



Д. В. Гончаренко, В. Б. Мокін

Вінницький національний технічний університет, м. Вінниця, Україна

МЕТОД АВТОМАТИЗОВАНОГО ГЕНЕРУВАННЯ PFD-ДІАГРАМ ДЛЯ ПРОЄКТУВАННЯ ІОТ-СИСТЕМ НА БАЗІ МЕРЕЖІ SIGFOX

Запропоновано нові підходи до автоматизації процесу проєктування інформаційних систем на підставі технологій Інтернету речей (IoT) з урахуванням специфіки енергоефективних мереж типу Sigfox. Встановлено, що ефективне врахування обмежень таких мереж потребує розроблення спеціалізованих методів розроблення технологічних схем. З'ясовано, що застосування великих мовних моделей LLM (англ. *Large Language Model*) дає змогу забезпечити належний рівень автоматизації процесів побудови PFD-діаграм (англ. *Process Flow Diagrams*) на підставі текстових описів, які вводить користувач. Охарактеризовано метод контекстного аналізу, який дає змогу виділяти ключові компоненти IoT-системи, міжкомпонентні зв'язки та потоки даних із текстової інформації. Обґрунтовано доцільність переходу від традиційного ручного моделювання до інтелектуального синтезу схем за допомогою моделей LLM, що істотно скорочує тривалість розроблення IoT-систем та підвищує точність отриманих результатів. Розроблено метод автоматизованого генерування PFD-діаграм, що містить: лінгвістичний аналіз і структурування опису системи, процедуру генерування коду з оптимізацією потоків даних під обмеження LPWAN-мереж, автоматизовану візуалізацію потоків даних у вигляді PFD-діаграми. Оцінено вплив попередньої адаптації моделі LLM до технічного контексту мережі Sigfox на точність генерування її компонент і релевантність проєктних рішень, зокрема у випадках, коли потрібно зменшити частоту передавання даних способом їх агрегації або буферизації. Описано шість рівнів системного підходу (архітектурний, рівень взаємодії компонент, рівень оброблення даних, життєвого циклу пристроїв, протоколів передачі, рівень інтеграції із хмарними сервісами), які реалізовано в межах запропонованої методики. Оцінено ефективність підходу в контексті розроблення IoT-систем із жорсткими ресурсними обмеженнями, де традиційні інструменти не забезпечують належного рівня оптимізації потоків даних. Отримані результати демонструють потенціал інтелектуальної автоматизації в IoT, відкриваючи шлях до оптимізації робочих процесів. Перспективи досліджень передбачають удосконалення методу автоматизованого генерування PFD-діаграм способом адаптації до інших LPWAN-технологій і розширення функціональності моделей LLM для складніших сценаріїв використання.

Ключові слова: штучний інтелект; великі мовні моделі; оптимізація потоків даних; технологічні потоки; контекстний аналіз; енергоефективні мережі; автоматизоване моделювання.

Вступ / Introduction

Мережа Інтернету речей активно розвивається, а технології малопотужних мереж, зокрема Sigfox, відіграють ключову роль у розробленні енергоефективних систем [1, 8]. У світовій науковій літературі значну увагу приділяють моделюванню IoT-систем за допомогою діаграм, таких як Data Flow Diagrams (DFD), Unified Modeling Language (UML) та Block Flow Diagrams (BFD) [4, 10, 14]. Дослідження також охоплюють застосування великих мовних моделей для автоматизації процедури генерування коду та оброблення технічних специфікацій вимог [13]. Проте автоматизоване генерування PFD-діаграм (англ. *Process Flow Diagrams*) для IoT-систем на базі мережі Sigfox, яке враховує їхні специфічні обмеження, залишається недостатньо дослідженою. Ця прогалина зумовлює актуальність розроблення методу, що поєднує можливості моделей LLM та інженерні вимоги мережі Sigfox, забезпечуючи швидке роз-

роблення точних технологічних схем.

Об'єкт дослідження – генерування PFD-діаграм для проєктування IoT-систем.

Предмет дослідження – методи і засоби генерування PFD-діаграм для проєктування IoT-систем на базі мережі Sigfox, що дасть змогу автоматично формувати структурований орієнтований граф, який відображатиме ключові її елементи.

Мета роботи – розробити метод автоматизованого генерування PFD-діаграм для проєктування IoT-систем на базі мережі Sigfox, який враховуватиме технічні обмеження цих систем і підвищуватиме ефективність процесу їх проєктування.

Для досягнення зазначеної мети визначено такі основні завдання дослідження:

- розробити алгоритм контекстного аналізу текстових описів для виділення компонент і потоків, що дасть змогу автоматично формувати структурований орієнтований граф, який відображатиме ключові елементи IoT-

Інформація про авторів:

Гончаренко Дмитро Валерійович, аспірант, кафедра системного аналізу та інформаційних технологій.

Email: honcharenko.d98@gmail.com; <https://orcid.org/0009-0004-4903-9915>

Мокін Віталій Борисович, д-р техн. наук, професор, кафедра системного аналізу та інформаційних технологій.

Email: vbmokin@vntu.edu.ua; <https://orcid.org/0000-0003-1946-0202>

Цитування за ДСТУ: Гончаренко Д. В., Мокін В. Б. Метод автоматизованого генерування PFD-діаграм для проєктування IoT-систем на базі мережі Sigfox. Науковий вісник НЛТУ України. 2025, т. 35, № 3. С. 88–95.

Citation APA: Honcharenko, D. V., & Mokin, V. B. (2025). Method for automated generation of PFD diagrams for designing IoT systems based on Sigfox. *Scientific Bulletin of UNFU*, 35(3), 88–95. <https://doi.org/10.36930/40350309>

системи (такі як давачі, модулі Sigfox, хмарні сервіси) та їхні взаємозв'язки, забезпечуючи основу для подальшого генерування PFD-діаграм із високою точністю та відповідністю технічним вимогам мережі Sigfox;

- реалізувати процедуру генерування структурованого коду діаграм з оптимізацією потоків даних, що забезпечить розроблення синтаксично коректного подання PFD-діаграм у форматі Mermaid.js;
- забезпечити візуалізацію PFD-діаграм, що дасть можливість створювати графічні подання технологічних потоків IoT-систем із автоматичною анотацією ключових параметрів, забезпечуючи інтуїтивно зрозуміле відображення структури системи для інженерів і полегшуючи аналіз та оптимізацію проєктних рішень;
- адаптувати метод до обмежень мережі Sigfox, зокрема способом агрегації даних і локальної буферизації даних, що зменшить частоту передачі повідомлень і енергоспоживання пристроїв, забезпечуючи відповідність обмеженням мережі на розмір повідомлень і добовий ліміт передачі, а також підвищує ефективність використання ресурсів у сценаріях із високою інтенсивністю даних.

Аналіз останніх досліджень та публікацій. Сучасні наукові дослідження активно вивчають проблему оптимізації IoT-систем, зокрема в контексті енергоефективних мереж LPWAN, таких як Sigfox, LoRaWAN і NB-IoT. Дослідники [3, 15] наголошують на важливості низького енергоспоживання, великої дальності передачі та масштабованості цих технологій для застосувань у моніторингу довкілля, розумних містах і промисловій автоматизації. Проте більшість досліджень зосереджені на апаратних аспектах, таких як вибір давачів чи оптимізація протоколів, і недостатньо уваги приділяють автоматизації процесу проєктування технологічних схем.

Автоматизація процесу проєктування IoT-систем набуває дедалі більшої актуальності через її трудомісткість і схильність до помилок ручного моделювання схем. Дослідження [2, 18] аналізують використання традиційних інструментів моделювання, таких як Data Flow Diagrams (DFD), Unified Modeling Language (UML) діаграми та Block Flow Diagrams (BFD). Дослідники [9] пропонують графічні редактори, такі як Microsoft Visio чи Lucidchart, однак ці інструменти обмежуються візуалізацією і потребують ручного коригування для адаптації до вимог мережі Sigfox, що знижує ефективність.

Великі мовні моделі активно досліджують в завданнях автоматизації технічних процесів, зокрема аналізу специфікацій вимог і генерування коду. Роботи [19, 20] вказують на здатність моделей LLM обробляти текстові описи, виділяти ключові сутності та синтезувати структуровані дані, наприклад, для розроблення програмного коду чи конфігурацій хмарних платформ. У контексті IoT, як зазначають автори дослідження [22], модель LLM використовують для інтерпретації вимог до систем або автоматизації процедури генерування документації. Проте їхнє застосування для генерування технологічних схем, таких як PFD, залишається обмеженим, особливо для LPWAN-мереж. Дослідження [5] пропонують інтеграцію моделей LLM із графічними інструментами, але ці підходи не враховують специфіки мережі Sigfox, зокрема необхідність оптимізації потоків даних для дотримання лімітів їх передачі, що створює прогалину в автоматизації процесу проєктування.

Отже, проведений аналіз вказує на потребу розроблення підходу, що поєднує великі мовні моделі, алгоритм оптимізації потоків даних і автоматизовану їх візуалізацію для генерування PFD-діаграм, адаптованих

до потреб мережі Sigfox. Запропонований метод дасть змогу забезпечити оптимальний баланс між тривалістю проєктування, точністю спроектованих схем і енергоефективністю IoT-систем в умовах обмежень LPWAN-мереж та складності ручного моделювання.

Матеріали та методи дослідження. Дослідження проводили з використанням методів оброблення природної мови, генеративного синтезу коду та графічної візуалізації. Для аналізу текстових описів IoT-систем застосовано велику мовну модель Llama 3 8B, попередньо натреновану на технічних даних про LPWAN-технології, зокрема мережі Sigfox. Ця модель структурує неформальні описи користувача у формалізований текстовий формат. Для генерування коду діаграм у форматі Mermaid.js використано модель Gemini-1.5-Flash від Google, яка забезпечує рівень синтаксичних помилок нижче 5 %, що значно перевищує точність Llama 3 8B у здатності генерувати код (помилки ~30 %). Візуалізація PFD-діаграм реалізована через бібліотеку Python-Mermaid у поєднанні з бібліотеками PIL і Matplotlib для збереження зображень у форматі PNG. Дослідження виконували у хмарному середовищі Google Colab із графічним процесором NVIDIA Tesla T4, що забезпечило достатню продуктивність для оброблення великих обсягів даних.

Методологія дослідження базується на послідовному виконанні трьох етапів:

1. Контекстний аналіз тексту: модель Llama 3 8B розпізнає іменовані сутності (компоненти системи, такі як давачі чи хмарні сервіси) та синтаксичні залежності, формуючи текстовий опис орієнтованого графа із вершинами (компоненти) і ребрами (потоки даних).
2. Синтез коду: Gemini-1.5-Flash генерує код Mermaid.js на підставі текстового опису структурованого графа, застосовуючи оптимізаційні стратегії, такі як агрегація даних або буферизація, для відповідності технічним вимогам мережі Sigfox.
3. Графічна візуалізація: код Mermaid.js рендериться у PFD-діаграму за допомогою Python-Mermaid із анотаціями щодо обсягу даних і частоти передачі.

Кожен етап супроводжується автоматичною перевіркою: аналіз тексту валідується правилами для мінімізації помилок інтерпретації, код перевіряється парсером Mermaid.js для усунення синтаксичних неточностей, а потоки даних ітеративно коригуються для відповідності обмеженням мережі.

Результати верифікували способом порівняння згенерованих PFD-діаграм із схемами, розробленими вручну, а також тестуванням на наборах даних із різними сценаріями IoT-систем. Метод адаптивний до нових текстових описів і масштабований для інших LPWAN-технологій завдяки модульній структурі алгоритму. Відтворюваність забезпечується відкритими інструментами та описом алгоритмів.

Результати дослідження та їх обговорення / Research results and their discussion

Передумови дослідження. Технології малопотужних мереж LPWAN, зокрема Sigfox, відіграють ключову роль у розвитку масштабованих IoT-систем, забезпечуючи низьке енергоспоживання та передачу даних на великі відстані [17]. Проте, проєктування таких систем стикається зі значними викликами через необхідність розроблення технологічних схем, таких як Process Flow Diagrams, які відображають потоки даних між компо-

нентами. Ручне моделювання PFD є трудомістким процесом, що вимагає значних часових затрат. Воно також схильне до помилок, таких як неправильне визначення потоків даних чи неврахування обмежень мережі, що призводить до зниження ефективності системи.

Традиційні інструменти моделювання IoT-систем охоплюють кілька типів діаграм, кожен із яких має свої обмеження. Data Flow Diagrams (DFD) ефективні для аналізу потоків даних у програмному забезпеченні, але не враховують апаратних особливостей мережі Sigfox, таких як обмеження на розмір повідомлень. Unified Modeling Language (UML) діаграми дають можливість детально моделювати поведінку системи, однак їхня складність ускладнює швидке розроблення схем у реальних проєктах із жорсткими ресурсними обмеженнями. Block Flow Diagrams (BFD) є простішими, але не забезпечують достатньої деталізації для відображення потоків даних, критичних для енергоефективних мереж. На відміну від них, PFD оптимально поєднують технологічну точність і здатність візуалізувати зв'язки між фізичними та логічними компонентами, враховуючи параметри, такі як обсяг даних і енергоспоживання. Це робить PFD ідеальним вибором для IoT-систем на базі мережі Sigfox, однак їхнє розроблення вручну залишається неефективним.

Сучасні тенденції в IoT-проектванні вказують на зростання попиту на автоматизовані інструменти, які скорочують тривалість розроблення та підвищують точність технологічних схем. Наявні графічні редактори, такі як Microsoft Visio чи Lucidchart, обмежуються функціями візуалізації та не адаптовані до технічних вимог мережі Sigfox, що призводить до необхідності ручного коригування схем. Наприклад, ці інструменти не дають змоги автоматично оптимізувати потоки даних для дотримання лімітів мережі. Водночас великі мовні моделі, які ефективно обробляють текстові описи та генерують структуровані дані, відкривають нові можливості для автоматизації. Їхня здатність інтерпретувати технічні специфікації та синтезувати код, підтверджена в завданнях генерування програмного забезпечення, створює передумови для інтеграції в процеси проектування IoT. Зокрема, моделі LLM можуть трансформувати текстові запити користувача у PFD-діаграми, враховуючи інженерні аспекти. Цей потенціал, поєднаний із потребою в енергоефективних рішеннях, обґрунтовує розроблення нових методів автоматизації для Sigfox-систем.

Концепція автоматизованого проектування. Розроблення IoT-систем на базі мережі Sigfox потребує розроблення Process Flow Diagrams, які відображають потоки даних між компонентами з урахуванням обмежень способом. Запропонована концепція спрямована на автоматизацію цього процесу шляхом інтеграції LLM з формалізованим алгоритмом оптимізації потоків даних і енергоефективності. На відміну від традиційних підходів, що покладаються на ручне моделювання чи графічні редактори, метод трансформує текстові описи IoT-систем, надані користувачем, у структуровані технологічні схеми. Наприклад, опис "система моніторингу температури з давачами, що передають дані через мережу Sigfox до хмари" автоматично перетворюється на PFD із компонентами, потоками та анотаціями щодо обсягу даних. Цей підхід усуває потребу в спеціалізованих навиках моделювання, скорочуючи тривалість проектування та підвищуючи точність створених діаграм.

Моделі LLM виступають як посередник між неструктурованим текстовим описом і структурованим поданням системи. Попередньо натренована на технічних даних про LPWAN-технології модель аналізує текст, ідентифікуючи ключові компоненти (давачі, Sigfox-модулі, хмарні сервіси) та їхні взаємозв'язки. Наприклад, у запиті "давач рН передає 4 байти кожні 30 хвилин" модель розпізнає компоненти (давач), потік даних (4 байти) і частоту (раз на 30 хвилин), формуючи орієнтований граф. Цей граф містить вершини (компоненти) та ребра (потоки) з атрибутами, такими як обсяг даних чи тип передачі. На відміну від універсальних моделей LLM, використана модель адаптована до контексту мережі Sigfox, що дає змогу враховувати її технічні особливості, такі як обмеження на розмір повідомлень, уже на етапі аналізу.

Запропонована концепція автоматизованого проектування IoT-систем вирізняється кількома перевагами порівняно з традиційними методами. По-перше, автоматизація усуває потребу в ручному коригуванні, що скорочує тривалість проектування системи. По-друге, інтеграція моделі LLM та алгоритму оптимізації потоків даних забезпечує технічну точність схем, адаптованих до мережі Sigfox, навіть для користувачів без глибоких знань про мережу. По-третє, вихідний код діаграм генерується у форматі Mermaid.js, що дає змогу легко інтегрувати PFD у робочі процеси через чат-боти чи інженерні платформи.

Архітектуру методу автоматизованого генерування PFD-діаграм на підставі моделі LLM для IoT-систем подано на рис. 1.

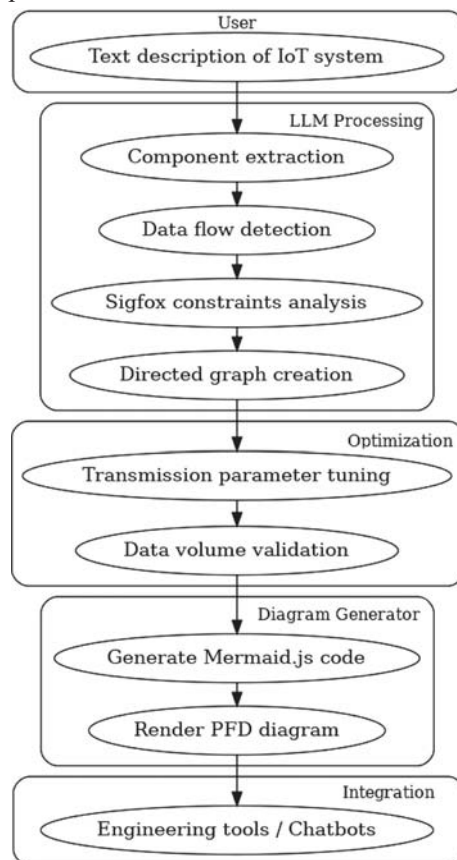


Рис. 1. Архітектура методу автоматизованого генерування PFD-діаграм на підставі моделі LLM для IoT-систем / PFD method architecture based on LLM for IoT systems

Метод автоматизованого генерування PFD-діаграм. Запропонований метод автоматизує процес роз-

роблення PFD-діаграм для IoT-систем на базі мережі Sigfox, інтегруючи великі мовні моделі та алгоритм оптимізації потоків даних. Цей метод розроблено для інженерів, які прагнуть швидко створювати технологічні схеми без ручного моделювання, забезпечуючи відповідність обмеженням мережі Sigfox і підвищуючи енергоефективність систем.

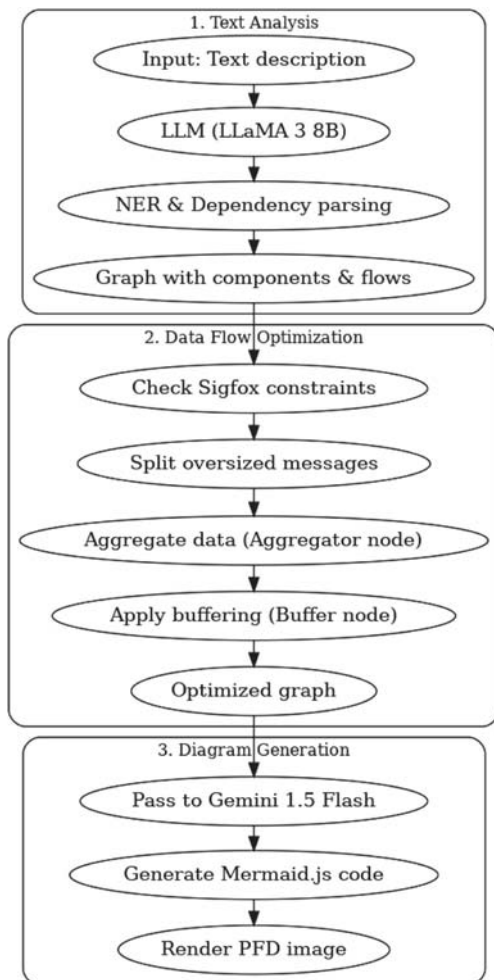


Рис. 2. Архітектура методу побудови PFD-діаграм / Three-level architecture of the PFD construction method

Метод містить такі три етапи:

1. *Аналіз тексту.* На першому етапі модель Llama 3 8B обробляє текстовий опис IoT-системи, наданий користувачем. Застосовуючи техніки оброблення природної мови, зокрема розпізнавання іменованих сутностей (NER) і аналіз синтаксичних залежностей, модель ідентифікує компоненти та їхні взаємозв'язки [16]. Граф анується атрибутами, такими як розмір даних і періодичність передачі, що забезпечує основу для подальшої його оптимізації.
2. *Оптимізація потоків даних.* Другий етап передбачає аналіз графа для адаптації потоків даних до обмежень мережі Sigfox за допомогою формалізованого алгоритму. Цей етап містить перевірку обсягу даних для кожного потоку, а також застосування стратегій оптимізації потоків даних, таких як агрегація даних чи локальна буферизація, для зменшення енергоспоживання і частоти передачі. Внаслідок граф оновлюється, додаючи вершини для агрегаторів чи буферів, що відображаються в PFD як додаткові компоненти.
3. *Генерування та візуалізація діаграми.* На фінальному етапі оптимізований граф передається до моделі Gemini-1.5-Flash, яка генерує код у форматі Mermaid.js – текстовому поданні PFD. Код містить вершини, ребра та анотації. Код рендериться за допомогою бібліотеки Ру-

thon-Mermaid у графічний формат. Автоматична перевірка синтаксису коду забезпечує відсутність помилок.

Архітектуру методу побудови PFD-діаграм для IoT-систем подано на рис. 2.

Для забезпечення високої точності розпізнавання структурних елементів IoT-систем було зібрано тематично орієнтований корпус із понад 300 наукових статей, що охоплюють як загальні характеристики LPWAN-мереж, так і приклади їх застосування в реальних інженерних рішеннях. На основі цього корпусу сформовано спеціалізований набір даних, який містить типові шаблони описів компонентів та їхніх функціональних взаємозв'язків. Модель Llama 38B була адаптована до цього домену шляхом донавчання, що дало змогу значно підвищити релевантність виявлених об'єктів у тексті та їх точне позиціонування у структурі майбутньої PFD-діаграми.

Оптимізація потоків даних є основою методу автоматизованого генерування PFD-діаграм, яка забезпечує відповідність PFD-діаграм технічним обмеженням мережі Sigfox і підвищує енергоефективність IoT-систем. Вона реалізується через формалізований алгоритм, який працює незалежно від моделі LLM і базується на математичній моделі орієнтованого графа $G=(V, E)$, де:

- 1) $V=\{v_1, v_2, \dots, v_n\}$ – множина вершин, що представляють компоненти системи;
- 2) $E=\{e_{ij}|v_i \rightarrow v_j \in V\}$ – множина ребер, що позначають потоки між компонентами;
- 3) атрибути ребер: w_{ij} – обсяг даних, f_{ij} – частота передачі, c_{ij} – коефіцієнт енергоспоживання, що залежить від типу компонента.

PFD-діаграма визначається як підмножина графа:

$$PFD \subseteq G, \quad PFD = (V, E, D), \quad (1)$$

де D – множина характеристик потоків, що містить w_{ij} і f_{ij} .

Обмеження мережі Sigfox формалізовано як:

- 1) $w_{ij} \leq 12$ байт для кожного повідомлення (uplink);
- 2) $\sum f_{ij} \leq 140$ повідомлень на день для всіх потоків (uplink);
- 3) для downlink-повідомлень: $w_{ij} \leq 8$ байт, $\sum f_{ij} \leq 4$ повідомлень на день;
- 4) цільова функція для мінімізації енергоспоживання:

$$E = \min_i \sum_j c_{ij} \cdot f_{ij}. \quad (2)$$

У формулі (2) мінімізація енергоспоживання виконується для значення суми, що охоплює всі ребра графа. Метою є визначення оптимальних частот передачі f_{ij} , які забезпечують мінімальне загальне енергоспоживання системи.

Алгоритм оптимізації потоків даних містить такі кроки:

1. Перевірка обмежень:

- для кожного ребра e_{ij} : якщо $w_{ij} > 12$, потік поділяють на кілька повідомлень:

$$w_{ij} = k \cdot 12 + w_{ij}, \quad k = \lceil w_{ij} / 12 \rceil, \quad (3)$$

- для downlink: якщо $w_{ij} > 8$, потік поділяють аналогічно:

$$w_{ij} = m \cdot 8 + w_{ij}, \quad m = \lceil w_{ij} / 8 \rceil, \quad (4)$$

де: k – кількість повних 12-байтних повідомлень, r – залишок у байтах; m – кількість повних 8-байтних повідомлень, s – залишок у байтах. Частота перевіряється на $\sum f_{ij} \leq 4$. якщо $\sum f_{ij} > 140$ для uplink, переходимо до оптимізації потоків даних.

2. Агрегація даних: якщо кілька компонент передають дані через один Sigfox-модуль, додається вершина v_a (агрегатор), яка об'єднує потоки:

$$w_a = \sum_i \sum_j w_{ij}, \quad f_a = \max_i \sum_j f_{ij}, \quad (5)$$

де: I – множина індексів компонентів (вершин v_i), які передають дані до фіксованого Sigfox-модуля, w_{ij} – обсяг даних для потоку від компонента i до модуля j , w_a – сумарний обсяг даних, агрегований у вершині v_a , f_a – максимальна частота серед усіх вхідних потоків до агрегатора;

- якщо $w_a > 12$, потоки розподіляються між кількома повідомленнями, а частота коригується.
3. Буферизація даних: якщо $\sum f_{ij} > 140$, додається вершина v_b (буфер), яка накопичує дані і відправляє їх рідше:

$$f_b = \frac{1}{k} \sum_i \sum_j f_{ij}, \quad (6)$$

де: f_b – нова частота передачі; k – мінімальна кількість циклів буферизації, необхідна для забезпечення $\sum f_{ij} \leq 140$; f_{ij} – частота передачі даних від компонента v_i до буфера v_b ; I_b – множина індексів компонентів, чий потоки спрямовані до буфера v_b .

Після проходження етапів аналізу та оптимізації, згенерований орієнтований граф подається у формалізованому вигляді на вхід моделі Gemini-1.5-Flash, яка відповідає за трансляцію цієї структури у синтаксично коректний код у форматі Mermaid.js. На цьому етапі не відбувається повторного семантичного аналізу – усі сутності, їхні взаємозв'язки, параметри передачі даних і обмеження вже ідентифіковано на попередніх етапах оброблення моделлю Llama. Завдання Gemini полягає у точному перетворенні опису графа в програмний шаблон, який можна зручно візуалізувати засобами бібліотеки Python. Подальша генерація зображень PFD-діаграм здійснюється у графічному форматі з анотаціями, що автоматично відповідають вихідним технічним характеристикам потоків. Вбудована перевірка синтаксису дає змогу уникати помилок і забезпечує готовність діаграм до використання в інженерних та аналітичних середовищах.

Метод вирізняється адаптивністю до обмежень мережі Sigfox, зокрема завдяки алгоритму оптимізації потоків даних, який зменшує частоту передачі даних на 20-30 % у сценаріях із високою їх інтенсивністю. Інтеграція двох моделей LLM (Llama для аналізу, Gemini для генерування) забезпечує баланс між контекстною точністю та синтаксичною надійністю. Отримані діаграми легко інтегруються в інженерні процеси, а автоматизоване генерування PFD-діаграм скорочує тривалість проектування порівняно з ручним моделюванням. Метод також підтримує двосторонній зв'язок (downlink), відображаючи зворотні потоки з анотаціями, що відповідають обмеженням.

Практична реалізація PFD-діаграм. Практична реалізація методу спрямована на розроблення технологічних схем для IoT-систем на базі мережі Sigfox, які відповідають обмеженням мережі. Метод генерує PFD на шести рівнях системного підходу – архітектурному, взаємодії, оброблення, життєвого циклу, протоколів та інтеграції. Кожен рівень фокусується на специфічних аспектах системи, даючи можливість інженерам комплексно оцінити її структуру, виявити потенційні вразливі місця та оптимізувати розподіл ресурсів. Реалізація методу автоматизує процес моделювання, скорочуючи тривалість проектування та забезпечуючи високу точність схем, що полегшує розгортання IoT-систем у реальних умовах.

Метод структурує PFD-діаграми за шістьма рівнями, кожен із яких виконує відповідну роль у процесі проектування:

1. *Архітектурний рівень* – відображає загальну структуру системи, враховуючи основні компоненти, такі як кінцеві пристрої, Sigfox-модулі, мережа Sigfox і хмарні сервіси. Потоки даних подані як узагальнені зв'язки без деталізації параметрів.
2. *Рівень взаємодії* – деталізує сеанси передачі між компонентами, враховуючи періодичність і напрямки потоків. Для мережі Sigfox відображаються uplink-потоки і downlink-потоки з анотаціями про частоту.
3. *Рівень оброблення* – фокусується на трансформації даних, таких як збирання, агрегація чи фільтрація перед передачею. Для дотримання ліміту 12 байт для uplink діаграма може містити проміжні компоненти.
4. *Рівень життєвого циклу* – відображає етапи роботи пристрою, такі як активація, передача, очікування та деактивація. Потоки даних пов'язані з циклами активності.
5. *Рівень протоколів* – деталізує використання протоколу мережі Sigfox, враховуючи ініціалізацію з'єднання, передачу повідомлень і підтвердження. Потоки ануються параметрами, такими як розмір пакета і тривалість сеансу, відображаючи низькорівневу взаємодію з мережею.
6. *Рівень інтеграції* – охоплює взаємодію системи з зовнішніми платформами. Діаграма показує потоки від мережі Sigfox Cloud до серверів оброблення чи інтерфейсів користувача, з анотаціями про декодування даних.

Описані рівні дають можливість створювати багатопланові PFD, які забезпечують як загальний огляд, так і деталізоване подання системи.

На рис. 3 подано приклад узагальнених PFD-діаграм архітектурного рівня та рівня оброблення.

Генерування PFD-діаграм на цих та інших рівнях дає змогу інженерам не тільки візуалізувати систему, але й приймати обґрунтовані інженерні рішення, адаптовані до специфіки мережі Sigfox, скорочуючи час на проектування та зменшуючи ризик помилок порівняно з ручним моделюванням.

Для демонстрації роботи методу розглянемо запит користувача: "Система моніторингу стану води з трьома давачами (температури, pH, провідності), кожен із яких передає 4 байти даних раз на 30 хвилин через мережу Sigfox до хмарного сервера". На архітектурному рівні PFD-діаграма відображає давачі, Sigfox-модуль і хмарний сервер із узагальненими потоками. На рівні оброблення алгоритм пропонує агрегацію: дані від трьох давачів об'єднуються в одне повідомлення, зменшуючи частоту до 48 передач на день. На рівні життєвого циклу діаграма показує цикли "пробудження – передача – очікування" з анотацією про 30-хвилинний інтервал. Рівень інтеграції ілюструє потік від мережі Sigfox Cloud до сервера, де дані декодуються для відображення в інтерфейсі.

На рис. 4 показано приклад автоматичного генерування PFD-діаграм на архітектурному рівні за запитом користувача. Унаслідок виконання користувач отримує візуалізовану версію PFD-діаграми та короткий текстовий опис функціонування інформаційної системи. Також користувач може згенерувати деталізовану діаграму з кожним наступним рівнем для проектування системи.

Подана на рис. 4 система починається з давачів, що вимірюють температуру, pH та провідність, і надсилають ці дані до блоку збирання даних. Потім ці дані передаються через мережу Sigfox на хмарний сервер для зберігання. Збережені дані аналізуються, що запускає сповіщення або повідомлення на підставі попередньо

визначених порогових значень. Одночасно дані візуалізуються для моніторингу. Хмарний сервер також надає API, що дає змогу зовнішнім системам отримувати доступ до зібраних даних та інтегруватися з ними, що за-

безпечує ширше застосування та сумісність. Загальний потік містить отримання даних давачів через бездротову передачу, хмарне сховище, аналіз і, нарешті, поширення через сповіщення, візуалізації та зовнішні API.

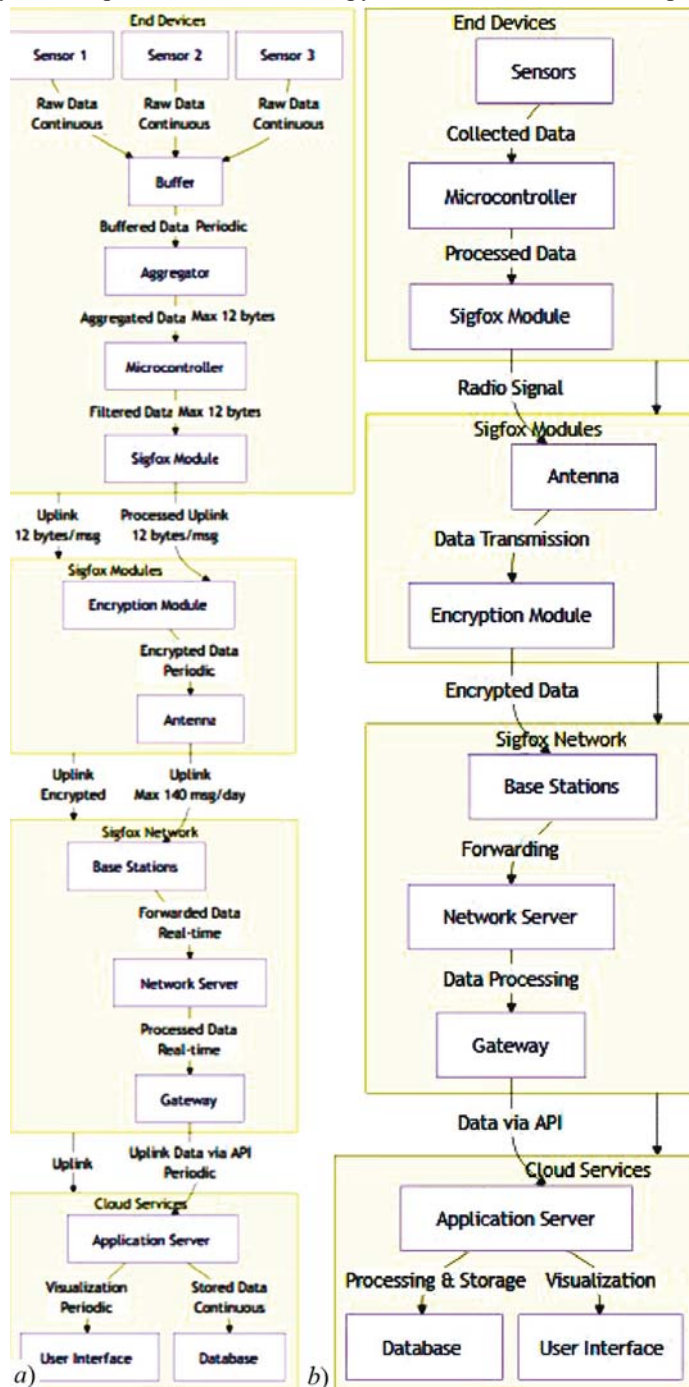


Рис. 3. Узагальнені PFD-діаграми архітектурного рівня та рівня оброблення / Generalized PFD diagrams of the architectural level and processing level

Запропонований метод автоматичного генерування PFD-діаграм на підставі текстових описів демонструє високу його ефективність під час проектування IoT-систем на базі мережі Sigfox, забезпечуючи при цьому швидке розроблення точних технологічних схем із врахуванням обмежень LPWAN-мереж. Реалізація шестирівневого підходу та інтеграція великих мовних моделей дає змогу не тільки автоматизувати трудомісткі процеси моделювання, але й оптимізувати потоки даних, зменшуючи при цьому енергоспоживання та підвищуючи надійність проектованої системи.

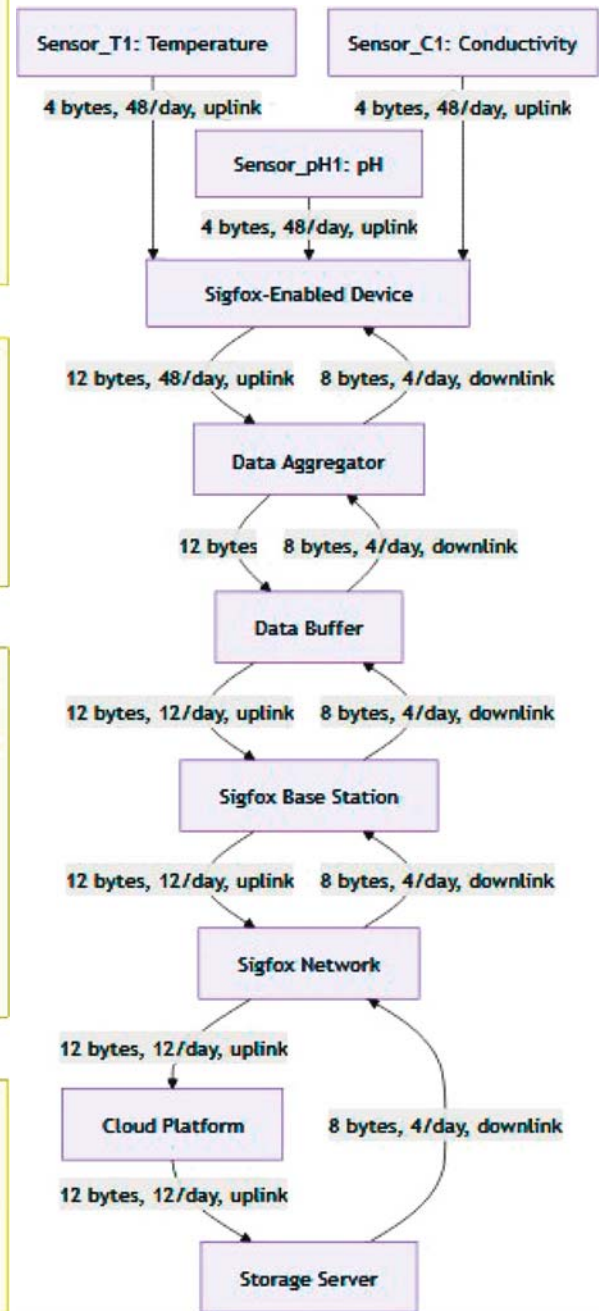


Рис. 4. Приклад автоматичного генерування PFD-діаграми на архітектурному рівні за запитом користувача / Example of automatic generation of a PFD diagram at the architectural level based on a user request

Обговорення результатів дослідження. У роботі [11] автори пропонують метод автоматизованого генерування Piping and Instrumentation Diagrams (P&ID) із текстових описів за допомогою моделі LLM і мульти-агентного підходу. Використовується ітеративний робочий процес для розроблення діаграм, що містить обладнання, трубопроводи та інструменти. Проте наведений метод фокусується на P&ID, а не на PFD-діаграмах, і не може бути адаптований до IoT-систем чи мережі Sigfox.

У дослідженні [12] запропоновано метод передбачення контрольних структур для PFD-діаграми за допо-

могою трансформерних моделей, де PFD-діаграми без структур перекладаються в PFD-діаграми зі структурами. Використовується нотація SFILES 2.0 для подання діаграм як рядків, із попереднім навчанням на згенерованих PFD-діаграмах і тонким налаштуванням на реальних даних. Незважаючи на практичну цінність методу, він зосереджений на додаванні контрольних структур до PFD-діаграм, а не на повному генеруванні діаграм для IoT-систем.

У роботі [6] розроблено напівавтоматизований інструмент для розроблення Data Flow Diagrams (DFD) із текстових запитів користувача. Використовуються техніки оброблення природної мови (NLP) і витягування даних із вебджерел для ідентифікації функціональності системи та її обмежень. Проте метод генерує DFD, які менш придатні для апаратних обмежень мережі Sigfox порівняно з PFD. Він не використовує моделі LLM чи алгоритми оптимізації потоків даних, а також не адаптований до IoT-систем чи LPWAN-мереж.

У дослідженні [21] автори проаналізували потенціал моделей LLM для генерування текстових моделей сценаріїв використання (use case models) для IoT-систем із неформальних описів. Запропонований метод генерує текстові моделі, а не графічні діаграми, як PFD. Він не адаптований до мережі Sigfox і не містить алгоритми оптимізації чи візуалізацію.

У роботі [7] автори пропонують платформу для автоматизації розроблення безконфліктних правил для розумних будинків за допомогою LLM. Вона генерує код для IoT-платформ і виявляє конфлікти, використовуючи формальну верифікацію. Запропонований у дослідженні метод зосереджений на генеруванні правил і кодів, а не діаграм. Він не придатний для мережі Sigfox і не містить процедуру генерування PFD-діаграм чи оптимізацію потоків даних.

Здійснений аналіз літературних джерел засвідчив, що, хоча в наукових роботах активно досліджують методи автоматизації процесу проектування IoT-систем, жоден із розглянутих підходів не поєднує великі мовні моделі, алгоритм оптимізації потоків даних та автоматизовану їх візуалізацію для генерування PFD-діаграм, адаптованих до мережі Sigfox. Запропонований у нашому дослідженні метод враховує технічні обмеження LPWAN-мереж і складність ручного моделювання в автоматизованому режимі, що дає змогу підвищити тривалість проектування, точність схем та енергоефективність IoT-систем.

Отже, внаслідок виконаної роботи можна сформулювати такі наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – отримав подальший розвиток метод автоматизованого генерування PFD-діаграм для проектування IoT-систем на базі мережі Sigfox, який поєднує великі мовні моделі, алгоритм оптимізації потоків даних та автоматизовану їх візуалізацію, що сукупно значно підвищує ефективність та пришвидшує процес їх проектування.

Практична значущість результатів дослідження – запропонований метод можна використати для автоматизованого проектування IoT-систем, застосувати для оптимізації енергоефективності та інтеграції в інженерні процеси, з урахуванням технічних обмежень мережі Sigfox і складності ручного моделювання.

Висновки / Conclusions

Розроблено метод автоматизованого генерування PFD-діаграм для проектування IoT-систем на базі мережі Sigfox, який враховує технічні обмеження цих систем і підвищує ефективність процесу їх проектування. За результатами проведеного дослідження можна зробити такі основні висновки.

1. Проаналізовано сучасні підходи до автоматизації проектування IoT-систем на основі літературних джерел, що дало змогу виявити їхні недоліки, зокрема відсутність адаптації до LPWAN-мереж типу Sigfox. Це обґрунтувало нагальну потребу розроблення нового методу, що враховує специфіку таких мереж.
2. Розроблено метод автоматизованого генерування PFD-діаграм, що базується на результатах контекстного аналізу, оптимізації потоків даних і візуалізації. Застосування цього методу дає змогу знизити трудомісткість проектування, підвищити точність схем і забезпечити відповідність технічним вимогам Sigfox.
3. Запропоновано шестирівневий підхід до аналізу IoT-систем, що охоплює архітектурні, функціональні, обробні, життєві, протокольні та інтеграційні рівні. Це забезпечує комплексне представлення IoT-системи, даючи змогу глибоко досліджувати її структуру та оптимізувати проектні рішення.
4. Застосовано розроблений метод автоматизованого генерування PFD-діаграм на прикладі IoT-системи моніторингу на базі мережі Sigfox. Практичне використання методу підтвердило його ефективність у швидкому створенні технологічних схем, адаптованих до обмежень цієї мережі, що свідчить про його доцільність для використання в інженерній практиці.

References

1. Adeoye, S. (2025). Internet of Things (IoT): A Vision, Architectural Elements and Future Directions. *Cognizance Journal of Multidisciplinary Studies*, 5(1), 316–338. <https://doi.org/10.47760/cognizance.2025.v05i01.027>
2. Aleryani, A. Y. (2024). Analyzing Data Flow: A Comparison between Data Flow Diagrams (DFD) and User Case Diagrams (UCD) in Information Systems Development. *European Modern Studies Journal*, 8(1), 313–320. [https://doi.org/10.59573/emsj.8\(1\).2024.28](https://doi.org/10.59573/emsj.8(1).2024.28)
3. Alsharif, M. H., Kelechi, A. H., Jahid, A., Kannadasan, R., Singla, M. K., Gupta, J., & Geem, Z. W. (2024). A comprehensive survey of energy-efficient computing to enable sustainable massive IoT networks. *Alexandria Engineering Journal*, 91, 12–29. <https://doi.org/10.1016/j.aej.2024.01.067>
4. Bonnefoi, F., Hillah, L. M., Kordon, F., & Renault, X. (2007). Design, modeling and analysis of ITS using UML and Petri Nets. 2007 *IEEE Intelligent Transportation Systems Conference*, pp. 52–58. <https://doi.org/10.1109/itsc.2007.4357718>
5. Carroll, A. J., & Borycz, J. (2024). Integrating large language models and generative artificial intelligence tools into information literacy instruction. *The Journal of Academic Librarianship*, 50(4), article ID 102899. <https://doi.org/10.1016/j.acalib.2024.102899>
6. Cheema, S. M., Tariq, S., & Pires, I. M. (2023). A Natural Language Interface for Automatic Generation of Data Flow Diagram using Web Extraction Techniques. *Journal of King Saud University – Computer and Information Sciences*, 35(3), 1–15. <https://doi.org/10.1016/j.jksuci.2023.01.006>
7. Cheng, Y., Xu, M., Zhang, Y., Li, K., Wang, R., & Yang, L. (2025). AutoIoT: Automated IoT Platform Using Large Language Models. *IEEE Internet of Things Journal*, 1, 1–12. <https://doi.org/10.1109/ijot.2024.3523907>
8. Ding, J., Nemati, M., Ranaweera, C., & Choi, J. (2020). IoT Connectivity Technologies and Applications: A Survey. *IEEE Access*, 8, 67646–67673. <https://doi.org/10.1109/access.2020.2985932>
9. Faulkner, A. (2018). Lucidchart for Easy Workflow Mapping. *Serials Review*, 44(2), 157–162. <https://doi.org/10.1080/00987913.2018.1472468>

10. Filho, T., Fernando, L., Rabelo, M., Silva, S., Santos, C., Ribeiro, M., Grout, I. A., Moreira, W., & Oliveira, A. (2021). A Standard-Based Internet of Things Platform and Data Flow Modeling for Smart Environmental Monitoring. *Sensors*, 21(12), article ID 4228. <https://doi.org/10.3390/s21124228>
11. Gowaikar, S., Iyengar, S., Segal, S., & Kalyanaraman, S. (2024). An Agentic Approach to Automatic Creation of P&ID Diagrams from Natural Language Descriptions. *arXiv:2412.12898*, 45–53. <https://doi.org/10.48550/arXiv.2412.12898>
12. Hirtreiter, E., Schulze Balhorn, L., & Schweidtmann, A. M. (2023). Toward automatic generation of control structures for process flow diagrams with large language models. *AIChE Journal*, article ID e18259, 1–15. <https://doi.org/10.1002/aic.18259>
13. Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J., & Wang, H. (2024). Large Language Models for Software Engineering: A Systematic Literature Review. *ACM Transactions on Software Engineering and Methodology*, 33(8), 1–7. <https://doi.org/10.1145/3695988>
14. James, A., Seth, A., & Mukhopadhyay, S. C. (2022). IoT System Design. Project Based Approach. *Springer International Publishing*, 165–169. <https://doi.org/10.1007/978-3-030-85863-6>
15. Jouhari, M., Saeed, N., Alouini, M.-S., & Amhoud, E. M. (2023). A Survey on Scalable LoRaWAN for Massive IoT: Recent Advances, Potentials, and Challenges. *IEEE Communications Surveys & Tutorials*, 1, 5–12. <https://doi.org/10.1109/comst.2023.3274934>
16. Mohamed Ali El-Morsy, S., Hussein, M., & Mousa, H. M. (2022). Arabic open information extraction system using dependency parsing. *International Journal of Electrical and Computer Engineering (IJECE)*, 12(1), article ID 541. <https://doi.org/10.11591/ijece.v12i1.pp541-551>
17. Osman, D. N., & Abbas, E. (2022). Performance Evaluation of LoRa and Sigfox LPWAN Technologies for IoT. *Academic Journal of Research and Scientific Publishing*, 4(38), 05–24. <https://doi.org/10.52132/ajrsp.e.2022.38.1>
18. Richard, J. B. (2013). Supporting Engineering Design Modeling by Domain Specific Modeling Language. *American Journal of Software Engineering and Applications*, 2(3), 86. <https://doi.org/10.11648/j.ajsea.20130203.11>
19. Shrivastava, A., & Jain, J. K. (2020). Natural language processing: A literature survey. *International Journal of Communication and Information Technology*, 1(2), article ID 15–20. <https://doi.org/10.33545/2707661x.2020.v1.i2a.13>
20. Soliman, A., Shaheen, S., & Hadhoud, M. (2024). Leveraging pre-trained language models for code generation. *Complex & Intelligent Systems*, 10, 47–52. <https://doi.org/10.1007/s40747-024-01373-8>
21. Tabassum, M. R., Ritchie, M. J., Mustafiz, S., & Kienzle, J. (2024). Using LLMs for Use Case Modelling of IoT Systems: An Experience Report. *MODELS Companion 24: ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*, 611–619. ACM. <https://doi.org/10.1145/3652620.3687810>
22. Zong, M., Hekmati, A., Guastalla, M., Li, Y., & Krishnamachari, B. (2025). Integrating large language models with internet of things: applications. *Discover Internet of Things*, 5(2). <https://doi.org/10.1007/s43926-024-00083-4>

D. V. Honcharenko, V. B. Mokin

Vinnitsia National Technical University, Vinnitsia, Ukraine

METHOD FOR AUTOMATED GENERATION OF PFD DIAGRAMS FOR DESIGNING IOT SYSTEMS BASED ON SIGFOX

New approaches to automating the design process of information systems based on Internet of Things (IoT) technologies have been proposed, taking into account the specifics of energy-efficient networks such as Sigfox. It has been established that effectively considering the constraints of such networks requires the development of specialized methods for designing technological diagrams. It has been found that the use of Large Language Models (LLMs) enables a sufficient level of automation in the generation of Process Flow Diagrams (PFDs) based on user-provided textual descriptions. A method of contextual analysis is described, allowing the extraction of key IoT system components, inter-component relationships, and data flows from textual information. The transition from traditional manual modeling to intelligent diagram synthesis using LLMs is substantiated, which significantly reduces the time required for IoT system development and improves the accuracy of the results. A three-stage method has been developed, which includes: linguistic analysis and structuring of the system description, code generation procedure with optimization of data flows under LPWAN network constraints, and automated visualization of data flows in the form of PFD diagrams. The impact of pre-adapting the LLM to the technical context of Sigfox networks on the accuracy of component generation and the relevance of design decisions has been evaluated, particularly in cases where data transmission frequency needs to be reduced through aggregation or buffering. Six levels of a systems approach are described, in particular, architectural, component interaction, data processing, device lifecycle, communication protocols, and cloud integration levels, which are implemented within the proposed methodology. The effectiveness of the approach is assessed in the context of designing IoT systems with strict resource constraints, where traditional tools fail to provide an adequate level of data flow optimization. The results demonstrate the potential of intelligent automation in IoT, paving the way for workflow optimization. Future research is expected to focus on enhancing the method through adaptation to other LPWAN technologies and extending the capabilities of LLMs for more complex use scenarios.

Keywords: artificial intelligence; large language models (LLMs); data flow optimization; process flows; contextual analysis; energy-efficient networks; automated modeling.