

А.Р. Петросян, магістрант
Р.В. Петросян, ст. викладач
К.Р. Колос, д.пед.н., проф.

Державний університет «Житомирська політехніка»

Розробка платформи віддаленого управління інфраструктурою Інтернет речей

У статті розглядається платформа для віддаленого управління (моніторингу) IoT-пристроїв, які в останній час дуже широко поширюються по всьому світу. Водночас постає проблема підвищення рівня ефективності проектування інформаційних систем, що будуть надавати доступ до різноманітної інформації з будь-якого куточка світу, де є доступ до мережі «Інтернет». Тому важливим є уважний та ретельний підхід до вибору архітектури платформи.

IoT-платформа – це набір компонентів, які забезпечують: взаємодію з додатками, віддалений збір даних з датчиків, безпечно підключення та управління пристроями. IoT-платформа забезпечує перевагу при створенні IoT-систем за рахунок надання інструментів розробки, що робить Інтернет речей простіше і дешевше для кінцевих користувачів.

Було проаналізовано функціональні можливості платформи. Обґрунтовано основні підходи до її побудови. Запропоновано архітектуру платформи, яка дозволяє взаємодіяти користувальницьким додаткам з IoT-пристроями.

У результаті було отримано діючий прототип IoT-системи, що демонструє принцип роботи цієї платформи. Основу системи становить сервер, який взаємодіє з клієнтами та IoT-пристроями. Однак взаємодія з пристроями сервером виконується за допомогою MQTT-брокера. Взаємодія клієнтів із сервером виконується за протоколом WebSocket. Програмне забезпечення для сервера створено за допомогою мови програмування Node.js, TypeScript і фреймворка LoopBack. Для створення клієнтського інтерфейсу було використано такий стек технологій: CSS, HTML, Javascript, React, Material-UI.

Розроблена платформа є дуже гнучкою та дозволяє: підключати безліч різноманітних пристроїв; конфігурувати пристрої через браузер; створювати сценарії для керування системою в цілому (сценарії створюються за допомогою візуального програмування).

Ключові слова: Інтернет речей; IoT; платформа; архітектура; віддалене управління.

Актуальність теми. Підвищення рівня проживання – одне із завдань, яке постає перед людиною. Однак це неможливо без появи нової техніки та технологій. За останній час швидкими темпами розвивається цифрова електроніка, мікропроцесорна техніка, IT-індустрія тощо. Завдяки бурхливому розвитку техніки та появи нових технологій з'явилися малогабаритні сенсори, інтерфейсні модулі зв'язку, програмовані модулі управління на базі мікроконтролерів тощо. Все це зумовило виникнення такого напрямку, як Інтернет речей (англ. Internet of Things, скор. IoT).

Вперше термін «Інтернет речей» був введений Кевіном Ештоном у 1999 році [1]. Починаючи з цього моменту, IoT набрав значного оберту як технологія підключення фізичних об'єктів до інтернету. Інтернет речей – це концепція обчислювальної мережі фізичних предметів (речей), оснащених вбудованими технологіями для взаємодії один з одним та із зовнішнім середовищем.

Водночас постає проблема підвищення рівня ефективності проектування інформаційних систем, які будуть надавати доступ до різноманітної інформації з будь-якого куточка світу, де є доступ до мережі «Інтернет». Все більше IoT-пристроїв та систем з'являється з кожним днем, а разом з цим підвищується рівень їх складності, що, в свою чергу, обумовлює збільшення зусиль та часу на їхню підтримку й покращення зручності роботи з додатками. Тому важливим є уважний та ретельний підхід до вибору архітектури системи, що дозволить визначити подальший розвиток всього проекту. Також дуже важливим є інтерфейс взаємодій з IoT-системами. В цілому побудова великих і складних систем є непростюю задачею й потребує багато компромісних рішень.

Аналіз останніх досліджень та публікацій, на які спираються автори. IoT-технології перебувають на ранній стадії розвитку та активних досліджень. Однак в останні роки досягнуто суттєвого прогресу в області дослідження мереж Інтернет речей, опубліковано порівняно велику кількість робіт з цієї тематики і число збільшується з кожним днем [2]: 29 публікацій у 2000 році; 534 публікації в 2010 році; 22 097 публікацій у 2020 році.

У роботі [3] розглянуто основні напрями досліджень, які охоплюють архітектуру IoT. При створенні складних IoT-систем виникає досить велика кількість проблем [Ошибка! Источник ссылки не найден.], а саме: відсутність розроблених стандартів; уніфікованої архітектури систем тощо. В публікації

[5] розглядається стек технології та платформи для розробки IoT-систем та зазначаються напрями їхнього розвитку. У роботі [6] наводяться принципи функціонування та приклади IoT-систем.

Багато зусиль докладають вітчизняні дослідники для розвитку IoT-систем [7], а зокрема: Національний аерокосмічний університет ім. М.С. Жуковського «ХАІ»; Чернівецький національний університет; Східноукраїнський національний університет імені Володимира Даля та інші.

Мета статті – проаналізувати переваги та недоліки існуючих варіантів побудови IoT-систем, що дозволить обрати той, що найкраще відповідатиме поставленим завданням; виокремити ключові компоненти системи і сформулювати вимоги до неї; розробити платформу для моніторингу та керування IoT-пристроями. Платформа повинна мати можливість візуального програмування IoT-систем.

Викладення основного матеріалу. IoT-платформа – це набір компонентів, які забезпечують: взаємодію з додатками, віддалений збір даних з датчиків, безпечно підключення та управління пристроями. IoT-платформа забезпечує перевагу при створенні IoT-систем за рахунок надання інструментів розробки, що робить Інтернет речей простішим і дешевшим для кінцевих користувачів.

Повноцінна IoT-платформа має складатися з такого набору функціональних блоків [8]:

– *connectivity & normalization* (зв'язок і нормалізація) – об'єднує різні протоколи і формати даних в один «програмний» інтерфейс. Це необхідно для забезпечення можливості взаємодії з усіма пристроями і правильного читання даних. Просунуті пристрої надають інтерфейс прикладного програмування (англ. application programming interface, скор. API), який забезпечує уніфікацію інтерфейсу зв'язку з платформою. Однак дуже часто необхідно розробити і встановити програмні агенти, щоб IoT-платформа могла з'єднатися з пристроями;

– *device management* (управління пристроями) – забезпечує належне функціонування підключених IoT-пристроїв. Також виконує такі функції: ініціалізацію пристроїв, їхнє конфігурування, управління оновленням прошивок (firmware) та програмного забезпечення (software);

– *database* (база даних) – забезпечує зберігання даних і має задовольняти такі вимоги: збереження великих обсягів даних протягом тривалого часу, можливість збереження інформації від різних датчиків з різними форматами даних, обробка поточкових даних. Також забезпечувати масштабування та зберігання як структурованих, так і неструктурованих даних;

– *processing & action management* (обробка та управління діями) – забезпечує реакцію системи на отримані дані (події) від датчиків. Реакція на отримані дані виконується на основі певних правил, наприклад, виключення кондиціонера, коли люди залишають приміщення. Технічна реалізація часто здійснюється в формі правила «якщо це, то це» (If this, then that, скор. IFTTT);

– *analytics* (аналітика) – обробка даних для виявлення прихованих закономірностей, наприклад, виявлення рівня яскравості освітлення приміщення користувача залежно від часу доби і року. Цей функціонал може включати як найпростіші методи обробки, так і більш складні, наприклад, методи машинного навчання;

– *data visualization* (візуалізація даних) – забезпечення сприйняття інформації людиною. Візуалізація даних – це корисний інструмент кінцевого користувача IoT-додатка, тому що зорове сприйняття дозволяє виявити закономірності, які не можуть визначити чимало методів обробки даних;

– *додаткові інструменти* (additional tools) – набір інструментів, який дозволяє розробникам IoT-систем швидко створювати прототип системи, наприклад, використовуючи WYSIWYG-редактор (що бачиш, те й отримаєш; англ. what you see is what you get);

– *external interfaces* (зовнішні інтерфейси) – забезпечують інтеграцію платформи з іншими системами і додатками. Досягається це вбудованими API, комплектами розробки програмного забезпечення (англ. software development kit, скор. SDK) і шлюзами.

А що на сьогодні пропонує стандарт з розробки Інтернету речей [9]? У стандарті пропонується чотири основні структури підключення IoT-пристроїв у мережі:

1. Структура безпосереднього зв'язку між пристроями;
2. Структура зв'язку між пристроєм та хмарою;
3. Структура зв'язку пристроєм зі шлюзом;
4. Структура внутрішнього обміну даними.

Перша структура (рис. 1) дозволяє пристроям різних типів, що використовують один і той же протокол зв'язку, взаємодіяти між собою. Зазвичай використовується в системах домашньої автоматизації (розумний будинок, англ. smart house) для передачі невеликих пакетів даних між IoT-пристроями або розумними пристроями із відносно низькою швидкістю передачі даних. Це можуть бути лампи, датчик температури, датчик вологості, обігрівач тощо, що надсилає невелику кількість інформації один одному. Цей підхід може використовуватися для підключення розумних пристроїв, які можна носити, наприклад, фітнес-браслет, де дані не обов'язково мають передаватися у глобальну мережу.



Рис. 1. Безпосередній зв'язок між пристроями

Друга структура (рис. 2) дозволяє IoT-пристроєм підключатися безпосередньо до хмарної служби інтернету для обміну даними. Цей підхід використовує переваги існуючих систем зв'язку, таких як традиційні дротові з'єднання Ethernet або Wi-Fi, щоб встановити зв'язок між IoT-пристроєм та IP-мережею, яка в підсумку підключається до хмарної служби.

Хмарне підключення дозволяє користувачам отримати віддалений доступ до IoT-пристрою. Прикладом використання цього підходу є розумний будинок, коли користувач за допомогою мобільного телефону або комп'ютера може відстежувати стан пристроїв. Ще одним прикладом є передача даних у хмарну базу даних, де їх можна використати для аналізу, наприклад, споживаної енергії електроустаткуванням будинку.

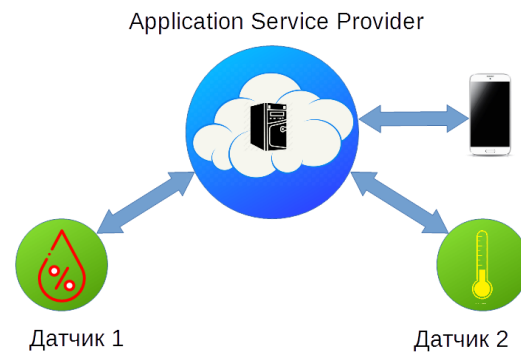


Рис. 2. Зв'язок між IoT-пристроєм та хмарою

Третя структура (рис. 3): IoT-пристрої підключаються до посередницького пристрою (шлюзу) для доступу до хмарної служби Інтернету. На локальному шлюзовому пристрої працює програмне забезпечення, яке є посередником між пристроєм та хмарною службою та виконує такі функції: безпечний обмін даними між IoT-пристроєм і хмарою; авторизацію, узгодження протоколів тощо.

У ролі локального шлюзу може бути смартфон з додатковим програмним забезпеченням, наприклад, для передачі інформації з фітнес-браслета, або спеціальне обладнання на базі мікроконтролера (міні-ПК, комп'ютера) для управління сенсорною мережею і забезпечення доступу до хмарної служби.

Четверта структура (рис. 4) по суті розширює структуру комунікації між IoT-пристроєм та хмарними службами. Друга та третя структури забезпечують IoT-пристроєм завантаження даних лише до однієї хмарної служби. Однак часто необхідна можливість експортувати та аналізувати дані в поєднанні з даними з інших джерел.

Прикладом використання такого підходу може бути система збору, обробки та зберігання споживаної енергії електроустаткуванням багатоквартирного будинку, вулиці, міста тощо.

При розробці IoT-систем існує три підходи: розробка системи з «нуля»; використання готових IoT-пристроїв з уніфікованими протоколами і розробка програмного забезпечення (написання модулів до існуючого програмного забезпечення); використання популярної платформи та розробка IoT-пристроїв з підтримкою API платформи. Останній підхід вимагає найменше часу для створення IoT-систем, тому такий підхід буде використовуватися також у нашій платформі.

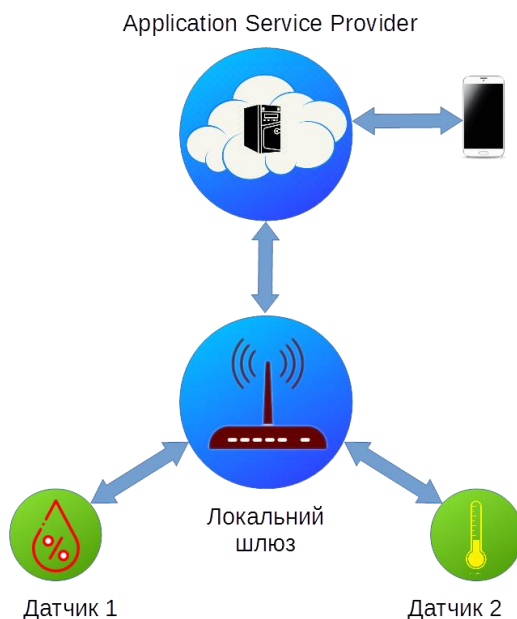


Рис. 3. Зв'язок між IoT-пристроєм та шлюзом

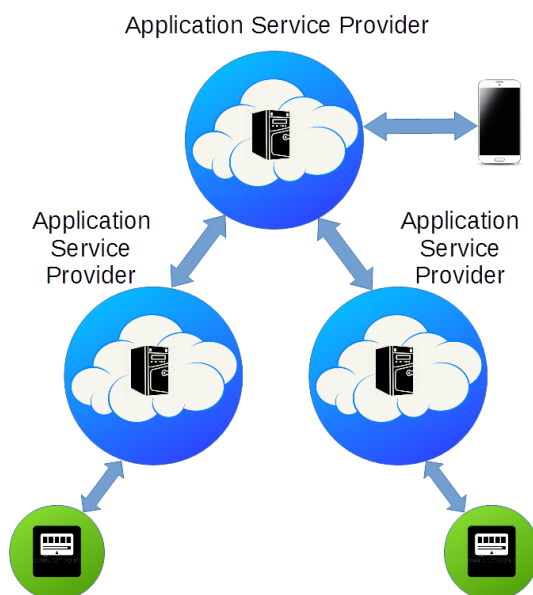


Рис. 4. Зв'язок між хмарними службами

Існує багато платформ для створення інфраструктури Інтернету речей. Проаналізуємо лише деякі, що можна розглядати як прототипи платформи, яка розробляється: Blynk [10]; RemoteXY [11]; Cayenne [12] тощо.

Blynk – платформа, що дозволяє у формі візуального конструктора створювати додатки для віддаленого керування IoT-пристроями. Вона дозволяє: створювати додаток, конфігурувати його; відображати стан датчиків; зберігати отримані дані тощо. Організація обміну між IoT-пристроєм та сервером виконується за допомогою: Wi-Fi, Bluetooth, Ethernet, USB (Serial), GSM тощо.

Платформа складається з трьох базових компонентів:

- Blynk App – додаток, призначений для створення графічного інтерфейсу для IoT-проектів за допомогою готових віджетів;
- Blynk Server – сервер, призначений для комунікації між смартфоном та IoT-пристроями. Сервер може бути як хмарний, так і локальний;
- Blynk Libraries – бібліотека для популярних апаратних платформ, яка дозволяє спростити роботу з API платформи.

RemoteXY – платформа для розробки IoT-систем з можливістю управління IoT-пристроями зі смартфона або планшета. До складу системи входять:

- вебредактор створення графічного інтерфейсу для моніторингу та управління IoT-пристроями;
- мобільний додаток.

Відмітні особливості:

- конфігурації графічного інтерфейсу зберігаються в IoT-пристрої;
- з одного мобільного додатка можна управляти усіма IoT-пристроями.

Підтримуються такі способи зв'язку між IoT-пристроєм та смартфоном за допомогою: хмарного сервера; Wi-Fi в режимі клієнта і точки доступу; Bluetooth; Ethernet тощо.

На відміну від *Blupk*, мобільний додаток *RemoteXY* може обмінюватися пакетами як через хмарний сервер, так і за допомогою локального сервера, який встановлений на IoT-пристрої.

Cayenne – платформа, яка схожа на *Blupk*, але передбачає підхід з мінімальною кількістю програмування на стороні IoT-пристрою (має механізм тригерів та подій).

Основні недоліки розглянутих платформ – це відсутність можливості створення сценаріїв автоматичного управління системою. Додатково були розглянуті платформи [13, 14]: *ThingSpeak*, *AgoControl*, *OpenHab*.

З огляду на викладене вище було розроблено архітектуру IoT-платформи (рис. 5).

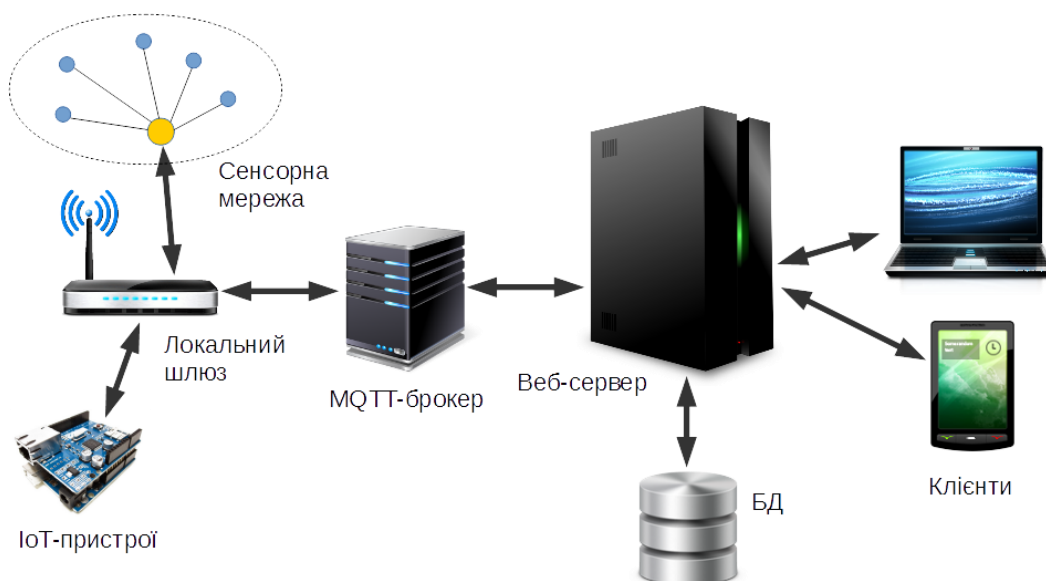


Рис. 5. Архітектура платформи

На нижньому шарі архітектури будуть розміщені IoT-пристрої (розумні пристрої) і/або сенсорні мережі. Сенсорна мережа – розподілена мережа, що самоорганізується та складається із безлічі датчиків і виконавчих пристроїв, які об'єднані між собою за допомогою каналу зв'язку (найчастіше радіоканалу). Нижній шар взаємодіє з локальним шлюзом. Як протоколи на цьому шарі використовуються такі протоколи фізичного рівня, як [15]: Z-Wave, ZigBee, Wi-Fi, Bluetooth тощо.

Локальний шлюз вирішує проблему сумісності протоколів як між різними пристроями, так і між пристроями та інтернетом. Завдяки шлюзу окремі пристрої можуть взаємодіяти безпосередньо між собою без виходу в інтернет.

Локальний шлюз в архітектурі взаємодіє не безпосередньо з сервером, а використовується MQTT-брокер. MQTT – це протокол обміну повідомленнями, який використовує шаблон проектування видавець-передплатник (pub/sub). Система зв'язку, яка побудована на MQTT, складається з видавця, передплатника і сервера-брокера. Видавець формує повідомлення, а передплатник отримує ті з них, на які підписаний. Сервер-брокер необхідний для забезпечення доставки повідомлень. Кожен клієнт MQTT-брокера може бути як видавцем, так і передплатником. Таким чином, локальний шлюз, як і сервер, є клієнтами MQTT-брокера. У майбутньому, можливо, протокол буде змінений, але на цьому етапі використовується протокол MQTT.

На стороні сервера використовується мова програмування Node.js, TypeScript і фреймворк LoopBack.

LoopBack – це фреймворк з відкритим кодом, написаний на мові програмування TypeScript та Node.js, легко розширюваний, створений на базі фреймворка Express. Він дозволяє швидко розробляти API та мікросервіси.

Як базу даних було використано NoSQL базу даних – MongoDB.

API розробленої платформи дозволяє отримати доступ до периферії мікроконтролерів IoT-пристроїв.

Для створення клієнтського інтерфейсу було використано такий стек технологій: CSS, HTML, Javascript, React, Material-UI. Як протокол для взаємодії клієнта з сервером використано WebSocket.

Для завантаження стартової сторінки користувачу потрібно авторизуватися. Після виконання авторизації система відобразить інтерфейс користувача згідно з його роллю та конфігуруванням системи (рис. 6). Якщо пристрій не встановлений, то потрібно виконати деякі дії, а саме обрати пристрій зі списку, який буде запропоновано користувачу системою (рис. 7).

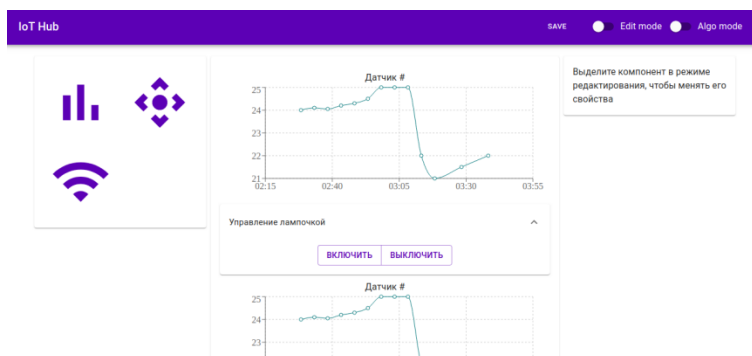


Рис. 6. Інтерфейс користувача

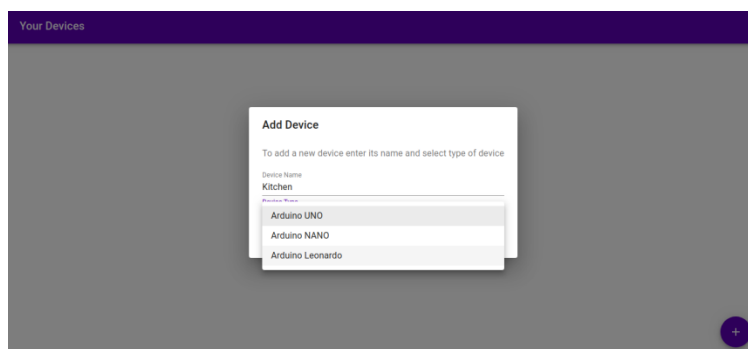


Рис. 7. Додавання пристрою користувачем

Користувач має можливість підключати декілька пристроїв, а також має редагувати та видаляти пристрій з системи, натиснувши відповідну кнопку. За замовчуванням редагування заблоковано. Для розблокування (рис. 6), потрібно вибрати режим «Edit mode».

Платформа дозволяє створювати сценарії роботи пристроїв (рис. 8). Для цього було використано Blockly [14].



Рис. 8. Сторінка створення сценарію

Blockly підтримує основні концепції програмування. Вона надає: умовні оператори, блоки циклів з постумовою та передумовою тощо, але Blockly не надає повного словника блоків або середовища виконання. Розробникам необхідно інтегрувати Blockly у вебдодаток, створити свій словниковий запас і вирішити, як буде працювати згенерований код.

Висновки та перспективи подальших досліджень. У роботі було проаналізовано переваги та недоліки існуючих варіантів побудови IoT-систем. Запропоновано архітектуру платформи, яка дозволяє взаємодіяти користувальницьким додаткам з IoT-пристроями.

У результаті було отримано діючий прототип IoT-системи, що демонструє принцип роботи цієї платформи. Основу системи становить сервер, який взаємодіє з клієнтами та IoT-пристроями. Однак взаємодія з пристроями сервером виконується за допомогою MQTT-брокера. Взаємодія клієнтів з сервером виконується за протоколом WebSocket. Програмне забезпечення для сервера створено за допомогою мови програмування Node.js, TypeScript і фреймворка LoopBack. Для створення клієнтського інтерфейсу було використано такий стек технологій: CSS, HTML, Javascript, React, Material-UI.

Розроблена платформа є дуже гнучкою та дозволяє: підключати безліч різноманітних пристроїв; конфігурувати пристрої через браузер; створювати сценарії для керування системою за допомогою візуального програмування.

Список використаної літератури:

1. Norman J. Invents the Term «The Internet of Things» / J.Norman, K.Ashton [Електронний ресурс]. – Режим доступу : <https://www.historyofinformation.com/detail.php?id=3411>.
2. Internet of Things [Електронний ресурс]. – Режим доступу : <https://academic.microsoft.com/topic/81860439/publication/search?q=Internet of Things>.
3. The 10 Research Topics in the Internet of Things : 2020 IEEE 6th International Conference on Collaboration and Internet Computing (Virtual Conference, 1–3.12.2020) / Wei Emma Zhang, Quan Z. Sheng, Adnan Mahmood et al. – Danvers, 2020. – P. 34–43.
4. Sanjay S. The Internet of Things: Roadmap to a Connected World / S.Sanjay. – 2016 [Електронний ресурс]. – Режим доступу : <https://www.technologyreview.com/2016/03/11/108875/the-internet-of-things-roadmap-to-a-connected-world/>.
5. Wortmann F. Internet of Things / F.Wortmann, K.Flüchter // Bus Inf Syst Eng. – 2015. – Vol. 57. – P. 221–224. DOI: 10.1007/s12599-015-0383-3.
6. Marco S. Internet of Things with ESP8266 / S.Marco. – 2016. – 266 p.
7. Проект Erasmus + Internet of Things: Emerging Curriculum for Industry and Human Applications ALIOT (reference number 573818-EPP-1-2016-1-UK-EPPKA2-SVHE-JP) (Інтернет речей: нова навчальна програма для потреб промисловості та суспільства) [Електронний ресурс]. – Режим доступу : <https://zp.edu.ua/projekt-erasmus-internet-things-emerging-curriculum-industry-and-human-applications-aliot-reference>.
8. IOT PLATFORMS [Електронний ресурс]. – Режим доступу : <http://www.iot-analytics.com/wp/wp-content/uploads/2016/01/White-paper-IoT-platforms-The-central-backbone-for-the-Internet-of-Things-Nov-2015-vfi5.pdf>.
9. Architectural Considerations in Smart Object Networking [Електронний ресурс]. – Режим доступу : <https://datatracker.ietf.org/doc/html/rfc7452>.
10. How Blynk Works [Електронний ресурс]. – Режим доступу : <http://docs.blynk.cc/>.
11. RemoteXY. Как это работает? [Електронний ресурс]. – Режим доступу : <https://remotexy.com/ru/help/>.
12. Getting Started [Електронний ресурс]. – Режим доступу : <https://developers.mydevices.com/cayenne/docs/getting-started/>.
13. Петросян А.Р. Організація сценаріїв автоматизації в IoT-системах : Тези III Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», 26–27 листопада 2020 р. / А.Р. Петросян, Р.В. Петросян. – Житомир : Житомирська політехніка, 2020. – С. 71–74.
14. Петросян А.Р. Аналіз програмних засобів візуального програмування для застосування в IoT. Розвиток освіти, науки та бізнесу: результати 2020 : Тези доп. міжнар. науково-практичної інтернет-конференції, 3–4 грудня 2020 р. / А.Р. Петросян, Р.В. Петросян. – Україна, Дніпро, 2020. – Т. 2. – С. 237–240.
15. Андриєвич В.М. Огляд протоколів управління «Розумним будинком» : Тези доповідей III Всеукраїнської науково-практичної інтернет-конференції здобувачів вищої освіти і молодих учених «Інформаційно-комп'ютерні технології: стан, досягнення та перспективи розвитку», м. Житомир, 26–27 листопада 2020 р. / В.М. Андриєвич, Р.В. Петросян. – Житомир : Житомирська політехніка, 2020. – С. 39–40.

References:

1. Norman, J. and Ashton, K., *Invents the Term «The Internet of Things»*, [Online], available at: <https://www.historyofinformation.com/detail.php?id=3411>
2. Internet of Things, [Online], available at: <https://academic.microsoft.com/topic/81860439/publication/search?q=Internet of Things>
3. Zhang, Wei Emma, Sheng, Quan Z. and Mahmood, Adnan et al. (2020), «The 10 Research Topics in the Internet of Things», *2020 IEEE 6th International Conference on Collaboration and Internet Computing (Virtual Conference, 1–3.12.2020)*, Danvers, pp. 34–43.
4. Sanjay, S. (2016), *The Internet of Things: Roadmap to a Connected World*, [Online], available at: <https://www.technologyreview.com/2016/03/11/108875/the-internet-of-things-roadmap-to-a-connected-world/>
5. Wortmann, F. and Flüchter, K. (2015), «Internet of Things», *Bus Inf Syst Eng*, Vol. 57, pp. 221–224, doi: 10.1007/s12599-015-0383-3.
6. Marco, S. (2016), *Internet of Things with ESP8266*, 266 p.

7. Proekt Erasmus + Internet of Things: Emerging Curriculum for Industry and Human Applications ALIOT (reference number 573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP, [Online], available at: <https://zp.edu.ua/proekt-erasmus-internet-things-emerging-curriculum-industry-and-human-applications-aliot-reference>
8. IOT PLATFORMS, [Online], available at: <http://www.iot-analytics.com/wp/wp-content/uploads/2016/01/White-paper-IoT-platforms-The-central-backbone-for-the-Internet-of-Things-Nov-2015-vfi5.pdf>
9. Architectural Considerations in Smart Object Networking, [Online], available at: <https://datatracker.ietf.org/doc/html/rfc7452>
10. How Blynk Works, [Online], available at: <http://docs.blynk.cc/>
11. RemoteXY. Kak eto rabotaet?, [Online], available at: <https://remotexy.com/ru/help/>
12. Getting Started, [Online], available at: <https://developers.mydevices.com/cayenne/docs/getting-started/>
13. Petrosjan, A.R. and Petrosjan, R.V. (2020), «Organizacija scenarii'v avtomatyzacii' v IoT-systemah», *Tezy III Vseukrai'ns'koi' naukovo-tehnichnoi' konferencii' «Komp'juterni tehnologii': innovacii', problemy, rishennja», 26–27 lystopada 2020 r.*, Zhytomyrs'ka politehnika, Zhytomyr, pp. 71–74, Ukraine.
14. Petrosjan, A.R. and Petrosjan, R.V. (2020), «Analiz programnyh zasobiv vizual'nogo programuvannja dlja zastosuvannja v IoT. Rozvytok osvity, nauky ta biznesu: rezul'taty 2020», *Tezy dop. mizhnar. naukovo-praktychnoi' internet-konferencii', 3–4 grudnja 2020 r.*, Ukrai'na, Dnipro, Vol. 2, pp. 237–240.
15. Andrijevych, V.M. and Petrosjan, R.V. (2020), «Ogljad protokoliv upravlinnja “Rozumnym budynkom”», *Tezy dopovidej III Vseukrai'ns'koi' naukovo-praktychnoi' internet-konferencii' zdobuvachiv vyshhoi' osvity i molodyh uchenyh «Informacijno-komp'juterni tehnologii': stan, dosjagnennja ta perspektyvy rozvytku», m. Zhytomyr, 26–27 lystopada 2020 r.*, Zhytomyrs'ka politehnika, Zhytomyr, pp. 39–40, Ukraine.

Петросян Арсен Русланович – студент 1-го курсу магістратури Державного університету «Житомирська політехніка».

<http://orcid.org/0000-0003-0960-8461>.

Наукові інтереси:

- комп'ютерні науки;
- Інтернет речей;
- вебпрограмування та вебтехнології;
- олімпіадне програмування.

Петросян Руслан Валерікович – старший викладач кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

<http://orcid.org/0000-0002-0388-8821>.

Наукові інтереси:

- цифрова обробка;
- штучний інтелект;
- математичне моделювання;
- комп'ютерні системи спеціального призначення;
- вебтехнології.

Колос Катерина Ростиславівна – доктор педагогічних наук, професор кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

<http://orcid.org/0000-0002-1038-8569>.

Наукові інтереси:

- комп'ютерні науки;
- комп'ютерно орієнтоване навчальне середовище.

Стаття надійшла до редакції 19.05.2021.