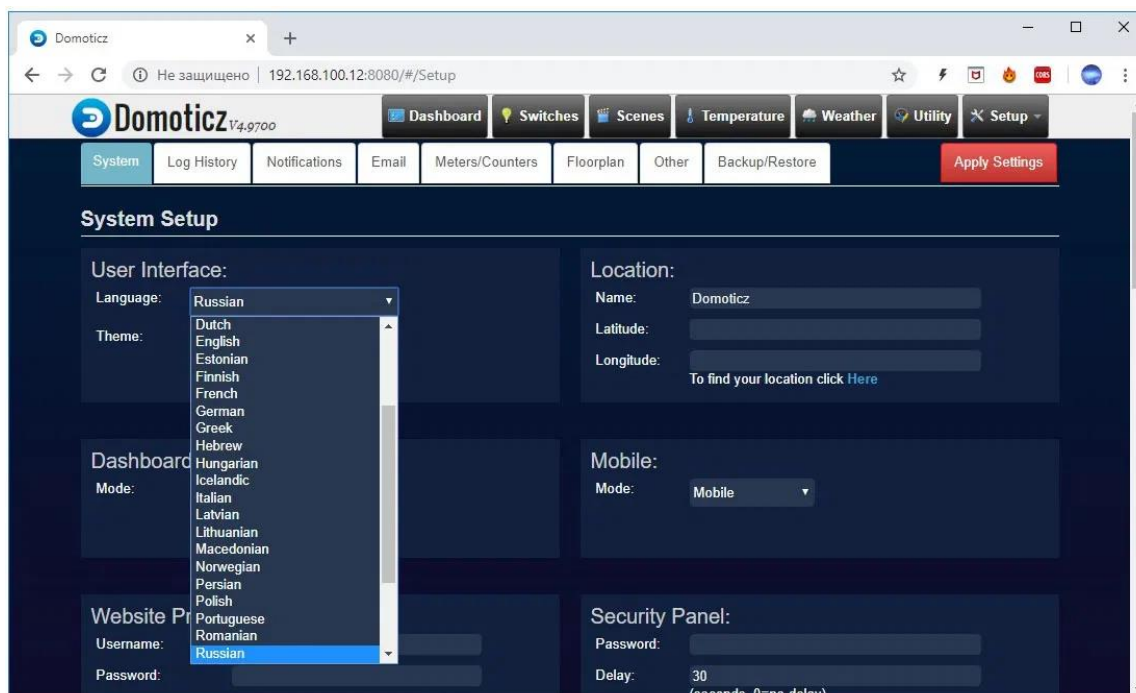


Рис. 2.10. панель налаштувань Domoticz

Підключення датчиків в програмі відбувається за допомогою веб-інтерфейсу. Для підключення потрібно перейти у вкладку Setup та обрати Hardware. у вікні що відкрилось можемо бачити список пристроїв, та вікно додавання нових (рис. 2.11). У вікні прописуємо ім'я пристрою (Name), наприклад “DS1820 #1”. Тип (Type) інтерфейсу встановлюємо “1-wire (System)”, шлях до патчу (OWFS Path) видаляємо.

інші налаштування залишаємо по-замовчуванням. Для збереження змін та додавання пристрою потрібно натиснути кнопку Add. Аналогічно підключається і конфігурується другий датчик DS18B20[18].

Далі конфігурується датчик атмосферного тиску та температури BMP180. У пункті Type обираємо інтерфейс I2C sensors і у вкладці SubType - I2C sensor BMP085/180 Temp+Baro (рис. 2.12). За тим же принципом налаштовується датчик освітлення TSL2561. Тільки для даного датчика обираємо у пункті SubType - I2C sensor TSL2561 Illuminance[19].

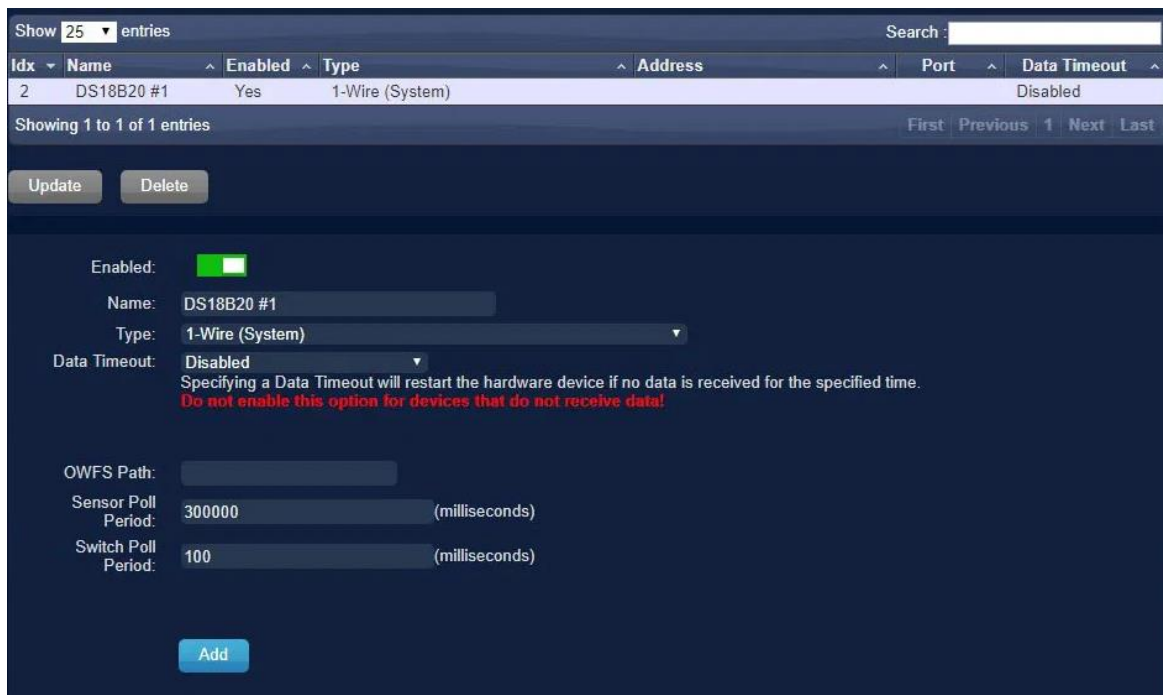


Рис. 2.11. Список підключених пристроїв та вікно додавання нового пристрою

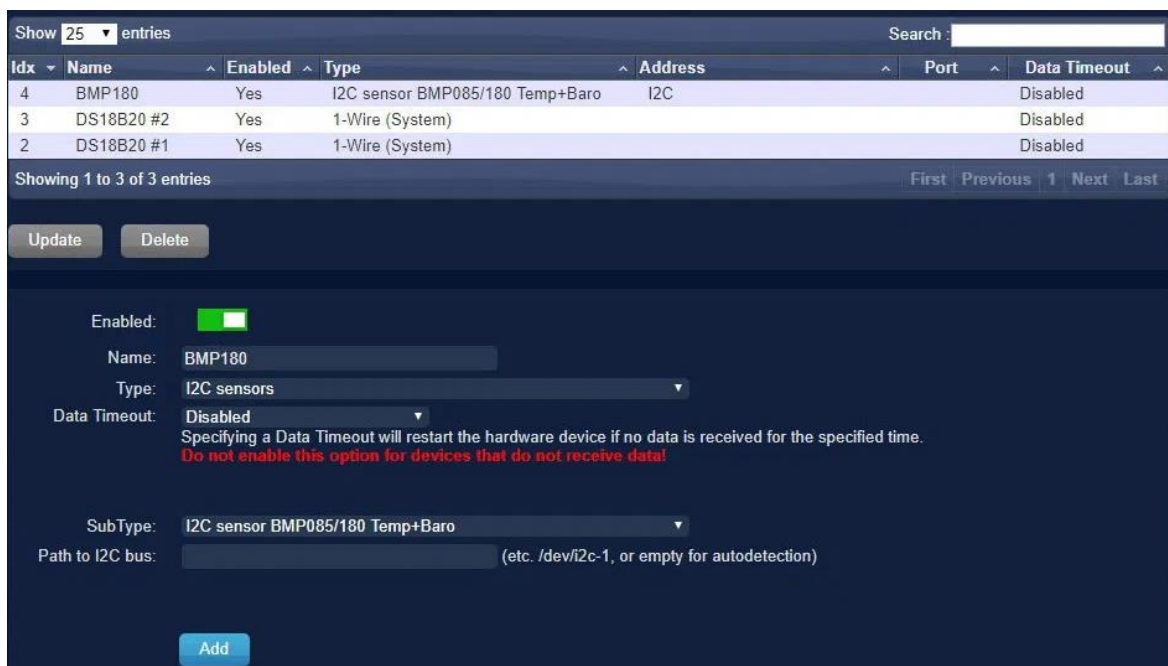


Рис. 2.12. Налаштування датчика атмосферного тиску та температури BMP180

Наразі датчики підключенні, але знаходяться у неактивному стані. Для їх активації потрібно перейти у вкладку Setup, обрати пункт Setting та активувати датчики. Для цього потрібно натиснути на зелене кільце напроти датчика, що потрібно активувати, та назвати його. Його назва буде відображатись в системі.

В системі датчики групуються по основній функціональності. Наприклад, датчики температури DS18B20 та BMP180 (температура) будуть відображатись у вкладці Temperature. А датчик BMP180 (атмосферний тиск) у вкладці Weather. Датчик освітленості TSL2561, у свою чергу, буде у вкладці Utility.

Те що датчики групуються по функціональності, дуже хороша функція при великій кількості датчиків різної функціональності. Та у всіх будуть датчики які будуть використовуватись набагато частіше. Такі датчики можна додати на головну панель - Dashboard (рис 2.13.). Для того щоб додати пристрій на головну панель, потрібно просто натиснути на символ зірки на панелі пристрою. При цьому колір змінюється на жовтий, а пристрій з'являється на панелі Dashboard.



Рис. 2.13. Панель Dashboard з усіма обраними датчиками

Також в системі Domoticz, є система логування інформації. Система дозволяє відслідковувати динаміку зміни показань датчиків за різний час. Щоб побачити графік потрібного датчика потрібно просто вибрати його, та натиснути на його іконку. Як приклад на рис 2.14. показано графік добової температури датчика DS18B20.

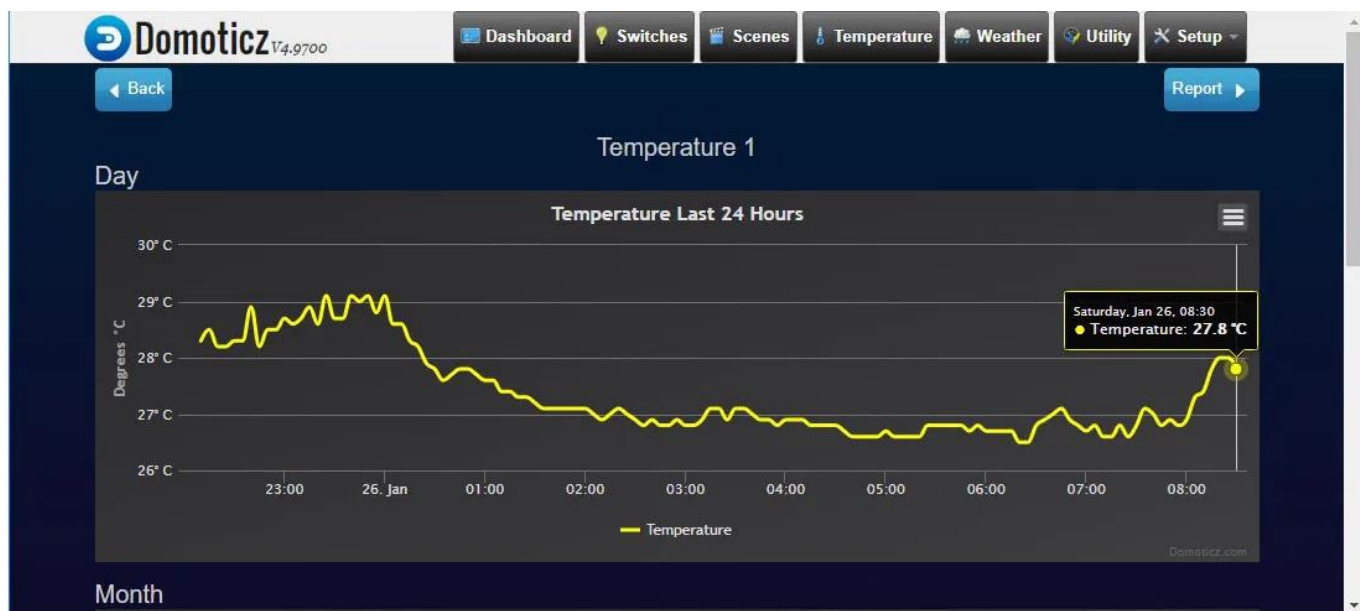


Рис. 2.14. Графік зміни температури у датчика DS18B20

2.5 Аналіз Domoticz API, як способу управління

Domoticz дозволяє всіляко взаємодіяти з усіма комутаторами та датчиками, що під'єднані до нього за допомогою JSON, або в інтерактивному режимі через браузер, або програмно з допомогою скриптів[20].

Система надає наступний формат доступу:

```
h□□□://< □□□□□□□□:□□□□□□□□@ > □□□□□□□□ - □□
<: □□□□ >/□□□□.h□□?□□□ - □□□□
```

- username:password - це логін і пароль для доступу;

- domoticz-ip - ip-адреса додатку;
- port - порт доступу, позаумовчуванням 8080;
- api-call - запит на виклик api.

Даний інтерфейс надає можливість виконання всіх методів, що є у системі. Від найпростіших, показати статус певного пристрою, або увімкнути лампу. До створення власних сценаріїв поведінки.

Приклад простого запиту для вимкнення лампи буде виглядати так:

```
/[domoticz-ip].[port]?[api-call] = [domoticz-ip]&[port] = [domoticz-ip]&[port] = 99&[domoticz-ip] = [port]
```

idx - це ідентифікатор пристрою, а switchcmd - назва команди в яку аргументами передаються параметри "on" або "off".

Якщо все пройшло успішно, у відповіді отримуємо json об'єкт з статусом та назвою пристрою (рис. 2.15).

```
{
  "status" : "OK",
  "title" : "SwitchLight"
}
```

Рис. 2.15. Приклад успішної відповіді від додатку

2.6 Висновок до розділу

У даному розділі проаналізовано одноплатний комп'ютер Raspberry Pi, що найкраще підходить для розгортання на своїй базі засобу моніторингу та управління елементами інтернету речей.

Здійснено налаштування одноплатного комп'ютера Raspberry Pi, його першочергову конфігурацію. Здійснено установку системи Domoticz на

одноплатний комп'ютер Raspberry Pi, та проаналізовано процес налаштування та досліджено можливості його користувацького програмованого інтерфейсу (API).

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАСОБІВ МОНІТОРИНГУ ТА УПРАВЛІННЯ ЕЛЕМЕНТАМИ ІОТ

3.1 Опис архітектури та інструментів розробки

Для розробки було обрано архітектуру клієнт-сервер. Під клієнтом розуміється додаток, що встановлений на одній машині, або в локальній мережі з програмою domoticz. Сервер в свою чергу може знаходитись на віддаленій машині, або на хостингу.

Така архітектура дозволяє серверу не залежати від клієнтів і бути повністю автономним. А клієнтам у свою чергу надається простий спосіб підключення до сервера, та незалежність від інших клієнтів.

Додаток розроблений на мові програмування Java версії 8 та фреймворку Spring Boot.

За зберігання даних буде відповідальна СКБД (система керування базами даних) PostgreSQL. Також для розробки та роботи з даними використовуватиметься ORM фреймворк Hibernate. У нього є власна потужна мова запитів HQL, яка є спрощеною, об'єкто-орієнтованим аналогом мови SQL. Великою перевагою є те, що Hibernate дозволяє розробнику уникнути написання великої кількості SQL-запитів (оскільки вони вже реалізовані), йому залишається тільки викликати методи, які надає фреймворк.

Веб-інтерфес розроблений з допомогою шаблонізатору Freemarker.

Середовищем розробки було обрано IntelliJ IDEA від компанії JetBrains. У нього є безплатна версія з потужними плагінами та інструментами розробки, що полегшує розробку на мові Java.

3.1.1 Архітектура серверного додатку. Додаток розроблено за архітектурним шаблоном MVC (model, view, controller). Така архітектура дозволяє слабо зв'язувати між собою модулі, що дає можливість легкої підтримки та заміни модулів[23-25].

Детальніше про модулі:

1. Model це робота з базою даних, обробка та валідація даних;
2. View, частина яка приймає вхідні дані, та надсилає оброблені;
3. Controller, основна бізнес-логіка, обробка, та виконання операції з даними.

Model інкапсулює ядро даних і основний функціонал їхньої обробки і не залежить від процесу вводу чи виводу даних.

View може мати декілька взаємопов'язаних областей, наприклад різні таблиці і поля форм, в яких відображаються дані.

У функції Controller входить відстеження визначених подій, що виникають в результаті дій користувача. Контролер дозволяє структурувати код шляхом групування пов'язаних дій в окремий клас. Наприклад у типовому MVC-проекті може бути користувацький контролер, що містить групу методів, пов'язаних з управлінням обліковим записом користувача, таких як реєстрація, авторизація, редагування профілю та зміна пароля.

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через контролер внесе зміни до моделі даних, то

інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.

3.1.2 Опис архітектури клієнтського додатку. Клієнтський додаток розроблений за монолітною архітектурою. Причиною обрання даної архітектури є те, що у даного додатку відсутня потреба підключення бази даних, та веб інтерфейсу[25].

Основне завдання додатку це підтримка з'єднання з віддаленим сервером, та трансфер повідомлень. Тому використання мікросервісної або MVC архітектур є надмірним.

3.1.3 Опис мови програмування Java та фреймворку Spring Boot. Java є об'єктно-орієнтованою мовою програмування, що випущена у 1995 році. У мови C-подібний синтаксис. Та основною особливістю мови є Java virtual machine (JVM). Ця віртуальна машина дозволяє розробляти однаковий код для різних платформ, що значно збільшує швидкість розробки, та здешевлює її вартість. JVM використовує байт-код Java, який як правило, але не завжди генерується з вихідних кодів мови програмування Java.

Основні причини обрання Java, як мови для розробки даного додатку:

1. Простий, об'єктно-орієнтований синтаксис;
2. Безпечна та безвідмовна робота з пам'яттю;
3. Незалежність від платформи та архітектури;
4. Висока продуктивність.

Spring Boot Framework - фреймворк який спрощує розробку додатків на мові Java, оскільки надає реалізацію багатьох функцій, які у "простій" Java потрібно реалізовувати самостійно. Він дозволяє легко створювати повноцінні, продуктивні Spring-додатки, про які можна сказати - що їх потрібно "просто запустити"[22]. У Spring-платформу включено і сторонні бібліотеки, що дає змогу запустити додаток з мінімальними зусиллями. Більшості Spring Boot додатків потребують невеликої конфігурації.

Можливості Spring Boot:

1. Створення повноцінних додатків, які не потребують сторонніх програм;
2. Встроєні контейнери сервлетів такі як Tomcat або Jetty;
3. Автоматична конфігурація, де це можливо;
4. Надає можливості моніторингу станів, розширену конфігурацію та метрики;
5. Конфігурація без генерації коду або написання XML файлів (на відміну від звичайної версії фреймворку Spring).

Також для розробки веб-додатку, для Spring Boot потрібно підключити залежність для роботи з веб-інтерфейсами. Завдяки тому, що фреймворк використовує засіб автоматизації роботи з проектами Maven, підключення цілої бібліотеки реалізується у декілька стрічок коду, що потрібно додати у файл з назвою pom.xml (рис. 3.1).

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-logging</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

Рис. 3.1 Підключення залежностей для web розробки для Spring Boot

3.1.4 PostgreSQL та Hibernate. PostgreSQL - це реаліційна, об'єктна система керування базами даних, вона поширюється, як система з відкритим кодом, та підтримується великою кількстю людей, оскільки не прив'язана до будь якої з компаній. База підтримує велику кількість типів даних, окрім того система надає можливість користувачам створювати свої типи, та використовувати їх. Основні можливості PostgreSQL:

1. Дотримання принципів ACID;

2. Підтримка запитів і підзапитів з JOIN, UNION і тд;
3. Послідовності;
4. Реплікація;
5. Контроль цілісності;
6. Рекурсивні запити й загальні табличні вирази.

Hibernate у свою чергу є засобом перетворення класів у Java на таблиці у СКБД, та надає зручні інтерфейси керування даними та базою. Метою Hibernate є скорочення часу розробки завдяки вже реалізованим запитам. Він сам контролює зв'язок класів і відповідних таблиць, та надає простий спосіб їх з'єднання.

Також у Hibernate є своя мова запитів HQL, що є аналогом SQL, але сильно спрощена для роботи з об'єктами, а не з таблицями.

3.1.4 Шаблонізатор Freemarker. Простий шаблонізатор, що дозволяє без знання мов програмування для веб інтерфейсів, таких як JavaScript, або TypeScript, розробляти гнучкий та зручний інтерфейс користувача. Основною особливістю є макроси, які дозволяють зменшити кількість написання коду, завдяки перевикористання вже написаного раніше.

3.2 Реалізація системи автоматизації збору даних

3.2.1 Реалізація моделей та бази даних. У системі реалізовані такі моделі даних:

- Device, модель яка представляє собою пристрій, що підключений до системи;
- DeviceDTO, модель пристрою призначена для серіалізації та передачі, яка не містить зайвих даних;
- DeviceState, модель стану пристрою;
- User, модель користувача системи;
- UserRole, модель ролі користувача;

- Diagram, модель графіка даних;
- Client, модель клієнтської частини системи;
- Command, модель команди, що може передаватись, та зберігатись для історії пристрою.

Керування моделями відбувається за допомогою репозиторіїв “Repository”, інтерфейси фреймворку Hibernate, імплементація інтерфейсів продемонстрована на рис. 3.2.

Така імплементація дозволяє уникнути написання коду, що повторюється, адже у іншому випадку для кожної моделі потрібно реалізовувати методи доступу до БД[24].

```
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface DeviceRepo extends JpaRepository<Device, Long> {  
}  
|
```

Рис. 3.2. Приклад наслідування інтерфейсів, для уникнення великої кількості коду при доступі до БД

Інтерфейси JpaRepository та CrudRepository надають лише базові методи для доступу до БД, такі як, знайти всі об’єкти, знайти об’єкти по ідентифікатору, видалити об’єкти по ідентифікатору та оновити інформацію про певний об’єкт. Якщо потрібна ширша функціональність, у JPA (Java Persistence API) закладений механізм який дозволяє додавати методи для управління БД, з допомогою додавання методів з спеціальною назвою.

Наприклад, потрібно знайти пристрої які знаходяться у статусі OFF, базовий інтерфейс надає можливість пошуку лише по ідентифікатору. Для цього потрібно правильно сформулювати назву методу. Якщо це пошук, то

спочатку вказується, що повинно повернутись після запиту, оскільки кількість вимкнених пристроїв може бути більша ніж 1, то ми очікуємо масив або список пристроїв “List<Device>” далі потрібно вказати, що це пошук, з допомогою ключового слова “find”, потім вказується, що потрібні всі записи “All” та що ми шукаємо по параметру “ByStatus”, і в параметрах передаємо статус (рис. 3.3).

За цим ж принципом реалізуються інтерфейси для доступу для інших моделей даних.

```
import org.springframework.data.jpa.repository.JpaRepository;

import java.util.List;

public interface DeviceRepo extends JpaRepository<Device, Long> {
    List<Device> findAllByStatus(DeviceStatus status);
}
```

Рис. 3.3. Приклад сформованого запиту на пошук пристрою по статусу

3.2.2 Реалізація бізнес-логіки додатку. Вся бізнес-логіка розміщена у пакеті service. Основна ідея додатку, це збір даних з декількох серверів, на яких встановлений Domoticz, їх систематизація в одному місці, збір статистики та можливий контроль.

У системі Domoticz є API, яке дозволяє через запити у форматі JSON, запитувати певні дані, або надсилати команди. Для відправки запитів буде використовуватись вбудований засіб Spring Boot під назвою RestTemplate. Він дозволяє без явних перетворень, надсилати дані. Клас має назву HttpUtil, та являє собою сервіс для відправки запитів. Клас наслідується від абстрактного класу ClosableHttpUtil (рис 3.4).

```

public static HttpClientResponse executeRequest(HttpUriRequest request, CloseableHttpClient closeableHttpClient) {
    String jsonText = null;
    int statusCode;

    try (CloseableHttpResponse response = closeableHttpClient.execute(request)) {
        statusCode = response.getStatusLine().getStatusCode();
        Log.debug("Status code: {}", statusCode);
        HttpEntity entity = response.getEntity();
        if (entity != null) {
            jsonText = extractJsonStringFromHttpResponse(entity);
            EntityUtils.consume(entity);
        }
    }
}

```

Рис. 3.4. Приклад методу для надсилання запитів

Основні сервіси:

- HttpUtil, сервіс для відправки запитів;
- DeviceService, керування пристроями;
- ComandService, для створення, та обробки команд;
- UserService, керування користувачами, їх реєстрація, авторизація і тд;
- DiagramService, обрахування та перетворення діаграм у дані, які підходять для відображення;
- MailService, розсилка листів з статистикою та іншими даними про роботу програми;
- StatisticService, збір та обробка статистики про пристрої та роботу додатку;
- InitializationService, ініціалізує з'єднання з серверами, та проводить первинне налаштування системи

3.2.3 Реалізація інтерфейсу користувача та обробки інформації, що приходить від користувачів. Користувацький інтерфейс розроблений з допомогою шаблонізатора Freemarker. Великою перевагою якого є макроси. Макроси дозволяють використовувати повторно вже написаний код, змінюючи невеликі його частини.

Наприклад, відображення діаграм. З їх допомогою, код відображення діаграм, написаний лише раз, а дані можна динамічно підставляти.

Для відображення діаграм було використано інструмент від компанії Google, Google chart. Це легковісний скрипт, який доволі легко використовувати. Все що потрібно, це підключити бібліотеку, вказати тип діаграми та дані для неї (рис 3.5).

Оформлення здійснено за допомогою бібліотеки Bootstrap 4. її перевагами є легкість в освоєні, та легковісність.

Обробку даних, що надсилаються з інтерфейсу користувача проводять контроллери. У них описується шлях, по якому вони будуть очікувати на дані, а після їх отримання обробляються та передають сервісу для подальших дій над даними.

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
  google.charts.load('current', {'packages':['corechart']});
  google.charts.setOnLoadCallback(drawChart);

  function drawChart() {
    var data = google.visualization.arrayToDataTable([
      ['Year', 'Sales', 'Expenses'],
      ['2004', 1000, 400],
      ['2005', 1170, 460],
      ['2006', 660, 1120],
      ['2007', 1030, 540]
    ]);
```

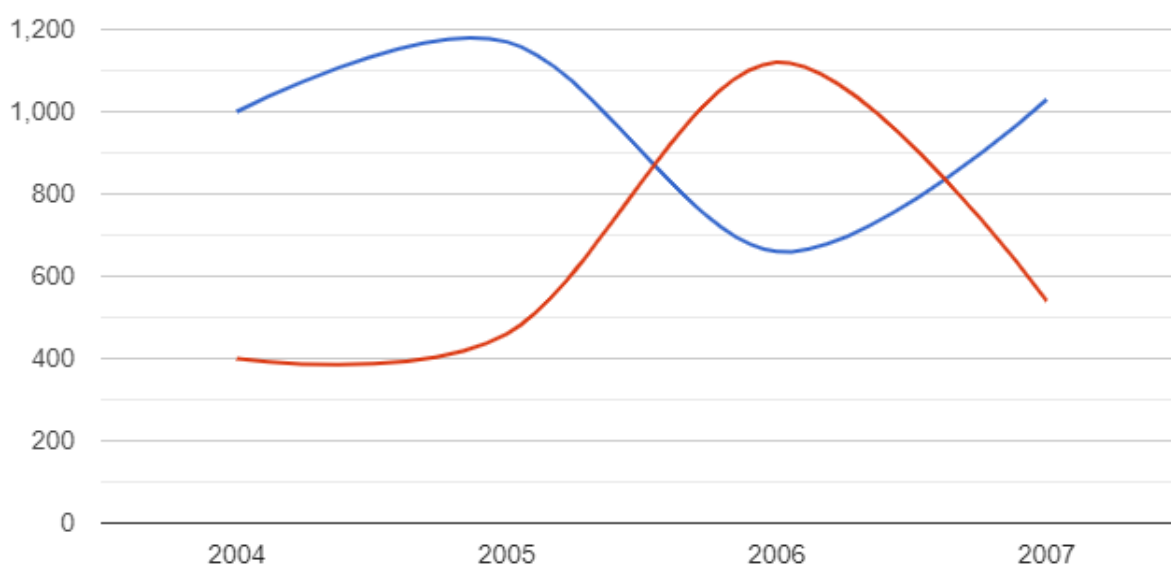


Рис. 3.5. Приклад коду діаграм, та їх вигляд після відображення

Оголошення контролера відбувається за допомогою анотації `@Controller` або `@RestController` (рис 3.6). А шлях який буде обробляти контролер задається з допомогою анотації `@RequestMapping(path)`. Причому, методи контролера, можуть мати свої шляхи, які будуть з'єднуватись з шляхом контролера. Наприклад, якщо у контролера вказаний шлях `"/device"`, а в метода додана анотація `@GetMapping("all")`, то в результаті такий метод буде обробляти шлях `"/device/all"` з HTTP методом GET.

```
@RestController
@RequiredArgsConstructor
@RequestMapping("/device")
public class DeviceController {
    private final DeviceService deviceService;

    @GetMapping("all")
    public ResponseEntity<List<Device>> getAll() {
        List<Device> all = deviceService.getAll();
        return ResponseEntity.ok(all);
    }
}
```

Рис. 3.6. Приклад оголошення шляху, який обробляє HTTP метод GET

3.3 Налаштування системи

3.3.1 Конфігурація серверної частини системи. Конфігурація доступу до бази даних проводиться з допомогою конфігураційного файлу `db.properties` (рис 3.7). Після зміни конфігурації потрібно перезапустити додаток, оскільки конфігурація проводиться при запуску додатку.

Основні пункти налаштування:

1. `datasource.url` - шлях до бази даних;
2. `datasource.username` - ім'я користувача бази даних;

3. `datasource.password` - пароль для підключення до бази даних;

```
datasource.url=jdbc:postgresql://localhost/iot-control-system
datasource.username=postgres
datasource.password=123
```

Рис. 3.7. Приклад конфігурації доступу до бази даних

3.3.2 Конфігурування клієнтської частини системи. Для налаштування потрібно у конфігураційний файл `application.properties`, встановити наступні налаштування:

1. `remote-server.ip` - IP-адреса віддаленого сервера куди будуть відправлятися дані;
2. `remote-server.port` - необов'язковий параметр, потрібно вказувати, лише якщо віддалений сервер використовує відмінний від 8080 порт;
3. `remote-server.key` - ключ безпеки, без якого запити будуть відхилятися;
4. `remote-server.local-id` - ідентифікатор клієнта;
5. `domoticz.login` - логін для підключення до `domoticz`;
6. `domoticz.password` - пароль для підключення до `domoticz`;
7. `domoticz.port` - необов'язковий параметр, потрібно вказувати, лише якщо додаток використовує відмінний від 8080 порт.

Після налаштування потрібно запустити клієнтську частину. Вона сама проведе підключення до `domoticz`, знайде всі підключенні пристрої, якщо попередні операції пройшли успішно, підключиться до віддаленого сервера.

Якщо віддалений сервер недоступний, програма увійде у чекаючий режим, і буде періодично пробувати відновити підключення.

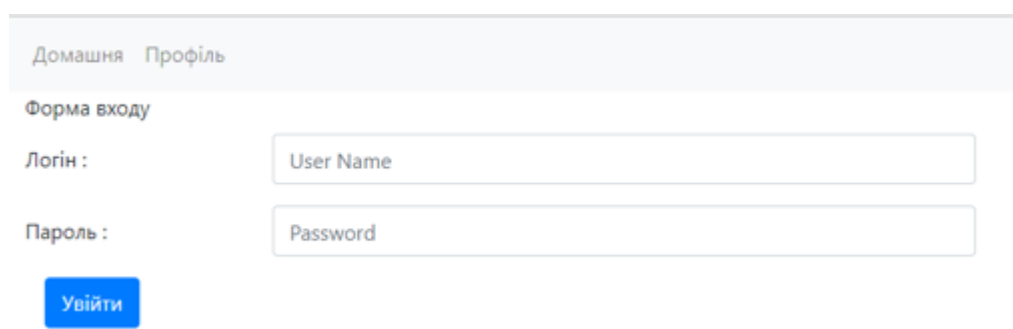
3.4 Демонстрація роботи

3.4.1 Встановлення та запуск. Для встановлення системи потрібно щоб на машині була встановлена JVM, не нижче 8-мої версії.

3.4.1.1 Клієнтський додаток. Потрібно скопіювати файл IoT-control-system-client.jar на машину з додатком Domoticz, та запустити. Запуск виконується за допомогою команди “java -jar IoT-control-system-client.jar”.

3.4.1.2 Серверний додаток. Для запуску додатка потрібна встановлена база даних з створеним користувачем. Після чого потрібно скопіювати файл IoT-control-system-server.jar, та запустити командою “java -jar IoT-control-system-server.jar”

3.4.2 Демонстрація роботи. користуватись системою можна, як з персональних комп'ютерів та ноутбуків, так і з мобільних пристроїв, основна вимога, підтримка браузера. Для початку користування потрібно перейти за адресою <https://<ip-адреса-сервера>:8080>. Після чого відкриється вікно входу в систему (рис 3.8).



Домашня Профіль

Форма входу

Логін :

Пароль :

Рис. 3.8. Форма входу в систему

Якщо це перший запуск, потрібно увійти під стандартними креденціалами “admin:admin”. Після чого змінити пароль, та додати інших користувачів за необхідності. Для цього в системі додана панель керування

користувачами (рис 3.9). На панелі можна редагувати інформацію та права користувачів, а також додавати нових.

Список користувачів					
Логін	Номер телефону	Електронна пошта	Ролі		
Bohdan	+380987278916	bod9.hom9k@gmail.com	MODERATOR, USER	Редагувати	Видалити
Ivan	+380987278916	rewq@gmail.com	MODERATOR, USER	Редагувати	Видалити

Рис. 3.9. Панель адміністратора

Для підключення додатків-клієнтів до системи, потрібно спочатку його створити на панелі створення клієнтів (рис 3.10). Де потрібно ввести назву клієнта, що буде використовуватись в системі як ідентифікатор, та згенерувати ключ безпеки.

Ім'я клієнта (також використовується як ідентифікатор)

Додати клієнта та згенерувати ключ

Ключ безпеки

Рис 3.10. Приклад додавання нового клієнта

Після додавання клієнта, потрібно сконфігурувати його, та запустити. Під час конфігурації потрібно використовувати ідентифікатор та ключ отримані під час додавання.

Після успішного підключення клієнт з'явиться у списку з статусом “активний” (рис. 3.11.). Якщо підключення ще не було, клієнт буде у статусі “новий”. Якщо підключення вже відбулось, але з деяких обставин перервалось, або пристрій вимкнений, клієнт перейде у статус “не активний”.

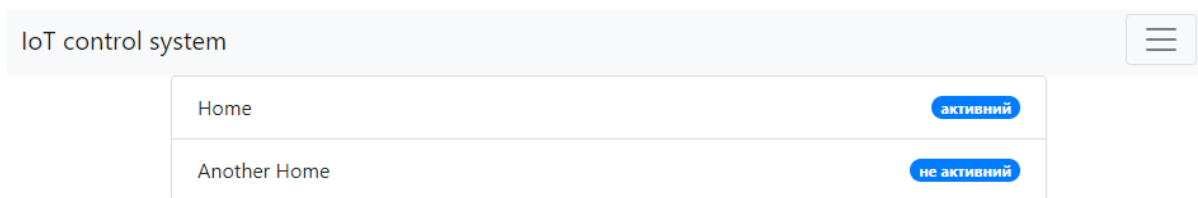


Рис. 3.11. Список клієнтів та їх статус

Для перегляду інформації про клієнта, потрібно натиснути на нього. Відкриється вікно з детальною інформацією про клієнта. Такою як, ім'я пристрою, його IP-адреса, час роботи, кількість підключених пристроїв, з можливістю переглянути інформацію про кожного з них окремо, та інша корисна інформація.

Для перегляду пристроїв, потрібно перейти на вкладку “Домашня”, тут відображені всі підключені на даний момент пристрої (рис 3.12). Їх можна сортувати за клієнтом, або типом пристрою (датчик температури, лампочка і тд).

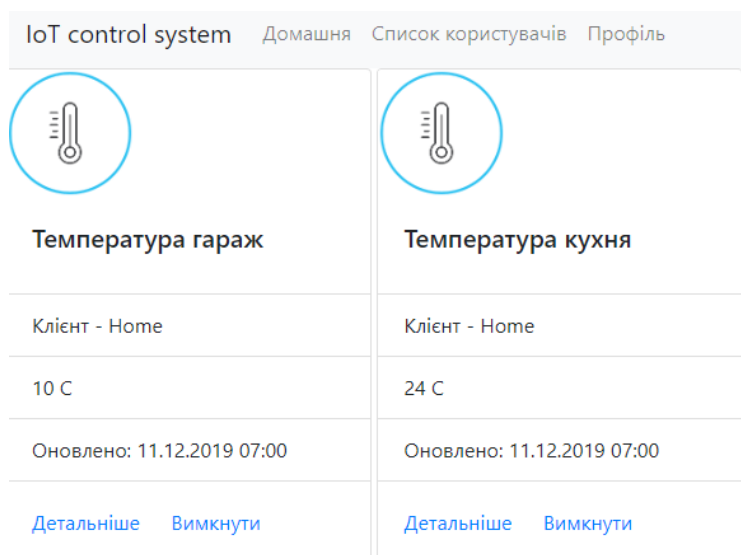


Рис. 3.12. Відображення на домашній сторінці датчиків температури

Також з домашньої сторінки можна перейти в детальніший огляд пристроїв. Та вимкнути пристрій, вимкнуті пристрої не відображаються на Домашній сторінці, і щоб їх увімкнути потрібно перейти в налаштування клієнта.

При переході за посиланням Детальніше на одному з пристроїв, відкривається вікно з детальною інформацією про пристрій. Його назва, показники, список команд які можна йому послати та якщо це наприклад датчик температури, то графік коливань температури за останню добу (рис. 3.12).



Рис. 3.12. Детальна інформація про пристрій та графік зміни температури

3.5 Висновки до розділу

У даному розділі описано інструменти, мову програмування Java та Spring Boot, з допомогою яких було розроблено систему яка працює поверх системи Domoticz та розширює її можливості. Використано незалежну архітектуру, що дозволяє швидко під'єднати ще декілька серверів-клієнтів до системи, без переписування коду системи.

Було продемонстровано принцип конфігурування клієнтів, можливість додавання нових клієнтів, їх конфігурацію та налаштування

РОЗДІЛ 4

ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

Економічне обґрунтування дипломної роботи магістра є суттю даного розділу, оскільки, дозволяє встановити доцільність проведення науководослідних робіт і економічно обґрунтувати доцільність застосування тих чи інших засобів.

Метою дипломної роботи магістра є дослідження засобів реалізації системи моніторингу та управління елементами інтернету речей.

Як відомо, розробка надійної і ефективної керуючої системи вимагає значних затрат часу. Слід зауважити, що затрати часу залежать від кваліфікації розробника і його можливостей. Розробник повинен у достатній мірі володіти навиками програмування, вміти адекватно застосовувати математичний апарат, бути добре обізнаним з об'єктом дослідження.

Розробку даної інформаційної системи можна поділити на такі етапи:

1. постановка задачі;
2. збір потрібної інформації і наступне її опрацювання;
3. прийняття рішень щодо вибору оптимального шляху розв'язання поставленої задачі;
4. аналіз математичної моделі інформаційної системи;
5. розробка алгоритму програми інформаційної системи;
6. налаштування середовища розробки і роботи вже готової програми;
7. написання програми;

8. написання і оформлення документації (електронної і паперової).

Для оцінки тривалості виконання окремих робіт використовують нормативи часу або попередній досвід. До таких нормативів відносять тривалість написання операцій (команд), які в деяких підприємствах становлять: для одної операції - 0,5-1,6 год та 8 годин для п'яти операцій (тривалість зміни).

У разі їх відсутності звертаються до експертних оцінок по встановленню тривалості кожного етапу (стадії):

при трьох оцінках:

$$T_{ec} = (t_{min} + t_{n.й} + t_{max}) / 3,$$

при двох оцінках:

$$T_{ec} = (3t_{min} + 2t_{max}) / 5,$$

де T_{ec} - очікуване (середнє) значення тривалості виконання етапу (стадії); t_{min} , $t_{n.й}$, t_{max} - відповідно мінімальна, найбільш імовірна і максимальна оцінки тривалості виконання етапу (стадії).

Для визначення загальної тривалості проведення НДР (розробки програмного продукту) доцільно дані витрат часу на виконання окремих стадій (етапів) звести у таблицю 4.1.

Витрати часу наукового керівника на виконання окремих стадій (етапів) при недостатній кількості інформації доцільно приймати в межах 5% сумарних витрат часу інженерів на виконання цих стадій (етапів).

Таблиця 4.1

Основні етапи і час їх виконання у НДР

№ з/п	Етап	Середній час виконання етапу, год	
		інженер	керівник
1	постановка задачі	3	10
2	збір потрібної інформації і наступне її опрацювання	10	5
3	прийняття рішень щодо вибору оптимального шляху розв'язання поставленої задачі	5	4
4	Встановлення та налаштування апаратної частини системи	15	10
5	розробка алгоритму програми керуючої системи	15	5
6	налаштування середовища розробки і роботи вже готової програми	2	1
7	написання програми	80	5
8	написання і оформлення документації (електронної і паперової)	20	10
разом		150	50

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

Відповідно до Закону України “Про оплату праці” заробітна плата – це “винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу”.

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та

господарської діяльності підприємства. Заробітна плата складається з основної та додаткової оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов'язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Основна з/п складається із прямої з/п і доплати, яка при укрупнених розрахунках становить 25% - 35% від прямої з/п. При розрахунку з/п кількість робочих днів в місяці слід приймати – 25,4 дні/міс., що відповідає 203,2 год./міс. Розмір місячних окладів керівника та інженерів слід приймати згідно існуючих на даний час норм. Основна заробітна плата розраховується за формулою:

$$Z_{осн} = T_c \times K_z, \quad (4.1)$$

де T_c – тарифна ставка, грн.;

K_z - кількість відпрацьованих годин.

Посадові оклади (тарифні ставки) за розрядами Єдиної тарифної сітки визначаються шляхом множення окладу (ставки) працівника 1 тарифного розряду на відповідний тарифний коефіцієнт. У разі коли посадовий оклад (тарифна ставка) визначені у гривнях з копійками, цифри до 0,5 відкидаються, від 0,5 і вище - заокруглюються до однієї гривні. У 2019 році посадові оклади (тарифні ставки) розраховуються згідно з Законом України "Про Державний бюджет України на 2019 рік".

Мінімальна зарплата в 2019 р. прирівняна до прожиткового мінімуму для працездатних осіб 4173 грн., тобто 25,13 грн/год., прийmemo 100 грн. для інженера, для керівника — 120 грн.

Тарифні ставки: керівник проекту – 180 грн./год., інженер – 100,0 грн./год.

Основна заробітна плата становитиме:

$$Z_{осн} = T_{осн} \times K_{ГОД}$$

Керівник проекту

$$Z_{осн} = 180 \times 50 = 9000 \text{ грн.}$$

Інженер

$$Z_{осн} = 100 \times 150 = 15000 \text{ грн.}$$

Додаткова заробітна плата становить 10–15 % від суми основної заробітної плати:

$$Z_{дод} = Z_{осн} \times K_{дод}, \quad (4.2)$$

Керівник проекту

$$Z_{дод} = 9000 \times 0.1 = 900 \text{ грн.}$$

Інженер

$$Z_{дод} = 15000 \times 0.1 = 1500 \text{ грн. ,}$$

де $K_{дод}$ – коефіцієнт додаткових виплат працівникам 0,1.

Звідси загальні витрати на оплату праці ($V_{оп}$) визначаються за формулою, і становлять:

$$V_{оп} = Z_{осн} + Z_{дод}, \quad (4.3)$$

Керівник проекту:

$$B_{\text{ОП}}=9000+2700=11700 \text{ грн.}$$

Інженер :

$$B_{\text{ОП}}=15000+4500=19500 \text{ грн. .}$$

Таким чином загальна сума становить 31200 грн. Крім того, слід визначити відрахування на соціальні заходи:

- фонд страхування від безробіття – 2,1%;
- пенсійний фонд - 32%;
- фонд соціального страхування - 2,9%;
- фонд соціального страхування від нещасних випадків і професійних захворювань — 1%.

У сумі зазначені відрахування становлять 38 %. Отже, сума відрахувань на соціальні заходи буде становити:

$$B_{\text{с.з.}} = \text{ФОП} \times 0,375,$$

$$B_{\text{с.з.}} = 31200 \times 0,38 = 11856 \text{ грн.},$$

де *ФОП* – фонд оплати праці, грн.

Проведені розрахунки витрат на оплату праці зведемо у наступну табл.

4.2.

Таблиця 4.2

Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	Нарах. на ФОП, грн.	Всього витрати на оплату праці, грн. 6=3+4+5
		Тариф на ставка, грн.	К-сть від-працьов. год.	Фактично нарах. з/пл., грн.			
1	2	3	4	5	6	7	8
А	Б	1	2	3	4	5	6

1.	Керівник проекту	180	50	9000	2700	4446	16146
2.	Інженер	100	150	15000	4500	7410	26910
Разом				24000	7200	11856	43056

4.3 Розрахунок витрат на електроенергію

Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_e = W \times T \times S, \quad (4.4)$$

де W – необхідна потужність, кВт;

T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії.

Згідно з постановою НКРЕ України від 24.04.2017 р. № 538 вартість електроенергії становить 224,473 коп./кВт.год.

Потужність комп'ютера – 500 Вт з підключеним маршрутизатором, кількість годин роботи обладнання згідно таблиці 4.1 – 200 годин.

$$Z_e = 0,5 \times 200 \times 2,24473 = 224,473 \text{ грн.}$$

4.4 Розрахунок витрат на матеріали

Результати розрахунку затрат на матеріали зводяться в таблицю 4.3.

Таблиця 4.3

Визначення величини затрат на матеріали

Найменування матеріальних ресурсів	Одиниця виміру	Норма витрат	Ціна за одиницю грн	Затрати матеріалів грн	Транспортнозаготівельні витрати, грн	Загальна сума витрат на матеріали, грн
Папір А4-80	Пачка	2	100	200	-	200
Ватман	Штук	8	20	160	-	160
Заправка картриджа до принтера	Штук	1	100	100	-	100
Плата за інтернет	Штук	1	200	200	-	200
Разом						660

4.5 Розрахунок суми амортизаційних відрахувань

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

Комп'ютери та оргтехніка належать до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60 % (квартальна – 15 %).

Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{B_B \cdot H_A}{100}, \quad (4.5)$$

де A – амортизаційні відрахування за звітний період, грн..

B_B – балансова вартість комп'ютера, на початок звітного періоду, грн..

H_A – норма амортизації, %.

$$A = \frac{30000 \times 15}{100} = 4500$$

4.6 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

Накладні витрати можуть становити 20% від суми основної та додаткової заробітної плати працівників:

$$H_B = V_{O.ПХ} \cdot 0.2, \quad (4.6)$$

$$H_{\epsilon} = 31200 \times 0,2 = 6240 \text{ грн.}$$

4.7 Складання кошторису витрат та визначення собівартості НДР

Результати проведених вище розрахунків зведемо у табл. 4.3. Собівартість (C_B) НДР розрахуємо за формулою:

$$C_{\epsilon} = V_{o.n.} + V_{c.z.} + Z_{m.v.} + Z_e + T_{\epsilon} + A + H_{\epsilon}, \quad (4.7)$$

$$C_{\epsilon} = 31200 + 11856 + 225 + 660 + 4500 + 6240 = 54681 \text{ грн.}$$

Таблиця 4.4

Кошторис витрат на НДР

Зміст витрат	Сума, грн.	В % до загальної суми
1	2	3
Витрати на оплату праці (основну і додаткову заробітну плату)	31200	57,06
Відрахування на соціальні заходи	11856	21,68
Матеріальні витрати	660	1,21
Витрати на електроенергію	225	0,41
Амортизаційні відрахування	4500	8,23
Накладні витрати	6240	11,41
Собівартість	54681	100

4.8 Розрахунок ціни НДР

Ціну НДР можна визначити за формулою:

$$Ц = \frac{C_v \cdot (1 + P_{рен}) + K \cdot B_{н.і.}}{K} \cdot (1 + ПДВ), \quad (4.8)$$

$P_{рен.}$ – рівень рентабельності, 30 %;

K – кількість замовлень;

$B_{н.і.}$ – вартість носія інформації, грн;

$ПДВ$ – ставка податку на додану вартість, (20%).

Оскільки розробка є прикладною, і використовуватиметься тільки для одного підприємства, то для розрахунку ціни не потрібно вказувати коефіцієнти K та $B_{н.і.}$, оскільки їх в даному випадку не потрібно.

Тоді, формула для обчислення ціни розробки буде мати вигляд:

$$Ц = C_B \cdot (1 + P_{рен}) \cdot (1 + ПДВ) \quad (4.9)$$

Таким чином ціна, згідно з формулою, рівна 85 302,36 грн.

Визначимо величину прибутку:

$$П = Ц - C_B, \quad (4.10)$$

Згідно даної формули отримаємо 30639 грн.

4.9 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Економічна ефективність (E_p) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_p = П / C_B, \quad (4.11)$$

де $П$ – прибуток;

C_B – собівартість.

$$E_p = 30639 / 54681 = 0,56.$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_p):

$$T_p = \frac{1}{E_p}, \quad (4.12)$$

$$T_p = 1 / 0,56 = 1,78 \text{ р.}$$

Про доцільність розробки програми можна сказати при врахуванні наступних критеріїв:

Таблиця 4.5

Техніко-економічні показники НДР

№ п/п	Показник	Значення
1.	Собівартість, грн.	54681
2.	Плановий прибуток, грн.	30639
3.	Ціна, грн.	85320
4.	Економічна ефективність	0,56
5.	Термін окупності, рік	1,78 року

4.10 Висновки до розділу

У результаті проведення розрахунків можна зробити висновок: розробка матиме оптимальну економічну ефективність 0,56 і термін окупності становитиме рік і 9 місяці (1,78 року). Варто зазначити, що дані розрахунки носять номінальний характер і основна їх мета оцінити приблизну вартість дослідження та створення даного продукту. Номінальний характер розрахунків зумовлений тим, що даний програмний продукт має дослідницьке призначення.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ ТА БЕЗПЕКИ В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1 Охорона праці

У роботі досліджуються засоби моніторингу та управління елементами інтернету речей, які в свою чергу використовують одноплатні комп'ютери та мережеве обладнання, зокрема: маршрутизатори, комутатори, повторювачі, сервери, тому при роботі з даною технікою є необхідним дотримання правил охорони праці, техніки безпеки та протипожежної безпеки.

Згідно з законом України, нормативно-правові акти з охорони праці є обов'язковими для виконання у виробничих майстернях, лабораторіях, цехах, на ділянках та в інших місцях трудового і професійного навчання, облаштованих у будь-яких навчальних закладах[31].

Всі пристрої мережевої взаємодії потребують живлення від електромереж, при цьому електромагнітні поля, які створюються цими пристроями, особливо негативно впливають на організм людини, яка безпосередньо працює з джерелом випромінювання. При дії випромінювання на людину можливі гострі та хронічні форми порушення фізіологічних функцій організму. Такі порушення виникають в результаті дії електричної складової випромінювання на нервову систему, а також на структуру кори головного та спинного мозку, серцево-судинної системи.

Приміщення, в яких планується установка та подальша робота з комп'ютеризованими приладами, за допомогою яких буде будуватися віртуальний макет комп'ютерної мережі, повинні відповідати проектній документації будинку, погоджений з уповноваженими державними органами. Крім того, повинні бути враховані санітарні нормативи

освітлення, вимоги до параметрів мікроклімату (температура, відносна вологість), ступеня і сили вібрації, звукового шуму і вогнестійкості приміщення. Конкретні показники зазначених санітарних норм знаходяться в Державних санітарних правилах і нормах роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПН 3.3.2.007-98.

Відповідне приміщення повинно бути укомплектоване системами центрального або індивідуального опалення, кондиціонування чи вентиляції повітря. При установці зазначених систем, необхідно переконатись, що батареї опалення, водопровідні труби, вентиляційні кабелі тощо, надійно сховані під захисними щитками, які перешкоджатимуть можливому потраплянню робітника під напругу[34].

У процесі роботи з модельованою комп'ютерною мережею, необхідно дотримувати правильний режим праці та відпочинку. В іншому випадку у робочого персоналу значно збільшується напруга зорового апарату, з'являються скарги на незадоволеність роботою, головні болі, дратівливість, порушення сну, втому і хворобливі відчуття в очах, попереку, в області шиї і руках.

Робочі місця слід розташовувати так, щоб уникнути попадання в очі прямого світла. Джерела освітлення рекомендується розташовувати з обох боків екрану паралельно напрямку погляду. Для уникнення світлових відблисків екрану, клавіатури в напрямку очей користувача, від світильників загального освітлення або сонячних променів, необхідно використовувати антиполюсківі сітки, спеціальні фільтри для екранів, захисні козирки, на вікнах - жалюзі.

Екран дисплея повинен бути розташованим перпендикулярно до напрямку погляду. Якщо він розташований під кутом, то стає причиною сутулості. Відстань від дисплея до очей повинна трохи перевищувати звичну відстань між книгою та очима.

Монітор повинен бути розташований на робочому місці так, щоб поверхня екрана знаходилась в центрі поля зору на відстані 400-700 мм від очей користувача.

Залежно від джерела світла виробниче освітлення може бути: природним, що створюється прямими сонячними променями та розсіяним світлом небосхилу; штучним, що створюється електричними джерелами світла, та суміщеним, при якому недостатнє за нормами природне освітлення доповнюється штучним. Природне освітлення поділяється на: бокове (одно - або двостороннє), що здійснюється через світлові отвори (вікна) в зовнішніх стінах; верхнє - через ліхтарі та отвори в дахах і перекриттях; комбіноване - поєднання верхнього та бокового освітлення. Штучне освітлення може бути загальним та комбінованим[34].

Окрім вищезазначеного, важливою складовою охорони праці є пожежна безпека приміщень. Основними напрямками забезпечення пожежної безпеки є усунення умов виникнення пожежі та мінімізація її наслідків. Об'єкти повинні мати системи пожежної безпеки, спрямовані на запобігання пожежі, дії на людей та матеріальні цінності небезпечних факторів пожежі, в тому числі їх вторинних проявів. До таких факторів, згідно з ГОСТ 12.1.004-91, належать:

- полум'я та іскри;
- підвищена температура навколишнього середовища;
- токсичні продукти горіння й термічного розкладу матеріалів і речовин дим;
- знижена концентрація кисню;

Відповідно до ГОСТ 12.1.004-91, пожежна безпека об'єкта повинна забезпечуватися системою запобігання пожежі, системою протипожежного захисту і системою організаційно-технічних заходів.

Основними вихідними даними при розробці комплексу технічних і організаційних рішень щодо забезпечення потрібного рівня пожежної

безпеки в кожному конкретному випадку є чинна законодавча і нормативно-технічна база з питань пожежної безпеки, властивості матеріалів і речовин, що застосовуються у виробничому циклі, матеріалів, речовин і особливості виробництва.

Під час виконання роботи, був проведений аналіз діючих законодавчих актів та норм щодо облаштування робочих приміщень. Проведений опис параметрів, особливостей розміщення джерел світла та заходів протипожежної безпеки у приміщеннях, де проводиться робота над проектуванням макету для дослідження методів та засобів резервування та агрегації каналів комп'ютерних мереж.

5.2 Основи фізіологічної праці й комфортних умов життєдіяльності. Класифікація основних форм діяльності людини. Особливості фізичної та розумової праці.

5.2.1 Основи фізіологічної праці й комфортних умов життєдіяльності. Фізіологія праці – це галузь фізіології, що вивчає зміни стану організму людини в процесі різних форм трудової діяльності та розробляє найбільш сприятливі режими праці і відпочинку[34]. Праця людини є функціональним процесом, у якому використовуються фізіологічні та психологічні якості працівника. В процес праці залучаються всі органи й системи організму людини – мозок, м'язи, судини, серце, легені та ін. При цьому витрачається нервова та м'язова енергія. Отже, праця - це фізіологічний процес витрачання людської енергії. Крім того, в процесі праці активізуються усі психічні функції людини: сприймання, мислення, пам'ять, відчуття, уява, вольові якості, уважність, зацікавленість, задоволення, зосередженість, напруження, стомлення тощо.

У процесі праці людина сприймає і переробляє інформацію, реалізує комунікативну функцію психіки, проявляє індивідуальні властивості

поведінки у тих чи інших небезпечних ситуаціях. Тобто під час праці відбуваються два види навантажень людини: м'язові і нервові.

Залежно від співвідношення м'язових і нервових навантажень праця умовно поділяється на фізичну, з перевагою м'язових навантажень, розумову, з перевагою навантажень на кору головного мозку, пов'язаних із вищими психічними функціями. Зміст праці характеризують співвідношення затрат м'язової та нервової енергії, виконавських і творчих функцій, механічних дій і операцій мислення у трудовому процесі.

Умови, в яких працює людина, впливають на результати виробництва це продуктивність праці, якість і собівартість продукції, що випускається. Продуктивність праці підвищується за рахунок збереження здоров'я людини, підвищення рівня використання робочого часу, продовження періоду активної трудової діяльності людини.

Одним з необхідних умов здорової і високопродуктивної праці є забезпечення чистоти повітря і нормальних метеорологічних умов в робочій зоні приміщень, тобто просторі заввишки до 2 метрів над рівнем підлоги або майданчика, де знаходяться робочі місця[37].

Комфортними умовами вважаються:

- температура повітря на робочому місці, С:
 - у приміщенні в теплий період 18-22;
 - у приміщенні в холодний період 20-22;
 - на відкритому повітрі в теплий період 18-22;
 - на відкритому повітрі в холодний період 7-10;
- відносна вологість повітря 40-54%;
- швидкість руху повітря менше 0,2м\с;
- токсичні речовини (кратність перевищення ГДК) менше 0,8;
- промислова пил (кратність перевищення ПКД) менше 0,8;
- освітленість, кратність перевищення або зменшення 1,3-1,5 норми;
- шум, рівень звуку дБ менше 68.

5.2.2 Класифікація основних форм діяльності людини. Діяльність — це специфічний спосіб ставлення людини до світу. Вона поєднує біологічну, соціальну та духовно-культурну сутність людини. Діяльність постає як засіб перетворення природи на предмети споживання, творіння культури.

Характерні ознаки діяльності людини:

- вона діє під впливом тих чи інших мотивів для задоволення певної потреби;
- вона існує завдяки взаємодії з навколишнім середовищем (інші люди, предмети; природа тощо);
- обмінюється інформацією з іншими людьми, тобто бере участь у спілкуванні;
- з самого початку життя людина грається, вчиться, а далі працює;
- саме завдяки діям, взаємодіям набуває певного досвіду;
- відчуває вплив умов у житті як на рівні оточення (мікросередовище), так і на рівні суспільства (макросередовище);
- діяльність має ціле усвідомлений і цілеспрямований характер.

Розрізняють фізичну та розумову діяльність. Під час фізичної діяльності, в основному, відбуваються зміни у системах, що забезпечують м'язову діяльність, дихання, кровообіг, терморегуляцію. Під час розумової діяльності навантаження несе нервова система. Для одних видів розумової діяльності характерне підвищення емоційної напруги, для інших — одноманітність.

5.2.3 Особливості фізичної та розумової праці. За характером фізичну працю (роботу м'язів) можна поділити на два види — статичну і динамічну. Статична робота пов'язана з дією навантаження на верхні кінцівки, м'язи корпусу і ніг, при утриманні вантажу, при виконанні роботи стоячи або сидячи. При статичній роботі підвищується обмін речовин, збільшується витрата енергетичних ресурсів, але в меншій мірі, ніж при динамічній.

Особливістю такого виду праці є її виражена стомлююча дія, що зумовлена довготривалим скороченням і напруженням м'язів, відсутністю умов для кровообігу, внаслідок чого відбувається накопичення кінцевих і проміжних продуктів обміну тощо. Це дуже швидко призводить до розвитку втоми.

Динамічна робота пов'язана з переміщенням у просторі тіла або його частин. У результаті енергія, яка витрачається, перетворюється на механічну і теплову. Динамічні скорочення м'язів мають переривний характер, що сприяє повноцінному кровопостачанню і кисневому обміну, а це, в свою чергу, призводить до меншої втомлюваності.

Працездатність м'язів залежить також і від обсягу навантаження – чим більше, тим швидше втомлюються м'язи.

До розумової праці належать управління, творчість, викладання, науки, навчання. Ця праця об'єднує роботи, пов'язані з прийомом та переробкою інформації, які потребують переважного напруження уваги, сенсорного апарату, пам'яті, а також активації процесів мислення, емоційної сфери.

На відміну від фізичної, розумова діяльність супроводжується меншими витратами енергетичних запасів, але це не говорить про її легкість. Основним працюючим органом під час такого виду праці виступає мозок.

5.3 Оцінка можливої радіаційної обстановки при аварії на АЕС

Оцінка радіаційної обстановки передбачає визначення методом прогнозування або за фактичним даними (за даними розвідки) масштабів і ступеня радіоактивного забруднення місцевості і атмосфери з метою визначення їхнього впливу на життєдіяльність населення, дій формувань або обґрунтування оптимальних режимів діяльності робітників та службовців об'єктів господарської діяльності.

Попередній прогноз радіаційної обстановки здійснюється шляхом рішення формалізованих завдань, що передбачають можливі наслідки впливу аварії на населення, особовий склад формувань при всіх видах їхніх дій і оптимізації режимів роботи формувань на забрудненій місцевості, режим роботи підприємств.

При розробці прогнозу для ймовірної радіаційної обстановки, вирішують такі завдання:

- визначають зони ймовірного радіаційного забруднення, та вносять їх на карту;
- визначають час початку випадання радіаційних опадів на території об'єкту;
- визначають дози опромінення, що може витримати людина, що знаходиться на зараженій території;
- розраховується тривалість можливого перебування на зараженій території;
- проводиться розрахунок можливих санітарних витрат при аварії.

Вихідними даними для проведення такого прогнозу є: тип і потужність ядерного реактора; кількість аварійних реакторів – n ; частка викинутих радіоактивних речовин PP – h (%); координати РНО; астрономічний час аварії – $T_{ав}$; метеоумови; відстань від об'єкта до аварійного реактора – R_k (км); час початку роботи робітників та службовців об'єкта – $T_{нач}$ (годин); тривалість роботи – $T_{раб}$ (годин); коефіцієнт ослаблення дози опромінення – $K_{осл}$.

Завдання по оцінці радіаційної обстановки вирішують аналітично і графоаналітично, за допомогою таблиць або спеціальних лінійок. Для рішення цих завдань спочатку необхідно знати радіаційну обстановку, обумовлену методом прогнозування або розвідки.

Для прогнозування можливого радіоактивного зараження необхідно знати: час вибуху; потужність і вид вибуху; швидкість і напрямок середнього вітру.

На карті (схемі) у першу чергу позначається місце вибуху і проводиться лінія в напрямку середнього вітру. Потім визначаються розміри зон, які наносяться на карту (рис 5.1). Оскільки прогноз дає погрішності, то обстановка додатково уточнюється за допомогою радіаційної розвідки.

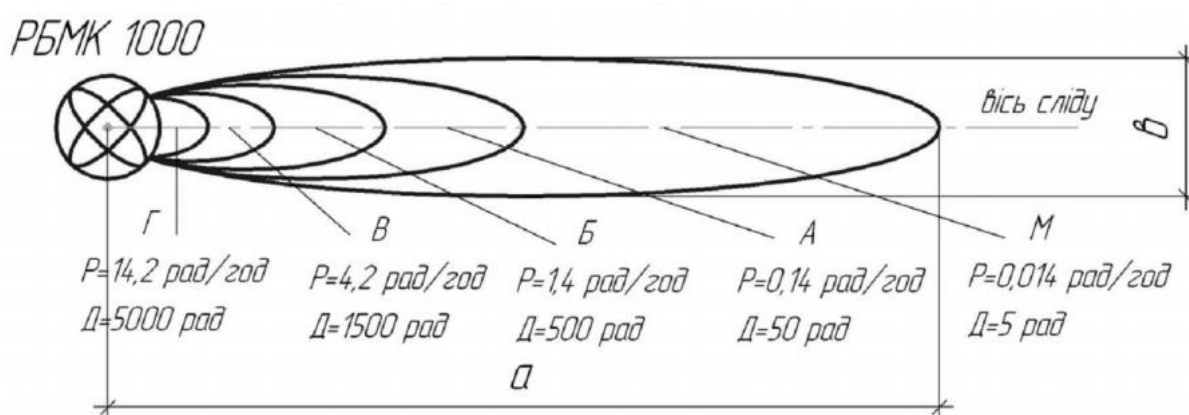


Рис. 5.1 Нанесення зон радіоактивного забруднення при аварії на АЕС

Під час ліквідації наслідків аварії незалежно від зони необхідно дотримуватися основних заходів радіаційного і дозиметричного контролю, захисту органів дихання, профілактичного прийому йодистих препаратів, санітарної обробки особового складу, дезактивації одягу і техніки.

У зоні А – помірного радіоактивного забруднення – виходячи з умов обстановки, треба намагатися скорочувати час перебування особового складу на відкритій місцевості, застосовувати засоби захисту органів дихання.

У зоні Б – сильного радіоактивного забруднення – люди повинні бути в захисних спорудах.

У зоні В – небезпечного радіоактивного забруднення – перебування людей можливе тільки в дуже захищеній техніці протягом кількох годин.

Після правильної оцінки радіаційного забруднення можна завчасно розробити засоби захисту особового складу і працюючих об'єкту, а також забезпечити проведення об'єктових навчань на кожному об'єкті народного господарства з питань цивільної оборони при загрозі виникнення надзвичайних ситуацій.

5.4 Висновки до розділу

У даному розділі було проведено аналіз діючих законодавчих актів та норм щодо облаштування робочих приміщень, було проаналізовано забезпечення нівелювання негативного впливу електромагнітного впливу на людину та апаратуру.

Також було розглянуто основи фізіологічної діяльності та комфортні умови праці. Описана класифікація основних форм діяльності людини та особливості фізичної та розумової праці. Та оглянуто методи оцінки можливого радіаційного забруднення при аварії на АЕС. Та описано умовні зони забруднення, на які поділяється місце аварії та її околиці.

РОЗДІЛ 6

ЕКОЛОГІЯ

6.1 Відчуження земель під лінії електропередач

Лінії електропередач надвисокої напруги є джерелами потужних електромагнітних полів, які існують навколо фазних проводів лінії. Ці електромагнітні поля чинять значний негативний вплив на екосистеми, що знаходяться поблизу трас ліній електропередач надвисокої напруги.

Розрізняють два типи впливу електромагнітних полів на біологічні організми – тепловий та інформаційний. Інформаційний проявляється в електромагнітних полях надвисокої частоти (300 МГц і більше) і на сьогодні вивчений недостатньо.

Для полів промислової частоти 50 Гц ліній надвисокої напруги основним є тепловий вплив від електричних струмів, індукованих в тілі біологічного організму[38].

Розрізняють наступні види впливу:

- безпосередній вплив, який проявляється при перебуванні людини в електричному полі. Ефект впливу підсилюється зі збільшенням напруженості поля і часу експозиції;
- вплив електричних зарядів (імпульсного струму), що виникають внаслідок дотику людини до ізольованих від землі конструкцій або при дотику людини, яка ізольована від землі, до рослин і різного роду заземлених конструкцій;
- вплив струму витоку.

Вплив електричного поля на організм людини визначається насамперед струмами, які протікають через тіло людини. Дослідження показали, що струми густиною менш ніж 0,1 мкА/см² зовсім не впливають на здоров'я і нервову систему людини, оскільки вони набагато менші від

імпульсних біострумів синапсів. Такому струму відповідає напруженість електричного поля на висоті 1,8 м від рівня землі (що відповідає зросту звичайної людини), яка дорівнює 15 кВ/м, яку й прийнято як припустиму під повітряними лініями надвисокої напруги.

Досліджуючи вплив електричного поля, варто пам'ятати, що безпосередньо під лінією спостерігається взаємна компенсація електричних полів, індукованих проводами сусідніх фаз, які перебувають під напругою, з сунутою за фазою на ± 120 ел. град. Саме тому найбільша напруженість електричного поля виникає під крайньою фазою електропередачі надвисокої напруги у разі горизонтального розміщення проводів у просторі й швидко зменшується з віддаленням від електропередачі (обернено пропорційно 2–3 степені відстані від осі лінії).

Вплив електричного поля на організм людини визначається не лише інтенсивністю (напруженістю) поля, але й часом експозиції. Припустимий час перебування (у хвилинах) людини в електричному полі виражається відомою екологам експонентною залежністю від напруженості електричного поля.

Не порушення верхньої межі припустимого часу забезпечує самовідновлення фізіологічного стану організму протягом доби без залишкових реакцій, функціональних або патологічних змін. Наприклад, за напруженості електричного поля 10 кВ/м допускається перебування персоналу під впливом поля не більше 180 хв. на добу, 15 кВ/м – 45 хв. на добу, 20 кВ/м – 10 хв. на добу.

З урахуванням відзначених негативних впливів електричних полів на організм людини нормативними документами передбачено наступні гранично припустимі рівні напруженості електричного поля:

- усередині житлових будинків – 0,5 кВ/м;
- на території зони житлової забудови – 1 кВ/м;
- у населеній місцевості поза житловою забудовою – 5 кВ/м;

- на ділянках перетину лінії з автомобільними шляхами – 10 кВ/м;
- у ненаселеній місцевості – 15 кВ/м;
- у важко доступній місцевості – 20 кВ/м. Дотримання даних норм реалізується на етапі проектування ліній електропередач надвисокої напруги виконанням вимог і обмежень, передбачених відповідними нормативними документами.

6.2 Етапи та техніка збору та обробки екологічної інформації

Екологічна інформація представляє собою сукупність даних про динаміку кількісних та якісних змін стану природних об'єктів довкілля, їх взаємозв'язок і закономірності розвитку. Ця сукупність даних є базою для проведення оцінки екологічного стану навколишнього середовища та прийняття обґрунтованих управлінських рішень в області екології. Накопичена екологічна інформація за багаторічний період формує банки еколого-економічних даних, які мають велике значення для створення ефективної інформаційної екологічної системи. Банки екоінформації - один з ефективних засобів пізнання законів і закономірностей екологічного стану НПС.

Інформацію про середовище та його екологічний стан можна одержати з різних джерел, до яких перш за все треба віднести:

джерела первинної інформації, які є результатами первинних досліджень через спостереження, експеримент та під час експедицій; вони становлять істотну частину фактичного матеріалу, який у сукупності з наявною системою попередніх знань і дає нове знання;

- джерела вторинної інформації, які дають зведену інформацію про стан довкілля і здоров'я людей, ступінь екологічної безпеки господарської діяльності та екологічні ситуації в окремих регіонах і на окремих об'єктах.

- джерела науково-теоретичної інформації, що відображають здобутки знань чи діяльності й викладені у формі карт, таблиць, описів чи фізичних теорій (див. список літератури); вони використовуються як будівельний матеріал - цеглинки або ж цілі блоки - для створення;

- джерела правової інформації, що дають знання про правову базу, правові основи природокористування.

Додатковим джерелом інформації є одноразові обстеження, інвентаризація викидів шкідливих речовин в атмосферу, воду і ґрунт, вибіркоче обстеження причин простоїв і неефективної роботи очисних споруджень[38].

Характер екологічної інформації. Екологічна інформація має різні аспекти і носить різний характер - синтетичний, аналітичний і оперативний характер.

Аспекти екологічної інформації. При зборі й обробці інформації варто брати до уваги наступні аспекти:

- новизну і розширення масштабів екологічної статистики;
- інерційність інформації;
- вплив фонових факторів;
- багатоетапний збір статистичних даних і нормативних параметрів.

Збір і обробка різнопланової інформації незалежно від розмірності показників, що враховуються, повинна мати закінчене еколого-економічне вираження: збиток і втрати, що нанесені народному господарству в результаті забруднення навколишнього середовища, нераціонального використання природних ресурсів і порушення рівноваги екосистем.

Екологічні дослідження вимагають систематичного дотримання чотирьох послідовних етапів:

- спостереження;

- формулювання на основі спостережень теорії про закономірність досліджуваного явища;
- перевірка теорії наступними спостереженнями і експериментами;
- спостереження за тим, чи є правдивими передбачення, оснований на цій теорії.

Існує багато методів збору інформації: польовий метод, метод безпосередніх спостережень, ландшафтно-екологічний підхід, ландшафтно-індикаційні, гідрохімічні, біохімічні, ґрунтовогазові, гідрогеологічні, радіоекологічні спостереження, геохімічні спостереження ландшафтів, дистанційні спостереження, експериментальні дослідження.

Польовий метод - один із основних методів, який проводиться в природних умовах. Його широко використовують в агрохімії, фізіології рослин, землеробстві, рослинництві, лісівництві, селекції. При цьому здійснюють фенологічні спостереження, агрофізичні, агрохімічні, мікробіологічні дослідження ґрунтів, ботанічні, фізіологічні та біохімічні дослідження рослин. Все це дає змогу виявити біоекологічні можливості виду чи сорту рослин, з'ясувати природу відмінності у врожаї та його якості тощо.

Ландшафтно-екологічний підхід дає змогу виділити екосистеми ландшафту, місцевості, урочища і, нарешті, фацій або асоціацій. Межі цих утворень і є межами біогеоценозу або екосистеми нижчого базового рівня. Вони легко картуються, описуються, досліджуються. Такий підхід дає змогу виділяти як природні, так і штучні біогеоценози, досліджувати їх генезис, прогнозувати сукцесії, здійснювати екологічний моніторинг.

Ландшафтно-індикаційні спостереження виконуються з метою виявлення характерних зовнішніх (наочних) особливостей місцевості, що дає можливість більш цілеспрямовано проводити екологічні роботи,

раціонально розташовувати мережу місць спостережень з урахуванням направленості змін рівня забруднення навколишнього середовища.

Гідрохімічні спостереження проводять з метою вивчення підземних вод, здійснюються пробо відбором з природних джерел, криниць і гідрогеологічних свердловин. В кожному конкретному випадку вони повинні обґрунтовуватись, виходячи з існуючої можливості відбору, природної захищеності водоносних горизонтів і рівня техногенних порушень дослідницької території.

Біохімічні спостереження проводяться з метою вивчення речовинного складу рослинності, насамперед її мікро компонентного складу. Однак при вивченні впливу на навколишнє середовище будь-якого специфічного забруднення, доцільно вивчення біоти саме за цим показником.

Дистанційні спостереження дозволяють одержувати інформацію про стан окремих компонентів природного середовища і його перетворення під впливом техногенезу, активності прояву екзогенних геологічних процесів тощо.

Різниця між польовим і лабораторним експериментом полягає в тому, що перший є практично неконтрольованим через безмежну кількість природних факторів, які діють на об'єкт, другий є життєво контрольований. Екологічний експеримент, одночасно як і спостереження над екосистемами, є ефективним лише в поєднанні з третім, дуже важливим, методом екології - методом моделювання.

6.3 Висновки до розділу

У першому пункті розділу розділі було розглянуто питання відчуження земель під лінії електропередач, та який вплив вони продукують на живі організми. Враховуючи негативний вплив електричних полів, було

описано, згідно нормативних документів, припустимі рівні напруженості електричного поля у різній місцевості.

У другому пункті описані етапи та техніки збору екологічної інформації, етапи її обробки, та наведено різницю між польовими та лабораторними способами проведення експерименту.

ВИСНОВКИ

У даній магістерській роботі досліджено засоби моніторингу та управління елементами IoT, з метою полегшити роботу з мережею та елементами інтернету речей. Основні результати та висновки проведеної роботи:

- досліджено сучасні засоби передачі інформації у мережі інтернету речей;
- проаналізовано протоколи та методи, що використовуються для передавання інформації в мережі інтернету речей, що дало змогу виявити позитивні та негативні якості кожного з протоколів, та обрати оптимальні для створення мережі;
- досліджено систему Domoticz, та її програмний інтерфейс, що лягли в основу розроблюваної системи моніторингу та управління елементами інтернету речей;
- на основі системи Domoticz, з допомогою мови програмування Java та фреймворку Spring Boot, розроблено систему, що дозволяє з декількох віддалених один від одного місць віддалено спостерігати та керувати елементами інтернету речей;
- продемонстровано приклад роботи системи.

Проведено економічні розрахунки, які спрямовані на визначення економічної ефективності та вартості проведення дослідження.

Здійснено опис вимог з охорони праці й техніки безпеки відповідно до нормативних документів щодо: організації робочого місця, електробезпеки, шуму та вібрації, освітленості, мікроклімату та пожежної безпеки.

Розглянуто питання екології, що стосуються магістерської роботи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Интернет вещей в научных исследованиях. 2017. URL: <https://cyberleninka.ru/article/v/internet-veschey-v-nauchnyh-issledovaniyah> (дата звернення: 01.09.2019)
2. Обзор и сравнительный анализ технологий LPWAN сетей. 2016. URL: www.sut.ru/doci/nauka/review/20164/33-48.pdf (дата звернення: 01.10.2019)
3. Z-Wave Technical Basics 2017 URL: <https://www.domotiga.nl/attachments/download/1075/Z-Wave%20Technical%20Basics-small.pdf> (дата звернення: 01.10.2019)
4. Технология NFC принципы работы и преимущества. 2016. URL: <http://www.fotokomok.ru/tehnologiya-nfc-principyu-raboty-i-preimushhestva/> (дата звернення: 01.10.2019)
5. RFID-технології та магнітні мітки. 2015. URL: <http://allta.com.ua/what-is-rfid> (дата звернення: 01.10.2019)
6. The Basics of Bluetooth Low Energy (BLE). 2010. URL: <https://www.novelbits.io/basics-bluetooth-low-energy/> (дата звернення: 01.10.2019)
7. Wi-Fi HaLow (IEEE 802.11ah) — дальнобойное беспроводное подключение с низким энергопотреблением для интернета вещей. 2016. URL: <https://www.ixbt.com/news/2016/01/05/wi-fi-halow-ieee-802-11ah.html> (дата звернення: 01.10.2019)
8. LoRa Mesh Networking with Simple Arduino-Based Modules. 2018. URL: <https://github.com/nootropicdesign/lora-mesh> (дата звернення: 15.10.2019).
9. Bryan Boyd et al. Building Real-time Mobile Solutions with MQTT and IBM MessageSight. IBM Redbooks, 2014

10. IBM Podcast: Piper, Diaz, Nipper – MQTT. 2011. URL: http://www.ibm.com/podcasts/software/websphere/connectivity/piper_diaz_nipper_mqtt_11182011.pdf.
11. Raspberry Pi Documentation. Official resource. URL: <https://www.raspberrypi.org/documentation/> (Last accessed: 14.10.2019).
12. THE OFFICIAL Raspberry Pi Beginner's Guide. URL: https://www.raspberrypi.org/magpi-issues/Beginners_Guide_v1.pdf (Last accessed: 14.10.2019).
13. Questions regarding armhf port for Raspberry Pi. URL: <https://lists.debian.org/debian-arm/2012/03/msg00002.html> (Last accessed: 14.10.2019).
14. PuTTY Documentation Page. URL: <https://www.chiark.greenend.org.uk/~sgtatham/putty/docs.html> (Last accessed: 14.10.2019).
15. Domoticz. Частина Друга. webhomepi. URL: <https://whp.home.blog/2019/02/02/domoticz-%d1%87%d0%b0%d1%81%d1%82%d1%8c-%d0%b2%d1%82%d0%be%d1%80%d0%b0%d1%8f/>
16. Domoticz. Open Source Home Automation System. URL: <https://www.domoticz.com/DomoticzManual.pdf>
17. Domoticz. Official GitHub Page. URL: <https://github.com/domoticz/domoticz>
18. DS18B20. Programmable Resolution 1-Wire Digital Thermometer. URL: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
19. BMP180 Datasheet (HTML) - Texas Instruments. URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/514264/TI1/BMP180.html>
20. Domoticz API/JSON URL's. Wiki page. URL: https://www.domoticz.com/wiki/Domoticz_API/JSON_URL%27s

21. The Java Language Environment. 1997 Sun Microsystems, Inc. URL: <https://www.oracle.com/technetwork/java/intro-141325.html>
22. Rob Harrop, Clarence Ho. Pro Spring 3. United Kingdom. 2012 944p
23. Joshua Bloch. Effective Java. Addison-Wesley Professional. 2018 412p
24. Андрій Будаї. Дизайн-патерни — просто, як двері. 2012 90p
25. Alex Berson. Client/Server Architecture (McGraw-Hill Computer Communications Series). 1996 569p
26. Christian Bauer, Gary Gregory, Gavin King. Java Persistence with Hibernate Second Edition. 2015 608p
27. MQTT Essentials. URL: <https://www.hivemq.com/mqtt-essentials/>
28. Robert C. Martin. Clean Code. 2008 464p
29. Brian Goetz, Tim Peierls, Joshua Bloch. Java Concurrency in Practice. 2006 432p
30. Eric Freeman, Elizabeth Robson. Head First Design Patterns. 2004 694p
31. Наказ Міністерства соціальної політики України «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» № 207, 2018. URL: <https://zakon.rada.gov.ua/laws/show/z0508-18> (дата звернення: 07.11.2019).
32. Закон України «Про охорону праці» № 49, 1992. URL: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення: 07.11.2019).
33. Наказ Міністерства внутрішніх справ України «Про затвердження Правил пожежної безпеки в Україні» № 1417, 2014. URL: <https://zakon.rada.gov.ua/laws/show/z0252-15> (дата звернення: 09.11.2019).
34. Мурашко В. О., Костенецький М. І., Рушак Л. В. Промислові радіаційні аварії з джерелами іонізуючого випромінювання, запобігання та порядок їх розслідування. Київ, 2013. 82 с.

35. Державний комітет ядерного регулювання України. Проект від 01.03.2008 р. Консультації щодо підвищення безпеки джерел іонізуючого випромінювання в Україні. Київ, 2008. 24 с.

36. Білявський Г.О., Бутченко Л.І., Навроцький В.М. Основи екології: Теорія і практикум: Навч. Посібник. Київ, 2002. 352 с.

37. Касьянов М. А., Васильчук М. В. Охорона праці користувачів ПК. Луганськ, видавництво СНУ ім. В. Даля. 2009. 101 с.

38. Маринець О. М., Мозговий А. М., Романовська Н. Г. Соціальна екологія: теоретичні аспекти: Методичні вказівки. Миколаїв, 2007. 28 с.

Додаток А
Тези конференцій

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

МАТЕРІАЛИ

VII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

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



11–12 грудня 2019 року

ТЕРНОПІЛЬ
2019

В. Лукашук ЗАСОБИ ДИСТАНЦІЙНОГО КОНТРОЛЮ ПАРАМЕТРІВ ВАНТАЖУ В ЛОГІСТИЧНИХ СИСТЕМАХ	128
А. Мельничук, М. Хвостівський, І. Горбовий ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ ДІАГНОСТИЧНИХ СИСТЕМ	129
К. Моха, М. Хвостівський, А. Кравчук КОМП'ЮТЕРНІ СИСТЕМИ ГЕНЕРУВАННЯ ТЕСТОВИХ СИГНАЛІВ КРОВОНОСНИХ СУДИН ТА СІТКІВКИ ОКА ЛЮДИНИ	130
В. Нестор, В. Яцишин ПРОЦЕДУРА КЛАСИФІКАЦІЇ АТРИБУТІВ ЗА ХАРАКТЕРИСТИКАМИ ЯКОСТІ КОМП'ЮТЕРНИХ СИСТЕМ	131
А. Паламар ПРОГРАМНО-АПАРАТНИЙ КОМПЛЕКС ДЛЯ ДИСТАНЦІЙНОГО МОНІТОРИНГУ СТАНУ ДЖЕРЕЛ БЕЗПЕРЕБІЙНОГО ЕЛЕКТРОЖИВЛЕННЯ	132
Н. Паляниця, В. Дорофей РОЗРОБКА ПРОГРАМНОГО ПАКЕТУ ДЛЯ РОЗМІЧУВАННЯ МЕДИЧНИХ ЗОБРАЖЕНЬ У МАШИННОМУ НАВЧАННІ	133
Л. Пуляк, С. Лупенко МЕТОДИ ОПРАЦЮВАННЯ МЕДИЧНИХ ЗОБРАЖЕНЬ В КОМП'ЮТЕРНИХ СИСТЕМАХ	135
Б. Равчак ХАРАКТЕРИСТИКА МЕТОДОЛОГІЇ JAMSTACK	136
Є. Сов'як, Є. Тиш МЕТОДИ ТА ЗАСОБИ ПОПЕРЕДНЬОГО ОПРАЦЮВАННЯ ЕКГ ДЛЯ СИСТЕМИ ТЕЛЕМОНІТОРИНГУ	137
В. Стеблик, У. Поливана МЕРЕЖЕВИЙ МОНІТОРИНГ ЯК ЗАСІБ АНАЛІЗУ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ У ЛОКАЛЬНІЙ І ГЛОБАЛЬНІЙ МЕРЕЖІ	138
Є. Тиш, О. Зима ВИБІР КРИТЕРІЇВ ЕФЕКТИВНОСТІ БЕЗПРОВІДНИХ ТЕЛЕМЕТРИЧНИХ МЕРЕЖ	139
С. Туркот НЕЙРОННІ МЕРЕЖІ В СИСТЕМАХ БІОМЕТРИЧНОЇ АУНТЕНТИФІКАЦІЇ	140
О. Цебрик МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ БЕНЗИНУ	141
Б. Цюприк, О. Ясній БЕЗПЕКА МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ	142
В. Часник, Н. Луцик ПРОЦЕС АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ МОВИ НА БАЗІ МІКРОКОНТРОЛЕРНОЇ СИСТЕМИ	143
Я. Чирський, В. Яцишин АНАЛІЗ МОДЕЛІ ЗРУЧНОСТІ ВИКОРИСТАННІ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ ЛЮДИНО-МАШИННОЇ ВЗАЄМОДІЇ	144
Х. Юркевич, А. Луцків, Н. Попович АНАЛІЗ ЕФЕКТИВНОГО ОПРАЦЮВАННЯ ВЕЛИКИХ ТЕКСТОВИХ ДАНИХ ЗАСОБАМИ ХМАРНИХ СЕРВІСІВ	145
Я. Юськів, Є. Тиш БАЗА ДАНИХ ПІДТРИМКИ ПРОЦЕСУ ОЦІНЮВАННЯ ВПЛИВУ ДЕФЕКТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА НАДІЙНІСТЬ КОМП'ЮТЕРНИХ СИСТЕМ	146

УДК 004.7

Б. Цюприк, О. Ясній

(Тернопільський національний технічний університет імені Івана Пулюя)

БЕЗПЕКА МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ

UDC 004.7

B. Tsiupryk, O. Yasniy

(Ternopil Ivan Puluj National Technical University, Ukraine)

INTERNET OF THINGS SECURITY

Інтернет речей (англ. Internet of Things, IoT) знаходиться лише на початку свого шляху, але вже розвивається з величезною швидкістю, і всі впроваджені нововведення додають серйозних проблем, що пов'язані з інформаційною безпекою.

Крім порушення конфіденційності традиційних мереж зв'язку (повтори, підслуховування, спотворення інформації і т. д.), виникають проблеми із захистом споживчої складової. Вони зумовлені:

1. відсутністю серйозного збитку;
2. відсутністю стандартів не тільки захисту, але і взаємодії;
3. відсутністю в наші дні інтересу у виробників, як першої шаблі реалізації.

Велику загрозу несе керування пристроїв за допомогою міжмашинної взаємодії. Жодну написану людиною програму не можна вважати стовідсотково точною; для неї пишуться різні патчі для виправлення помилок. Така ж доля чекає датчики в інтернет-пристроях. Із посиленням ролі даних пристроїв в житті людей буде збільшуватися загроза безпеці всіх даних, навіть найнезначніших на перший погляд. Необхідно оцінювати будь-який витік інформації, так як резюмування її складових може представляти небезпеку для життя як фізичних, так і юридичних осіб.

Компанія ESET описую три простих кроки для посилення захисту мережі інтернету речей:

- Багатофакторна аутентифікація: використовуйте апаратні токени або спеціальне програмне забезпечення для управління даними облікових записів. Двофакторна аутентифікація використовується на додаток до базової (наприклад, ім'я користувача та пароля) під час входу в систему або програму. Як правило, на попередньо визначену адресу електронної пошти або за допомогою текстового повідомлення надсилається одноразовий код. Ця комбінація може бути використана тільки для аутентифікації одного сеансу протягом обмеженого часу (наприклад, 60 секунд).

- Мережевий інтелект (network intelligence): багато пристроїв IoT здебільшого підключаються до роутера, тому пошук загроз можна здійснювати за допомогою аналізу аномалій мережевого трафіку. Різні постачальники пропонують обладнання, яке підключається до роутера та надає можливість дізнатися про підозрілі події, а також забезпечує огляд мережевої поведінки пристроїв IoT.

- Резервне копіювання. Забезпечення регулярних і надійних резервних копій систем і даних є необхідним кроком для запобігання втратам важливих даних. У разі наявності резервних копій можна відновити випадково видалений файл або дані на пошкодженому жорсткому диску, а також забезпечити безперервність роботи під час інцидентів.

Ці три базові кроки не забезпечать повного захисту світу Інтернету речей, але дозволять покращити стан безпеки корпоративної мережі, в якій використовуються пристрої IoT, та дозволять запобігти можливим фінансовим втратам.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Національна академія наук України
Університет імені П'єра і Марії Кюрі (Франція)
Маріборський університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Шяуляйська державна колегія (Литва)
Жешувський політехнічний університет ім. Лукасевича (Польща)
Білоруський національний технічний університет (Республіка Білорусь)
Міжнародний університет цивільної авіації (Марокко)
Національний університет біоресурсів і природокористування України (Україна)
Наукове товариство ім. Шевченка
ГО «Асоціація випускників Тернопільського національного технічного університету імені
Івана Пулюя»

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник

тез доповідей

Том II

VIII Міжнародної науково-технічної
конференції молодих учених та студентів

27-28 листопада 2019 року



УКРАЇНА
ТЕРНОПІЛЬ – 2019

ПЕРЕДАВАННЯ ЕКГ В СИСТЕМАХ ТЕЛЕМОНІТОРИНГУ

- | | | |
|-----|---|-----|
| 76. | С.В. Тиш, Я.І. Юськів
МОДЕЛЬ ВИЯВЛЕННЯ ВПЛИВУ ДЕФЕКТІВ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ НА НАДІЙНІСТЬ КОМП'ЮТЕРНИХ СИСТЕМ | 103 |
| 77. | Р.Б. Трембач, С. В. Кучірка
ДОСЛІДЖЕННЯ СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ ЗА
ДОПОМОГОЮ MATLAB | 105 |
| 78. | Р.М. Фудаль, М.І. Яворська
МОДЕЛЮВАННЯ РЕЖИМІВ РОБОТИ ПРИЛАДУ ДЛЯ КОНТРОЛЮ
ЯКОСТІ ДОРІЖОК ПІДШИПНИКІВ КОЧЕННЯ | 107 |
| 79. | Л.В. Хвостівська, І.Ю. Дедів, Д.В. Ісаснко
ГЕНЕРУВАННЯ РАДІОСИГНАЛІВ ДЛЯ ТЕСТУВАННЯ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ РАДІОСИСТЕМ | 108 |
| 80. | М.О. Хвостівський, І.М.Паньків, Я.С.Моргулець
СИНФАЗНИЙ МЕТОД ОЦІНЮВАННЯ ПСИХОЕМОЦІЙНОГО СТАНУ
ЛЮДИНИ ЗА ЕЛЕКТРОЕНЦЕФАЛОСИГНАЛАМИ | 110 |
| 81. | О. О.Цебрик
МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ СПЕЦІАЛІЗОВАНИХ
КОМП'ЮТЕРНИХ СИСТЕМ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ БЕНЗИНУ | 112 |
| 82. | О.П. Ясній, Б.І. Цюприк
ПРОТОКОЛ MQTTV IoT | 113 |
| 83. | В. А. Часник, Н. С. Луцик
МЕТОДИ ТА ЗАСОБИ АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ МОВИ НА
БАЗІ МІКРОКОНТРОЛЕРНОЇ СИСТЕМИ | 114 |
| 84. | Н.Я. Черкас, Ю.Ю. Замосьний, А.А. Липак
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗНАХОДЖЕННЯ ЕФЕКТИВНОГО
РОЗМІРУ ПОВІТРЯНОГО ЗАЗОРУ ЗА ДАНИМИ СЕНСОРІВ | 115 |
| 85. | І.В. Чихіра, С.Ю. Мокрійчук, Т.І. Афтанашук
АВТОМАТИЗОВАНА СИСТЕМА БЛОКУ КЕРУВАННЯ КОНТАКТОРОМ
ДЛЯ МЕРЕЖ ПОСТІЙНОГО ТА ЗМІННОГО СТРУМУ | 117 |
| 86. | І.В. Чихіра, О.О. Реміник, Т.Б. Смачило
АВТОМАТИЗОВАНА ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ОБЛІКУ
ДІЯЛЬНОСТІ АВТОРЕМОНТНОГО ПІДПРИЄМСТВА | 118 |
| 87. | Г.В. Шимчук, Р.М. Небесний
СТРАТЕГІЯ ПАРАЛЕЛІЗМУ ДЛЯ АЛГОРИТМУ MRRR | 119 |
| 88. | В.Р. Шищак, Р.М. Карабін, В.П. Кубашок, О.В. Тогосько
РОЗРОБКА ВИСОКОШВИДКІСНОЇ МАГІСТРАЛЬНОЇ ЛІНІЇ ПЕРЕДАЧІ | 121 |

УДК 004

О.П. Ясній, Dr., Prof., В.І. Цюприк

Тернопільський національний технічний університет імені Івана Пулюя, Україна

ПРОТОКОЛ MQTTB IoT

О.Р. Yasniy, Dr., Prof., В.І. Tsiupryk

MQTT PROTOCOL IN IoT

Інтернет речей стрімко розвивається, тому він потребує протоколу для обміну даними між пристроями. Ним став протокол MQTT, що працює за принципом видавець-підписник. Протокол працює на прикладному рівні та побудований на протоколі TCP/IP.

Система зв'язку, побудована на основі даного протоколу, складається з сервера-видавця, сервера-брокера та одного або декілька клієнтів. Клієнтам-підписникам не потрібна жодна інформація на конкретного видавця. В системі може бути декілька брокерів, які поширюють повідомлення.[1]

Розглянемо структуру повідомлення, що передається через протокол MQTT. Перший байт містить тип повідомлення (запит клієнта на підключення, підтвердження підпису, запит ping), прапорець дублювання, інструкцію для збереження повідомлення і інформацію про рівень якості обслуговування QoS. Другий байт містить інформацію про довжину повідомлення, деяку корисну інформацію і необов'язкові заголовки. Прапорець QoS дозволяє організувати змінну функціональність протоколу.[1,2]

Протокол MQTT – легкий протокол обміну повідомленнями, побудований для надійних мереж і пристроїв з обмеженим доступом до живлення і обчислювальними можливостями. Розглянемо детальніше, де можна застосовувати різні рівні QoS у пристроях IoT. Датчики навколишнього середовища знаходяться зазвичай у на відстані та регулярно передають дані з заданим інтервалом. Протокол MQTT підтримує модель доставки не більше одного разу. У мережах з частковим покриттям території це означає, що дані можуть втрачатись або дублюватись. Датчики, що знаходяться у на віддалі зазвичай малопотужні, тому даний протокол ідеально підходить для сенсорів IoT з відносно низьким пріоритетом передачі даних. З іншої боку, білінгові системи потребують пріоритетних та точних даних, котрих потрібно вчасно опрацьовувати. У бізнес ситуаціях, де дублювання або втрата даних неприпустимі, стане у нагоді прапорець QoS «точно один раз», що не допустить дублювання або втрати даних і зменшить кількість аномалій при обчисленнях.[1]

Таким чином, протокол ідеально підходить для передачі даних у мережі пристроїв IoT, завдяки своїй енергоефективності та надійності. Однак у нього є й один недолік: він працює над протоколом TCP/IP і не може застосовуватися у пристроях, які працюють на транспортному або нижчих рівнях моделі OSI

Література

1. Bryan Boyd et al. Building Real-time Mobile Solutions with MQTT and IBM MessageSight. IBM Redbooks, 2014
2. IBM Podcast: Piper, Diaz, Nipper – MQTT. [Електронний ресурс]. Режим доступу: http://www.ibm.com/podcasts/software/websphere/connectivity/piper_diaz_nipper_mqtt_11182011.pdf.

Додаток Б

Схема шини GPIO плати Raspberry Pi

