

МОДЕЛЬ ОБРОБКИ ПОДІЙ У СИСТЕМІ ЛОКАЛЬНОГО СПОВІЩЕННЯ НА БАЗІ RASPBERRY PI

Бойко Тарас Мирославович

Київський столичний університет імені Бориса Грінченка

ВСТУП

Актуальність проблеми локального сповіщення зростає в умовах, коли інформаційні повідомлення мають доставлятися не лише централізовано, а й безпосередньо в конкретних приміщеннях, закладах освіти, громадських просторах або виробничих локаціях. Для таких сценаріїв важливою є не тільки наявність каналу передавання даних, а й здатність кінцевого пристрою самостійно обробляти події, перевіряти їх актуальність, визначати пріоритет і запускати відповідне аудіоповідомлення. Ця логіка узгоджується з підходом *edge computing*, за якого частина обчислень і прийняття рішень переноситься ближче до місця виникнення подій [1].

У традиційних системах оповіщення значна частина керування виконується централізовано. Такий підхід є зручним для адміністрування, однак створює залежність від стабільності мережевого з'єднання, зовнішнього сервера та своєчасності доставки команди. Для локальних систем сповіщення це може бути критичним, оскільки окремі повідомлення мають бути відтворені у визначений момент часу або відразу після зміни стану небезпеки. Тому доцільним є використання програмно-апаратного вузла, який може працювати автономно, зберігати локальні правила обробки подій і виконувати частину сценаріїв навіть за відсутності інтернету.

Окремий напрям досліджень пов'язаний з IoT-системами попередження. У роботі I.-D. Filip, C.-M. Plescu та F. Pop запропоновано IoT-based warning system, у якій акцент зроблено на селективності, масштабованості та можливості доставляти попередження відповідно до конкретних умов [2]. Ця ідея є важливою для локальних систем, оскільки сповіщення не повинні бути універсальними для всіх вузлів: вони можуть залежати від регіону, приміщення, локації, типу загрози або локального сценарію використання.

У сучасних дослідженнях систем громадської безпеки також підкреслюється значення швидкого виявлення подій, їх класифікації та поширення попереджень у режимі реального часу. H. Zhang, R. Zhang та J. Sun описують IoT-based public safety alert system, що поєднує сенсори, edge-вузли, хмарні платформи та механізми автоматизованого інформування [3]. Попри інший прикладний контекст, така логіка підтверджує доцільність моделі «подія — обробка — рішення — сповіщення» для систем, що працюють у сфері безпеки та локального інформування.

Для проєктування системи локального сповіщення важливо розглядати події не як окремі команди запуску аудіофайлу, а як структуровані інформаційні об'єкти. У дослідженні J. Rosa Bilbao, R. González Álvarez та J. Boubeta Puig представлено платформу Zync, орієнтовану на керування IoT-пристроями та complex event monitoring [4]. Автори акцентують на подієво-

керованій архітектурі, правилах, порогах і сповіщеннях, що дає підстави використовувати подібну логіку для опису локальної системи сповіщення на Raspberry Pi.

Ще одним суттєвим аспектом є коректність часу. Для подій, прив'язаних до конкретного моменту, наприклад хвилини мовчання о 09:00 або шкільних дзвінків, система не може повністю залежати від мережевої синхронізації. А. Burlibaşa та співавтори показують, що для Raspberry Pi-систем важливою є автономна синхронізація часу, робота після вимкнення живлення, використання RTC і системних сервісів для відновлення коректного часу [5]. У межах цієї роботи цей підхід використовується як теоретичне обґрунтування застосування RTC-модуля в локальному вузлі сповіщення.

Метою дослідження є розроблення моделі обробки подій у системі локального сповіщення на базі Raspberry Pi, яка забезпечує приймання API-подій, обробку часових, розкладових і календарних подій, перевірку актуальності повідомлень, визначення пріоритетів та запуск відповідних аудіофайлів.

Поставлена проблема належить до галузі комп'ютерних наук, оскільки її ядром є не саме відтворення звуку, а організація інформаційних процесів: отримання даних із зовнішнього API, класифікація подій, збереження поточного стану, побудова правил переходу між станами, обробка пріоритетів, локальне прийняття рішення та журналювання стану системи.

КОРОТКИЙ ОПИС ДОСЛІДЖЕННЯ, ЙОГО МЕТОДІВ І ЗАСОБІВ

У дослідженні розглянуто прототип системи локального сповіщення, реалізований на базі Raspberry Pi 3B+ або Raspberry Pi 4. Вибір конкретної моделі в межах запропонованої задачі не є принциповим, оскільки основні функції системи пов'язані з обробкою подій, запуском аудіофайлів, роботою з локальними конфігураціями та виконанням скриптів. Raspberry Pi у цій системі виконує роль локального edge-вузла, який не лише приймає зовнішні сигнали, а й самостійно визначає, чи потрібно відтворювати повідомлення.

Методологічною основою дослідження є системний аналіз, подієво-кероване моделювання, моделювання переходів між станами та алгоритмічний опис правил обробки подій. Системний аналіз використано для виокремлення джерел подій і функціональних модулів системи. Подієво-кероване моделювання застосовано для опису структури події та логіки її обробки. Модель переходів між станами використано для формалізації сценаріїв жовтого рівня небезпеки, червоного рівня небезпеки та відбою повітряної загрози.

У запропонованій моделі кожна подія описується такими ознаками: джерело виникнення, тип події, регіональна прив'язка, час виникнення або час виявлення, рівень пріоритету, відповідний аудіофайл і допустимий інтервал актуальності. Такий опис дає змогу відокремити власне джерело сигналу від логіки прийняття рішення. Наприклад, API-подія, календарна подія і розкладова подія мають різне походження, але проходять спільні етапи обробки: нормалізацію, перевірку умов, визначення пріоритету, вибір файлу та запуск відтворення.

Система обробляє кілька груп подій. До першої групи належать безпекові API-події, пов'язані зі станом повітряної загрози. У межах прототипу використовуються три основні стани: жовтий рівень небезпеки, червоний рівень небезпеки та відбій повітряної загрози. Дані надходять із підключеного API, який передає інформацію про поточні стани у форматі логічних значень true / false. Конфігураційні файли прив'язуються до відповідного регіону, що дає змогу відтворювати різні аудіофайли для різних локацій.

Друга група охоплює часові події, зокрема хвилину мовчання о 09:00. Для забезпечення роботи цього сценарію використовується RTC-модуль DC32-31, який дає змогу зберігати коректну дату й час навіть після тривалого вимкнення пристрою. Це важливо, оскільки хвилина мовчання має сенс саме в заданий момент часу, а її запуск із суттєвим запізненням не відповідає призначенню події.

Третя група подій пов'язана з локальним розкладом. У прототипі реалізовано сценарій шкільних дзвінків: для робочих днів задається розклад, обирається аудіофайл дзвінка, а у вихідні дні відтворення не виконується. Четверта група — календарні події, які можуть бути пов'язані з пам'ятними датами, тематичними днями або іншими локальними інформаційними повідомленнями. Такі події задаються через cron із прив'язкою до конкретної дати та відповідного аудіофайлу.

Для відтворення аудіофайлів використовується VLC player. Стан повітряної загрози фіксується у log-файлі, окремо ведеться log-файл для сценарію шкільних дзвінків. Журналювання календарних дат у поточній реалізації не передбачене, оскільки для цього сценарію достатньо факту виконання завдання через cron. Таке рішення відповідає прикладному характеру прототипу та не ускладнює систему зайвими механізмами обліку.

Узагальнена модель обробки подій у локальній системі сповіщення

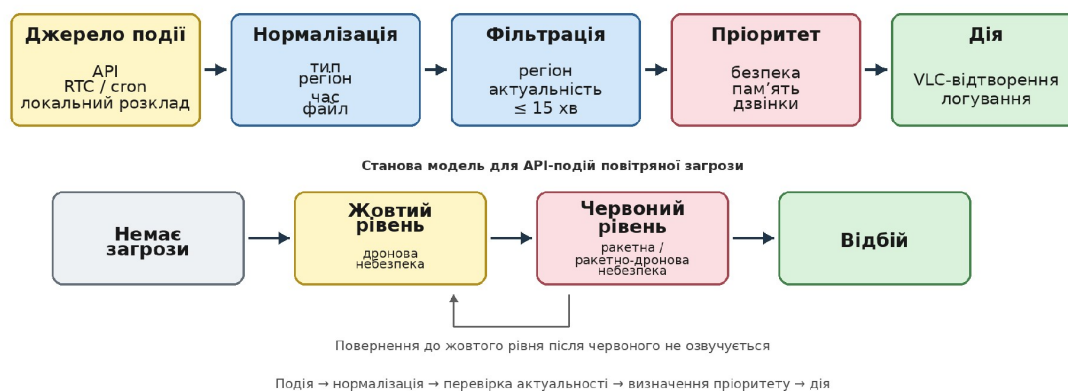


Рис. 1. Узагальнена модель обробки подій у локальній системі сповіщення

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Основним результатом дослідження є формалізація моделі обробки подій для локальної системи сповіщення на Raspberry Pi. Запропонована модель дає змогу об'єднати різні джерела подій у спільний алгоритм прийняття рішення. Це важливо, оскільки безпекові повідомлення, хвилина мовчання, шкільні

дзвінки та календарні аудіоподії мають різну природу, але всі вони конкурують за один ресурс — аудіоканал локального пристрою.

Таблиця 1.

Типи подій у системі локального сповіщення

Тип події	Джерело	Приклад	Особливість обробки
API-подія	API повітряних загроз	Жовтий рівень, червоний рівень, відбій	Перевірка регіону, поточного стану та актуальності
Часова подія	RTC-модуль DC32-31	Хвилина мовчання о 09:00	Має виконуватися у точний момент часу
Розкладова подія	Локальний розклад / cron	Шкільний дзвінок	Виконується у будні дні за заданим графіком
Календарна подія	cron / дата року	Пам'ятна дата або локальне оголошення	Відтворює визначений файл у конкретну дату

У межах моделі ключовим є не сам факт отримання сигналу, а його інтерпретація з урахуванням попереднього стану. Наприклад, жовтий рівень небезпеки запускає повідомлення про дроніву небезпеку. Якщо після цього надходить червоний рівень небезпеки, система відтворює інше повідомлення — про ракетну або ракетно-дроніву небезпеку. Водночас після фіксації червоного рівня повернення до жовтого не озвучується, оскільки користувачі вже були повідомлені про вищий рівень ризику. Після червоного рівня очікується лише відбій повітряної загрози.

Таблиця 2.

Модель переходів між станами повітряної загрози

Поточний стан	Нова подія	Дія системи
Немає активної загрози	Жовтий рівень небезпеки	Відтворити аудіофайл про жовтий рівень небезпеки
Жовтий рівень небезпеки	Червоний рівень небезпеки	Відтворити аудіофайл про червоний рівень небезпеки
Червоний рівень небезпеки	Жовтий рівень небезпеки	Не відтворювати повідомлення, оскільки вищий рівень уже зафіксовано
Жовтий або червоний рівень небезпеки	Відбій повітряної загрози	Відтворити аудіофайл про відбій
Будь-який стан	Подія старша за 15 хвилин	Не відтворювати повідомлення як застаріле

Окремим результатом є реалізація правила актуальності події. Якщо мережеве з'єднання тимчасово зникає, а після його відновлення система отримує інформацію про подію, що сталася раніше, то повідомлення не завжди має відтворюватися. У прототипі використано поріг 15 хвилин: якщо з моменту події минуло не більше 15 хвилин, відтворення допускається; якщо минуло більше 15 хвилин, подія вважається застарілою і не озвучується. Це правило запобігає ситуації, коли користувачі отримують аудіоповідомлення про небезпеку із суттєвим запізненням, яке вже не відповідає поточному контексту.

Модель пріоритетів у системі побудована навколо принципу безпекової першості. Найвищий пріоритет мають АРІ-сповіщення, пов'язані з повітряною загрозою. Далі розташовуються події, прив'язані до суспільно значущого часу, зокрема хвилина мовчання о 09:00. Нижчий пріоритет мають звичайні розкладові події, наприклад шкільні дзвінки. Водночас пріоритетність не є жорстко незмінною: для конкретної локації вона може налаштовуватися відповідно до локального сценарію використання.

Таблиця 3.

Узагальнена модель пріоритетів подій

Пріоритет	Тип події	Приклад дії системи
1	Безпекове АРІ-сповіщення	Озвучення жовтого або червоного рівня небезпеки, відбою
2	Часова подія суспільного значення	Відтворення хвилини мовчання о 09:00
3	Розкладова подія	Відтворення шкільного дзвінка за графіком
4	Календарна інформаційна подія	Відтворення аудіофайлу до визначеної дати

Практична цінність запропонованої моделі полягає в можливості локалізувати зміст аудіоповідомлення. Для кожної окремої локації можна підготувати файл, у якому зазначено, куди саме рухатися, де розташоване найближче укриття або які дії мають виконати присутні у приміщенні. Така локалізація є відмінністю від загальних централізованих повідомлень, оскільки система не просто інформує про факт загрози, а може надавати інструкцію, релевантну конкретному простору.

З погляду інформаційної архітектури запропонована модель містить п'ять послідовних етапів: отримання або формування події; нормалізацію її параметрів; перевірку регіональної належності й актуальності; визначення пріоритету щодо інших потенційних подій; запуск відповідного аудіофайлу та фіксацію стану в журналі. Така структура дає змогу надалі розширювати систему без зміни її основної логіки. Наприклад, до моделі можна додати інші типи АРІ-подій, локальні датчики або окремий модуль адміністрування конфігурацій.

Запропонована система відрізняється від звичайного аудіоплеєра тим, що відтворення не є головною функцією, а лише кінцевою дією після обробки інформаційної події. Саме тому центральним елементом дослідження є модель

обробки подій, а не програмний засіб відтворення. Такий підхід відповідає логіці подієво-керованих IoT-систем, у яких рішення формується на підставі стану, правил і контексту [4].

Застосування RTC-модуля DC32-31 дає змогу зменшити залежність часових сценаріїв від інтернету. Це особливо важливо для хвилини мовчання та розкладових подій, оскільки вони мають виконуватися у визначений момент часу. У дослідженнях офлайн-синхронізації для Raspberry Pi підкреслюється, що автономне збереження часу й передбачуване відновлення після вимкнення живлення є важливими умовами для довготривалих локальних інсталяцій [5-6]. У розглянутому прототипі ця ідея реалізована на прикладному рівні через використання апаратного RTC-модуля та локальних правил запуску.

Отже, результати дослідження показують, що для локальної системи сповіщення доцільно використовувати не набір незалежних скриптів, а єдину модель подій. Вона дозволяє описати джерело, зміст, час, регіон, актуальність і пріоритет кожного повідомлення. Це створює основу для подальшого масштабування системи, додавання нових подій і формального тестування її поведінки в різних сценаріях.

ВИСНОВКИ

У роботі запропоновано модель обробки подій у системі локального сповіщення на базі Raspberry Pi. Модель орієнтована на використання Raspberry Pi 3B+ або Raspberry Pi 4 як локального edge-вузла, здатного приймати API-події, виконувати розкладові й календарні сценарії, перевіряти актуальність повідомлень і запускати відповідні аудіофайли через VLC player.

Встановлено, що ключовою вимогою до такої системи є не лише своєчасне отримання даних, а й коректна інтерпретація події. Для цього в моделі передбачено опис події через джерело, тип, регіон, часову мітку, пріоритет і пов'язаний аудіофайл. Такий підхід дає змогу поєднати в одній системі безпекові API-сповіщення, хвилину мовчання, шкільні дзвінки та календарні аудіоподії.

Формалізовано логіку переходів між жовтим рівнем небезпеки, червоним рівнем небезпеки та відбоєм повітряної загрози. Особливістю моделі є те, що після червоного рівня жовтий статус повторно не озвучується, а система очікує відбій. Також реалізовано правило фільтрації застарілих подій: якщо після втрати інтернету повідомлення стало доступним пізніше ніж через 15 хвилин, воно не відтворюється.

Підтверджено доцільність використання RTC-модуля DC32-31 для сценаріїв, прив'язаних до точного часу. Це дозволяє виконувати хвилину мовчання о 09:00, розкладові та календарні події навіть за умов нестабільного мережевого з'єднання або після тривалого вимкнення пристрою.

Подальший розвиток дослідження може бути спрямований на розширення журналювання подій, створення веб-інтерфейсу для налаштування правил, формалізацію тестових сценаріїв і експериментальну оцінку затримки реакції системи на API-події. Водночас уже на поточному етапі запропонована модель може бути використана як основа для локальних систем сповіщення в

закладах освіти, громадських просторах або інших об'єктах, де важливі автономність, регіональна прив'язка та пріоритетність повідомлень.

ДЖЕРЕЛА

1. Shi W., Cao J., Zhang Q., Li Y., Xu L. Edge Computing: Vision and Challenges. IEEE Internet of Things Journal. 2016. Vol. 3, No. 5. P. 637–646. DOI: 10.1109/JIOT.2016.2579198.
2. Filip I.-D., Iliescu C.-M., Pop F. Assertive, Selective, Scalable IoT-Based Warning System. Sensors. 2022. Vol. 22, No. 3. Article 1015. DOI: 10.3390/s22031015.
3. Zhang H., Zhang R., Sun J. Developing real-time IoT-based public safety alert and emergency response systems. Scientific Reports. 2025. Vol. 15. Article 29056. DOI: 10.1038/s41598-025-13465-7.
4. Rosa Bilbao J., González Álvarez R., Boubeta Puig J. Zync: A microservice platform for IoT device management and complex event monitoring. Computers and Electrical Engineering. 2026. Vol. 129. Article 110752. DOI: 10.1016/j.compeleceng.2025.110752.
5. Burlibaşa A., Epure S., Culea M., Dache C. R., Lungu C. V., Marin G.-A., Vlad C. System-Level Offline Time Synchronization Architecture for Distributed Electrical Signal Monitoring Using Raspberry Pi 5. Sensors. 2026. Vol. 26, No. 8. Article 2519. DOI: 10.3390/s26082519.
6. Abramov, V., Astafieva, M., Boiko, M., Bodnenko, D., Bushma, A., Vember, V., Hlushak, O., Zhylytsov, O., Ilich, L., Kobets, N., Kovaliuk, T., Kuchakovska, H., Lytvyn, O., Lytvyn, P., Mashkina, I., Morze, N., Nosenko, T., Proshkin, V., Radchenko, S., & Yaskevych, V. (2021). Theoretical and practical aspects of the use of mathematical methods and information technology in education and science.
URL: <https://doi.org/10.28925/9720213284km>