

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра електроніки, загальної та прикладної фізики

«До захисту допущено»
Завідувач кафедри

_____ Іван ПРОЦЕНКО
_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 171 Електроніка освітньо-професійної програми «Електронні інформаційні системи»

на тему: **Мікроконтролерні платформи в IoT мережах**

Здобувача групи ЕП-91 _____ Кужеля Віталія Васильовича_____

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Віталій Кужель_____

Керівник ст.викладач, канд. ф.-м.н. Костянтин Тищенко _____

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра електроніки, загальної та прикладної фізики
Спеціальність 171 – Електроніка, освітньо-професійна програма
«Електронні інформаційні системи»

ЗАТВЕРДЖУЮ
Зав. кафедри ЕЗПФ

І.Ю.Проценко

«29» травня 2023 року

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА
Кужеля Віталія Васильовича

1. Тема роботи: **Мікроконтролерні платформи в IoT мережах**

затверджена наказом по університету від «15» травня 2023 р., № 0499-VI

2. Термін здачі студентом закінченої роботи: 09 червня 2023 року

3. Вихідні дані до роботи (актуальність, мета)

Мікроконторолерні інформаційні системи моніторингу та керування є основною складовою інтернету речей. Забезпечення функцій керування системою типу IoT може бути реалізована багатьма способами, один із яких є написання власного програмного забезпечення для контролера керування вузлом окремим вузлом мережі. Такий підхід дозволяє конфігурувати поведінку системи, залежно від потреб кінцевого користувача та супутніх факторів. Також використання мікроконтролерних платформ, що випускаються серійно, дозволяє уніфікувати кінцеву систему, та забезпечити сумісність (як програмну, так і апаратну) з розповсюдженими хмарними сервісами.

Метою роботи полягає в розробці схеми та програмного забезпечення IoT системи на базі платформи Arduino для автоматизації процесів у межах однієї кімнати, для контролю та керування параметрами мікроклімату та автоматизації окремих процесів.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що належить їх розробити)

1. Архітектура систем типу Інтернет речей
2. Хмарні сервіси для інтеграції IoT мереж

3. Мікроконтролерні платформи для IoT
4. Розробка програмного забезпечення вузла мережі IoT.
5. Висновки.

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

Слайди № 1-2 – Загальна інформація

Слайди № 3-6 – Архітектура IoT -мереж, особливості реалізації розподілених мереж

Слайди № 7-9 – Мікроконтролерні платформи для побудови вузлів IoT-систем

Слайди № 10 – Методика експерименту

Слайди № 11-13 – Експериментальні результати з розробки програмного забезпечення IoT-системи

Слайд № 14 – Висновки

6. Дата видачі завдання 30.05.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи магістрів	Термін виконання етапів роботи	Примітка
1.	Аналіз літературних даних	до 02.06.2023 р.	<i>вик.</i>
2.	Проведення експерименту, моделювання, розрахунків, обробка результатів	до 06.06.2023 р.	<i>вик.</i>
3.	Оформлення тексту кваліфікаційної роботи.	до 09.06.2023 р.	<i>вик.</i>
4.	Попередній захист роботи	12.06.2023 р., 10 ⁰⁵ (дистанційно)	<i>вик.</i>
5.	Захист роботи в екзаменаційній комісії	20-23.06.2023 р., 10 ⁰⁵ (дистанційно)	<i>вик.</i>

Здобувач вищої освіти

В.В. Кужель

Керівник роботи

К.В. Тищенко

АНОТАЦІЯ

Кваліфікаційна робота викладена на 36 сторінках, зокрема, містить 9 рисунків, 2 таблиці, список використаних джерел із 21 найменувань.

Актуальність теми: сьогодні широкого поширення набули системи типу Інтернет речей, вони є не лише системами автоматизованого керування процесами, а й інтелектуальною мережею пристроїв, які на умовах рівних прав та паритетів взаємодіють між собою, з людиною та навколишнім середовищем. Такі системи реалізуються з урахуванням того, що у кожного її учасника є так званий віртуальний «агент», який представляє його інтереси у віртуальному середовищі. Сьогодні на ринку є багато мікроконтролерних та мікропроцесорних платформ, які дозволяють створювати системи IoT мережі, кожна із них має набір характеристик, які забезпечують її найефективніше використання у певній сфері.

Мікроконтролерна платформа Arduino може бути використана для побудови IoT мереж, оскільки вона забезпечує підключення великої кількості сенсорів навколишнього середовища та може керувати підключенням навантаження за допомогою силових модулів розширення, а також дозволяє реалізувати взаємодію з системами і мережами подібного призначення.

Мета кваліфікаційної роботи бакалавра полягає в розробці схеми та програмного забезпечення IoT мережі на базі платформи Arduino для автоматизації процесів у межах однієї кімнати.

Під час виконання роботи використовували онлайн середовище проєктування електронних схем tinkercad, яке містить віртуальні аналоги Arduino, датчиків фізичних величин та базові електронні компоненти.

У результаті проведених проєктування та розробки отримано схему та програмне забезпечення IoT системи яка керує освітленням, температурним режимом, відкриттям шухляд а також сигналізує про наявність газу в приміщенні.

Ключові слова: інтернет речей, шлюз, мікроконтролер, arduino, tinkercad

ЗМІСТ

	с.
ВСТУП	6
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ІНТЕРНЕТУ РЕЧЕЙ	7
1.1 Поняття «Інтернету речей»	7
1.2 Хмарні сервіси для керування IoT	9
1.3 Використання шлюзів в Інтернеті речей	12
1.4 Архітектура IoT	14
1.5 Принципи роботи Інтернету речей	16
1.6 Мікроконтролерні платформи для IoT.....	18
РОЗДІЛ 2. РОЗРОБКА ІОТ СИСТЕМ НА ОСНОВІ МІКРОКОНТРОЛЕРА	24
РОЗДІЛ 3. КОПОНУВАННЯ ТА ПРОГРАМУВАННЯ ІОТ СИСТЕМ НА ОСНОВІ ARDUINO	28
3.1. Створення функціональної схеми IoT системи.....	28
3.2. Створення програмного забезпечення IoT системи	29
ВИСНОВКИ	34
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	35

ВСТУП

Комфорт сучасної людини залежить від незліченної кількості чинників які несуть значний вплив на повсякденне життя. Одним з таких факторів є використання різних електронних пристроїв. При цьому широке застосування побутових та портативних гаджетів стимулює розвиток безліч галузей промисловості. З кожним днем, поступово, світ еволюціонує, і все більше чуємо про такі винаходи, як «розумний будинок» та «розумний холодильник».

Інтернет речей (IoT) – це новочасний етап довготривалої революції в обчислювальних і комунікаційних системах. Його обсяг, гетерогенність і вплив на повсякденне життя, ділову активність і державне управління перевершили всі попередні технологічні досягнення в історії. IoT - це термін, який використовується для опису набору взаємопідключених інтелектуальних пристроїв, починаючи від побутових приладів і закінчуючи невеликими датчиками. Інтеграція мобільних приймачів малого радіусу дії в різні пристрої та предмети повсякденного вжитку породжує нові форми контактування та взаємодії людини з предметами та безпосередньо контакту поміж пристроями. Станом на сьогодні інтернет поєднує мільярди побутових та промислових речей використовуючи хмарні системи. Ці пристрої передаючи інформацію датчиків, реагують відповідним чином на навколишнє середовище. Для більших системи, таких як великі підприємства або ж, крім того навіть міста, створюється спільне середовище керування системою, яке само модифікується [1].

"Речі" в IoT – це глибоко вбудовані пристрої. Вони характеризуються вузькою смугою пропускання, менш повторюваним збором даних, низьким енергоспоживанням тощо. Вони також мають низьке споживання ресурсів. Такі гаджети можуть обмінюватися інформацією один з одним і передавати відомості через інтерфейси. Деякі вбудовані пристрої Інтернет речей, такі як камери безпеки високої чіткості, відеотелефонія VoIP тощо, потребують високої пропускної здатності для роботи.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ІНТЕРНЕТУ РЕЧЕЙ

1.1 Поняття «Інтернету речей»

Термін "Інтернет речей" (IoT) був введений в 1999 році Кевіном Ештоном, засновником центру автоматичної ідентифікації в Університеті штату Массачусетс. Дуже велика кількість інтерпретацій цього терміну наведено в науковій літературі, де можна спостерігати розбіжності між визначеннями. Далі наведено тлумачення терміну «Інтернет речей», яке запропонувала компанія Gartner (саме ця компанія придумала термін ERP).

«Інтернет речей - це мережа фізичних об'єктів, які мають вбудовані технології, що дозволяють здійснювати взаємодію з зовнішнім середовищем, передавати відомості про свій стан і приймати дані ззовні» [2].

ITU-D, відділення Міжнародного союзу електрозв'язку, що відповідає за стандарти зв'язку, випустило рекомендацію під назвою «Огляд Інтернету речей» (Y.2060) [12]. Цей документ містить визначення, які пояснюють включені складові в IoT, зокрема:

Інтернет речей (IoT): всесвітня інфраструктура, яка дозволяє надавати більш складні послуги шляхом з'єднання (фізичних і віртуальних) об'єктів на основі існуючих і розроблюваних інформаційних і комунікаційних технологій, які є функціонально сумісними.

Річ: у контексті IoT це означає фізичний об'єкт у реальному світі або віртуальний об'єкт у цифровому світі, який можна ідентифікувати та інтегрувати в комунікаційні мережі.

Пристрій: у контексті IoT це апаратне забезпечення, яке має обов'язкові можливості зв'язку та додаткові можливості для вимірювання, запуску, а також введення, зберігання та обробки даних.

На основі досліджень наукових джерел можна надати власне визначення поняття «Інтернету речей».

Інтернет речей (Internet of Things, IoT) – це взаємний зв'язок між

фізичними пристроями, які мають вбудовані датчики, виконавчі механізми та програмне забезпечення, яке взаємодіє з навколишнім середовищем, що утворюють, або є частиною мережевої системи [3, 4, 5]. Ця система може передавати дані між комп'ютерним обладнанням та фізичними об'єктами в автоматизованому режимі, використовуючи стандартні протоколи передачі даних в мережі Інтернет.

Основою Інтернету речей є можливість з'єднувати різні предмети та гаджети, що користуються люди. Інтернет речей забезпечує взаємозв'язок, передачу даних та управління підключеними пристроями. Для цього всі об'єкти повинні бути оснащені датчиками та засобами передачі даних використовуючи Інтернет мережу (наприклад, WiFi або Ethernet адаптерами), що надає дозвіл на контроль їх роботи. Наявність керованих виконавчих пристроїв не є обов'язковою, але значно збільшує можливості системи, оскільки змінює режим роботи обладнання без втручання людини.

На сьогодні, Інтернет складається з багатьох комп'ютерних мереж, включаючи корпоративні, наукові, урядові та домашні мережі. Ці мережі з'єднані між собою за допомогою протоколу IP, незважаючи на різницю в архітектурі та топології. Кожній особі або групі осіб, задіяних у мережі, призначається IP-адреса, яка може бути фіксованою або тимчасовою (динамічною). Прикладом пристроїв Інтернет речей можуть бути: камера відеоспостереження, комп'ютер, годинник, машина, холодильник, літак та інші пристрої побуту та повсякденного використання людиною.

Отже, основними компонентами системи IoT є:

- Фізичні предмети, які мають датчики та пристрої передачі даних за допомогою мережі Інтернет;
- Централізовані або децентралізовані платформи, що забезпечують рішення завдань захисту, обробки даних та управління підключеними пристроями.

Для розробки початкового компонента необхідно налаштувати датчики та мережеві адаптери на пристрої. Основні проблеми в цьому процесі пов'язані

з фізичним встановленням обладнання, забезпеченням доступу адаптерів до живлення з відповідними функціями, визначенням відстані до обладнання та його переміщенням, якщо це необхідно під час роботи. В даний час індустрія регулярно збільшує різноманітність продуктів, які містять вбудовані датчики та мережеві адаптери, що значно полегшує їх використання в IoT.

Для створення другого компонента необхідно задіяти серверне обладнання, бази даних та програмне забезпечення, яке полегшує передачу даних через різні протоколи, такі як HTTP, HTTPS, MQTT тощо. Вимоги конкретної платформи встановлюють перелік функціональних можливостей, технічних характеристик обладнання та складу програмного забезпечення.

1.2 Хмарні сервіси для керування IoT

Сьогодні поширеним рішенням є використання хмарних сервісів для керування екосистемою IoT. У загальному випадку схематично таке рішення зображено на рисунку 1.1. При такій архітектурі мережі IoT пристрої, окрім того, що взаємодіють між собою, через шлюз підключаються до мережі Інтернет, і можуть керуватись віддаленими пристроями. Така структура сьогодні вважається класичною, а при підключенні її до хмарного сервісу можна отримати велику кількість додаткових можливостей. Хмарний сервіс починає виконувати ключову роль в IoT мережі, по-перше він надійно зберігає дані та налаштування пристроїв, по-друге виконує математичні обчислення. Перший аспект важливий ще і з тієї причини, що хмара виступає універсальною базою знань, які можуть бути примінені до усіх пристроїв конкретного типу, тим самим наділяючи розумні пристрої новими можливостями без участі людини, тобто реалізується автоматична система оновлень програмного забезпечення і функцій. Другий аспект важливий із тієї причини, що, зазвичай, розумні пристрої спроектовані максимально енергоефективними, і як наслідок – не мають технічної можливості виконувати аналіз даних і складні математичні розрахунки. Тут на допомогу

приходить хмарний сервіс, який і забезпечує обробку та аналіз інформації.

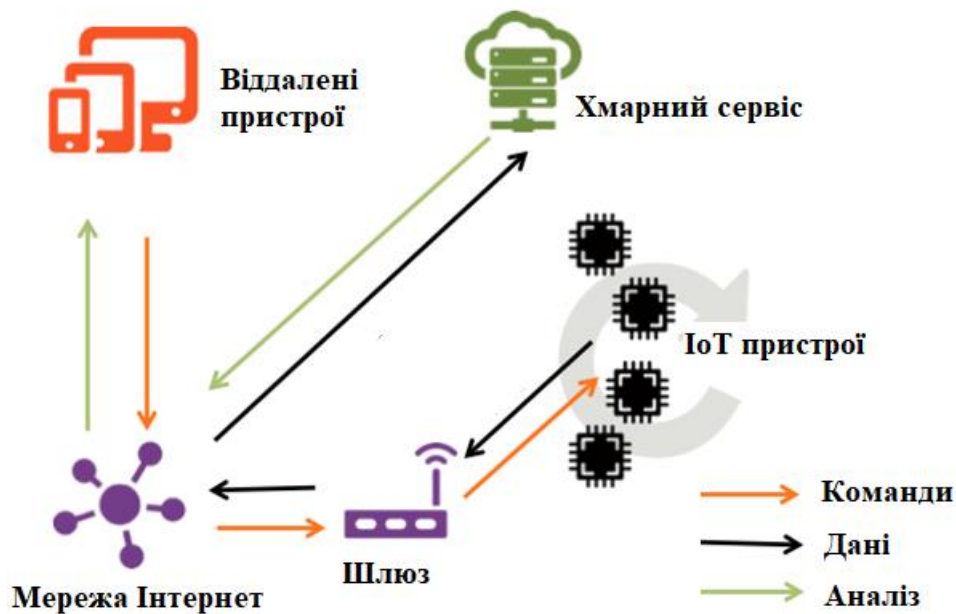


Рисунок 1.1 – IoT система з хмарним середовищем

Сьогодні багато компаній надають послуги хмарного IoT-сервісу, забезпечуючи роботу як з пристроями власного виробництва, так і зі сторонніми, які підтримують можливість хмарної комунікації. Компанії, які надають хмарні платформи IoT:

1. Arduino IoT Cloud [6]. Ця платформа пропонує веб-інтерфейс і комплексне рішення для кодування, конфігурації, завантаження та візуалізації, що дозволяє створювати проекти IoT. Основні функції цієї платформи полягають у моніторингу значень даних за допомогою інформаційної панелі, синхронізації змінних між пристроями, планування завдань у визначений часовий проміжок, підвантаження по повітрю (OTA), яке оновлює код програмного забезпечення пристрою без необхідності підключення до комп'ютер, інтеграція зі сторонніми службами через веб-хуки та керування правами доступу користувачів.

2. Google Cloud IoT [7] — це рішення Інтернету речей (IoT), побудоване на хмарній платформі Google, яке пропонує комплексний набір інструментів для підключення, обробки, зберігання та аналізу даних. Він поставляється з

інтегрованим набором технологій Google Cloud для обчислень. Хоча це складне рішення, його складність може стати проблемою для швидкого створення прототипів. Платформа в основному призначена для використання в різних програмах, включаючи прогнозне технічне обслуговування, відстеження активів у реальному часі, логістику та управління ланцюгами поставок, а також розробку розумних будинків і міст.

3. Платформа AWS [8] IoT від Amazon створена для управління та підключення величезної кількості пристроїв. Її мета – збирати, зберігати та аналізувати дані IoT для підтримки споживчих, промислових і комерційних додатків. Платформа пропонує такі функції, як контроль доступу, шифрування, масштабованість та інтеграція з іншими хмарними службами, які надає Amazon.

4. PTC ThingWorx [9] – це платформа, яка зосереджена на широкомасштабному впровадженні промислового Інтернету речей і пропонує спеціалізовані рішення, що покращують основні можливості Інтернету речей, як-от платформи доповненої реальності та керування життєвим циклом продукту. Він зазвичай використовується для виконання таких завдань, як розповсюдження товарів як послуг, віддалений моніторинг промислових об'єктів, створення цифрових робочих інструкцій і моніторинг параметрів у режимі реального часу.

5. Платформа Каа IoT [10] має на меті зробити впровадження IoT легким і швидким. Він поставляється з повним набором функцій Інтернету речей, включаючи аналітику Інтернету речей, і не потребує інтеграції додаткових модулів або служб. Каа є універсальним і може використовуватися для різних випадків використання IoT, зосереджуючись на простоті самообслуговування та привабливим для невеликих стартапів і великих компаній, які ще не готові до широкомасштабного впровадження IoT. Каа використовується в різних галузях, таких як автомобільний транспорт, охорона здоров'я, промисловий Інтернет речей, логістика та розумна роздрібна торгівля.

6. Платформа Azure IoT [11] від Microsoft є частиною їхнього хмарного

сервісу, що надає комплексне рішення, яке включає пристрої, інструменти, аналітику даних і функції безпеки, які допомагають реалізувати проекти IoT. Azure IoT складається з кількох компонентів і вимагає певного рівня досвіду, щоб налаштувати його відповідно до вимог компанії. Він використовується в різних галузях, таких як автомобільна промисловість, енергетика, охорона здоров'я, роздрібна торгівля, транспорт і логістика.

Більшість перерахованих платформ надають безкоштовне використання, хоча й з певними обмеженнями щодо кількості пристроїв і зберігання даних. Якщо ви хочете підвищити ці параметри, вам потрібно буде перейти на платну версію платформи. Крім того, у певних випадках можуть існувати обмеження безпеки щодо обробки даних третіми компаніями.

1.3 Використання шлюзів в Інтернеті речей

З експоненціальним зростанням датчиків і пристроїв у всьому світі та широким впровадженням хмарної інфраструктури підключення їх до хмари становить серйозну проблему. Отже, зростає потреба в проміжній лінії зв'язку, яка не тільки полегшує підключення пристрою до звичайного стеку TCP/IP, але й оптимізує використання пропускну здатності шляхом обробки даних перед передачею.

Шлюз IoT служить для цієї мети з таких причин:

- Необхідно перетворити дані, що передаються через такі протоколи, як Bluetooth, ZigBee, 6LoWPAN та інші, у стандартні Інтернет-пакети, а не покладатися виключно на стек TCP/IP.

- Необхідно підтримувати кілька локальних дротових або бездротових мереж IoT.

- Надійний зв'язок із пристроями є важливим, навіть якщо немає підключення до хмари. Буферизація даних від неінтелектуальних датчиків дозволяє пізніше передавати їх на сервер після відновлення з'єднання.

– Потрібне ефективне використання трафіку, що досягається шляхом агрегування та пакетування значних обсягів основного мережевого трафіку, створеного мільярдами пристроїв IoT. Збір різних показників у файли перед відправкою на сервер.

– Подолання проблем із відстанню між датчиками та обладнаними датчиками платами, розташованими у віддалених місцях, що потребує низького енергоспоживання, що робить прямий зв'язок із хмарою непрактичним.

– Забезпечення конфіденційності та безпеки даних під час спілкування з хмарою.

– Виконання початкової обробки даних для усунення затримок передавання, проблем зв'язку з сервером і обмеженої обчислювальної потужності енергоефективних кінцевих пристроїв.

На рисунку 1.2 показано спосіб використання шлюзу в IoT мережах. Тобто його роль – це проміжна ланка між певною локальною екосистемою і мережею Інтернет та хмарними сервісами. Шлюз забезпечує підключення пристроїв за IoT протоколами, виконує попередню обробку даних і передає їх в мережу інтернет. Такі самі перетворення він виконує і у зворотньому напрямку.

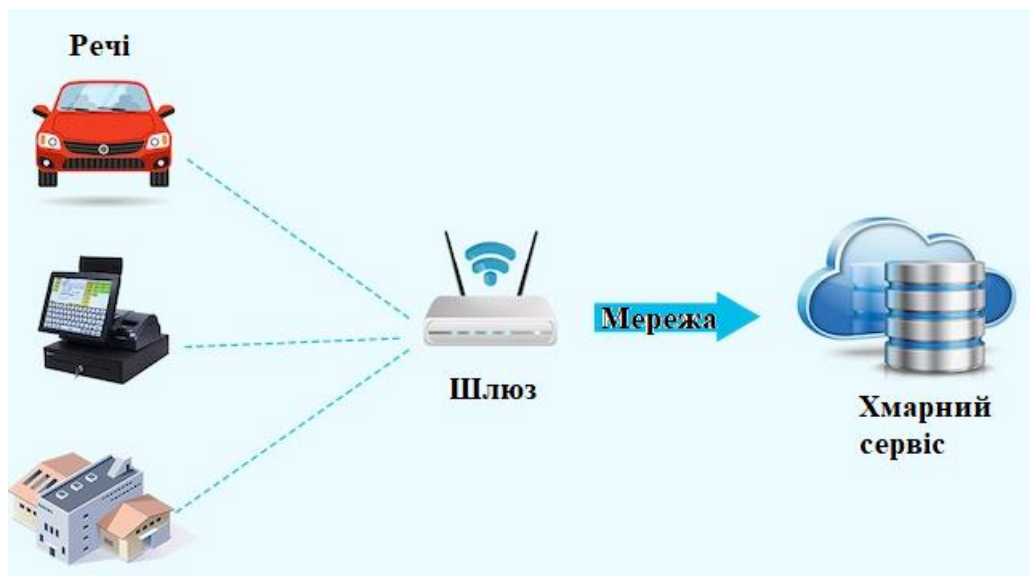


Рисунок 1.2 – Використання шлюзу в IoT мережах

Враховуючи сказане, шлюзи є потужними пристроями, які потребують фіксованого підключення до живлення. Вони є основою для систем реального часу, здатних швидко реагувати на певні дії управління та критичні зміни даних датчиків. Крім того, шлюзи полегшують керування компонентами рішення IoT як локально, так і віддалено.

1.4 Архітектура IoT

Ефективна комунікація є критично важливим аспектом будь-якого IoT-проекту, але, незважаючи на велику кількість доступних комунікаційних протоколів, кожен із них має певні недоліки, які роблять їх не оптимальними для додатків IoT. Основні проблеми включають споживання енергії, діапазон покриття та пропускну здатність. Більшість технологій радіозв'язку, таких як Zigbee, BLE, WiFi та інші, мають обмежений радіус дії, тоді як інші, такі як 3G і LTE, споживають значну кількість енергії та можуть не мати гарантованого покриття, особливо в країнах, що розвиваються.

Для цих цілей найбільш оптимальним і збалансованим рішенням могло б стати використання технології LoRaWan. LoRa (Long Range) – це запатентована бездротова технологія, яка поєднує в собі наднизьке енергоспоживання та ефективний зв'язок на великій відстані. Слід зазначити, що зона покриття сильно залежить від навколишнього середовища та можливих перешкод (LOS або N-LOS).

Роль архітектури IoT у тому, щоб:

- Запропонувати мережевим адміністраторам контрольний список для оцінки функціональності та повноти різноманітних пропозицій, висунутих різними постачальниками.
- Надати розробникам вказівки щодо необхідних функцій, необхідних для Інтернету речей, і того, як вони повинні взаємодіяти.
- Діяти як основа для стандартизації, яка сприяє сумісності та скорочує витрати.

Архітектура IoT складається з чотирьох рівнів (Рис. 1.3), кожен з яких відіграє життєво важливу роль у функціонуванні екосистеми IoT:

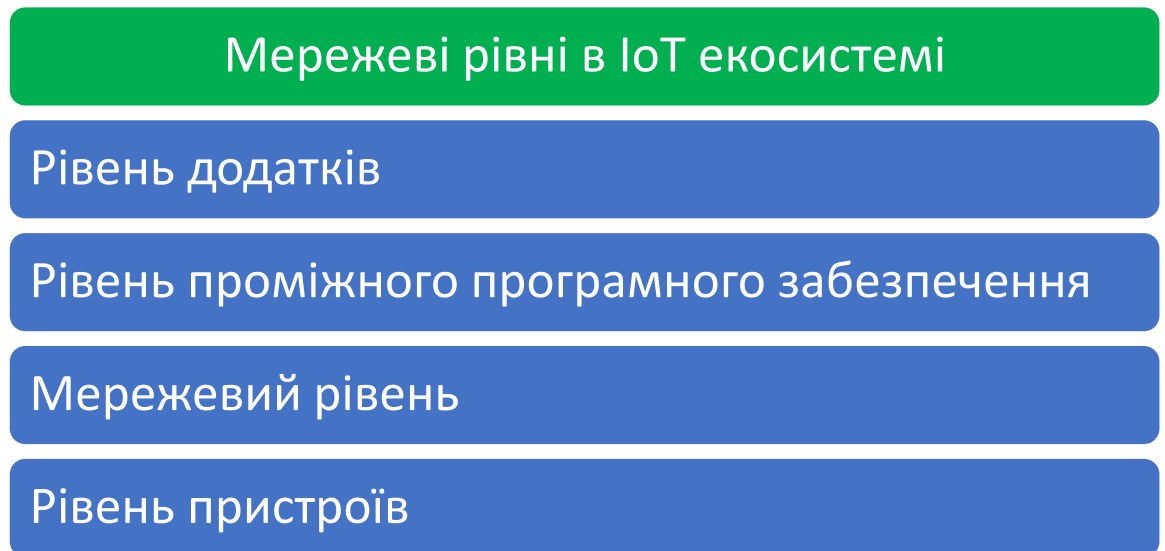


Рисунок 1.3 – Архітектура рівнів IoT

Рівень пристроїв – це найнижчий рівень архітектури IoT, який складається з усіх підключених пристроїв або «речей», які збирають і передають дані. Ці пристрої можуть включати датчики, приводи, камери та інші типи пристроїв. Рівень пристроїв відповідає за захоплення даних із фізичного світу та перетворення їх у цифровий формат, який може використовуватися рівнями вищого рівня.

Мережевий рівень: відповідає за передачу даних між пристроями та хмарою. Цей рівень містить різні протоколи, включаючи Wi-Fi, Bluetooth, Zigbee та інші, які забезпечують зв'язок між пристроями. Мережевий рівень забезпечує безпечну та ефективну передачу даних між пристроями та хмарою.

Рівень проміжного програмного забезпечення діє як міст між рівнем пристрою та прикладним рівнем. Він відповідає за керування та обробку даних, гарантуючи, що вони доставляються до відповідної програми своєчасно та ефективно. Рівень проміжного ПЗ може включати обробку, зберігання та аналіз даних, а також протоколи безпеки та інші служби.

Рівень додатків є найвищим рівнем архітектури IoT і відповідає за надання послуг і додатків, орієнтованих на користувачів. Цей рівень може включати інформаційні панелі, інструменти аналітики та інші програми, які дозволяють користувачам відстежувати та контролювати підключені пристрої. Рівень додатків – це місце, де реалізуються розуміння та цінність даних IoT, що робить його критично важливим рівнем як для компаній, так і для споживачів.

Щоб задовольнити вимоги до обчислювальних можливостей і одночасно керувати та контролювати операції шлюзу (наприклад, здійснення модифікацій параметрів у реальному часі, забезпечення моніторингу та діагностики пристроїв, впровадження заходів безпеки тощо), програмне забезпечення Java/OSGi Framework створено для шлюзів IoT.

1.5 Принципи роботи Інтернету речей

Ефективна організація роботи системи IoT вимагає обробки величезних обсягів різноманітних даних і перетворення їх у зручний і практичний формат для подальшого використання. Фільтрація та вибір важливих компонентів покращують швидкість виконання процесу. У загальному випадку алгоритм роботи IoT показана на рисунку 1.4 [15].

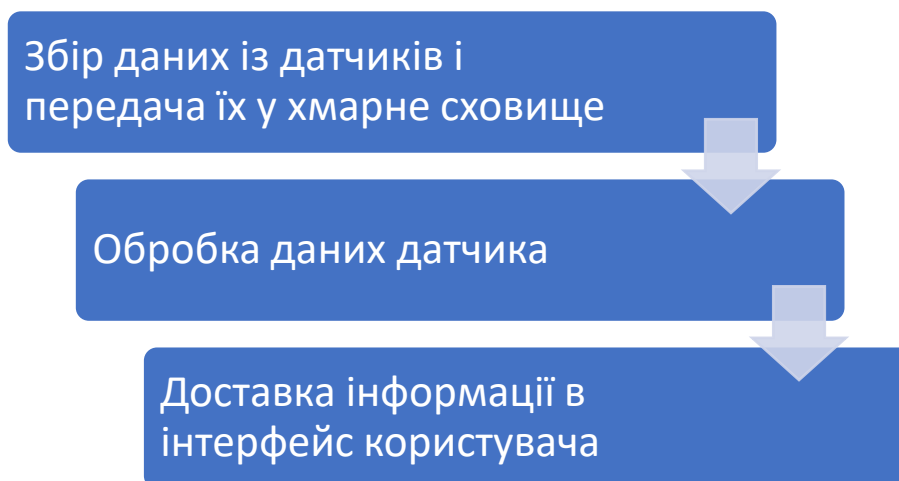


Рисунок 1.4 – Послідовність виконання операцій в IoT системах.

Система постійно контролює всі життєво важливі показники, які безпосередньо впливають на виконання основних етапів і процесів. Швидке виявлення проблемних моментів дозволяє створити оптимальні умови для ефективного їх усунення. Інтерфейс розробки дозволяє візуально відобразити процеси виконання дій, дозволяючи обслуговуючому працівнику візуально оцінювати стан технологічних елементів [15].

Платформа IoT виконує аналіз у режимі реального часу всіх необхідних даних, щоб виявити найбільш логічний і відповідний спосіб вирішення будь-яких проблем з діяльністю компанії або потенційного ризику виникнення таких проблем. Система налаштовується під конкретні типо-технологічні характеристики кожного окремого господарюючого суб'єкта, що значною мірою впливає на економічні показники діяльності за певний період часу.

Перевагами концепції IoT є:

Зручність. Технологія Інтернету речей полегшує життя завдяки автоматичним кавоваркам і автомобілям, які контролюють дорожні умови. Це також покращує якість життя, забезпечуючи свіжість їжі, замовленої з холодильника.

Ефективність і безпека. IoT зводить до мінімуму небезпечні ситуації та надзвичайні ситуації, що має вирішальне значення в промислових і комерційних умовах. Це також призводить до значного збільшення обсягів виробництва, оскільки багато операцій автоматизовані.

Підтримка прийняття рішень. Емпіричні дані, які надає IoT, допомагають визначити причинно-наслідкові зв'язки, передбачити тенденції та зменшити потребу в ручному аналізі даних у бізнесі.

Економія часу. Швидке реагування пристроїв IoT скорочує час, необхідний для виконання завдань, що може підвищити ефективність бізнесу та конкурентоспроможність. Це також економить час і забезпечує конфіденційність, оскільки немає необхідності вручну відстежувати інформацію.

Недоліками цієї технології є:

Необхідну **тривалу підготовку**, яка передбачає не лише навчання пристроїв розпізнаванню та позначенню різних об'єктів, але й встановлення зв'язку між пристроями різних брендів, оскільки наразі більшість гаджетів сумісні лише з продуктами одного виробника.

Відсутність сумісності та інтеграції даних є серйозною проблемою, оскільки сотні стандартів передачі й обробки даних можуть обмежувати сумісність. Існує також невизначеність щодо того, чи повинні пристрої працювати за тим самим стандартом, чи правила слід створювати на основі конкретного постачальника послуг Інтернету.

Іншим серйозним недоліком впровадження IoT є **проблема конфіденційності**, коли розумні домашні пристрої отримують велику кількість даних користувачів, таких як графіки, звички споживання, графіки прийому ліків і навіть місцезнаходження. Якщо ці дані потраплять у чужі руки, користувачі можуть зазнати значної шкоди та втрат, що зробить шифрування критичним аспектом розробки IoT.

1.6 Мікроконтролерні платформи для IoT

Для обробки та передачі сигналу від аналого-цифрового перетворювача (АЦП) можна використовувати будь-який обчислювальний пристрій, що має відповідні цифрові входи. Проте, коли мова йде про Інтернет речей (IoT), такий пристрій повинен бути встановлений всередині або поруч з обладнанням, до якого підключені датчики. Тому цілком раціональним вибором є використання недорогих та компактних пристроїв з мінімальним споживанням електроенергії. Сучасні мікроконтролери відповідають цим вимогам.

Мікроконтролер – це мікросхема, призначена для керування електронними пристроями. Вона поєднує функції процесора і периферійних пристроїв на одному кристалі і містить оперативну та (або) постійну пам'ять.

Головна відмінність від мікропроцесора полягає у наявності інтегрованих периферійних пристроїв (введення-виведення, таймери, аналого-цифрові перетворювачі, компаратори та інші).

Виробники мікроконтролерів пропонують різні лінійки пристроїв з різними технічними характеристиками та цінами. Серед них можна відзначити MCS 51 від Intel, ESP8266 і ESP32 від компанії Espressif, MSP430 від Texas Instruments, ARM розроблені компанією ARM Limited, AVR від Atmel, PIC від Microchip та багато інших.

Пряме використання мікросхеми вимагає створення печатної плати та значного обсягу роботи з монтажем всіх необхідних компонентів. Тому для прототипування широко використовуються універсальні платформи, зокрема Arduino.

Arduino – це електронна платформа з відкритим вихідним кодом, що включає просте у використанні апаратне та програмне забезпечення. Плати Arduino дозволяють зчитувати вхідні сигнали від датчиків (наприклад, світла, температури, маси і т.д.) та отримувати інформацію з зовнішніх джерел. За допомогою цих даних мікроконтролер може генерувати сигнали керування та надсилати повідомлення в інші системи. Для програмування використовується мова Arduino, яка базується на мові Wiring, і програмне забезпечення Arduino IDE, яке побудоване на основі Processing.

Класична версія цієї платформи базується на мікроконтролерах Atmel, але на сьогоднішній день існують моделі з мікроконтролерами Intel, AVR та іншими. На рисунку 1.5 показана одна з найбільш поширених моделей - Arduino Nano. Технічні характеристики цієї платформи представлені у таблиці 1.1.

Для платформи Arduino існують модулі розширення, які дозволяють підключити її до мережі Ethernet. Один із прикладів такого модуля - плата розширення Arduino Ethernet R3. Вона використовує контролер Wiznet W5100, який забезпечує можливість підключення Arduino до локальної мережі за допомогою Ethernet. Вона має вбудований мережевий стек, що дозволяє

здійснювати зв'язок з іншими пристроями в мережі, включаючи передачу та отримання даних через протоколи TCP/IP.



Рисунок 1.5 – Зовнішній вигляд мікроконтролерної платформи Arduino Nano

Таблиця 1.1 – Технічні характеристики Arduino Nano

Мікроконтролер	ATmega328P
Робоча напруга	5В
Вхідна рекомендована напруга	7-12В
Максимальний діапазон вхідної напруги	6-20В
Цифрові входи/виходи	14 шт. (з них 6 підключені до ШІМ)
Кількість виводі широтно-імпульсного модулятора (ШІМ)	6
Кількість аналогових входів	8
Постійний струм на портів вводу/виводу	20 мА
Флеш пам'ять	32 КБ (з них 2 КБ використовує bootloader)
SRAM	2 КБ
EEPROM	1 КБ
Частота тактового генератора	16 МГц

Плата розширення Arduino Ethernet R3 має роз'єм RJ45 для підключення Ethernet-кабелю, а також входи/виходи для з'єднання з Arduino. Вона дозволяє Arduino отримувати доступ до Інтернету, здійснювати комунікацію з веб-серверами, отримувати та надсилати дані по мережі. Цей модуль розширення є корисним для розробки проектів Інтернету речей (IoT), моніторингу, керування та інших додатків, які вимагають з'єднання з мережею Ethernet.

Particle Photon і Electron (Рис. 1.6) – це мікроконтролерні модулі, розроблені компанією Particle, спеціалізовані на розробці пристроїв Інтернету речей (IoT). Обидва модулі мають вбудовану підтримку зв'язку, але відрізняються за типом зв'язку, що вони використовують.

Particle Photon - це мікроконтролерний модуль, який працює на основі мікроконтролера STM32 ARM Cortex M3. Він має вбудовану підтримку Wi-Fi, що дозволяє забезпечити підключення до бездротової мережі. Модуль оснащений антеною для Wi-Fi, а також має вбудовану антену для Bluetooth.

Particle Photon і electron мають роз'єми для підключення зовнішніх пристроїв, таких як датчики, виходи, реле і інше. Вони можуть працювати з багатьма різними датчиками та пристроями завдяки своїм цифровим та аналоговим входам/виходам. Для програмування Particle Photon використовується мова C++.

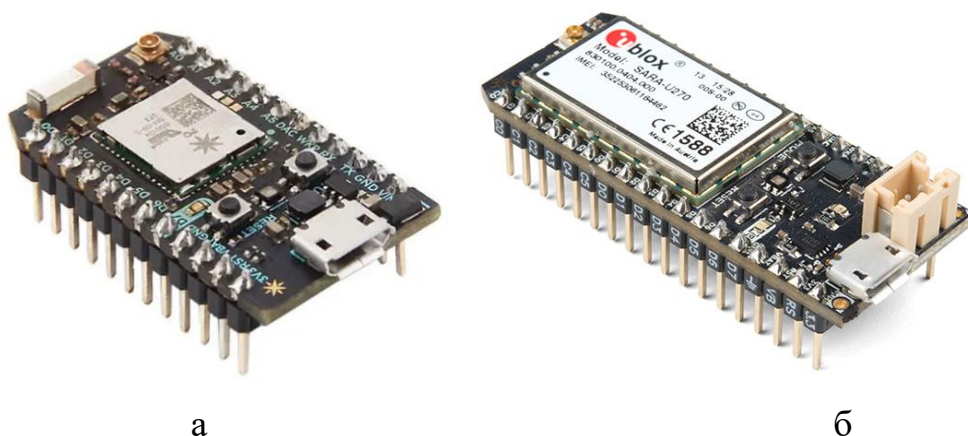


Рисунок 1.6 – Зовнішній вигляд мікроконтролерних платформ Particle Photon (а) і Electron (б)

Таблиця 1.2 – Технічні характеристики Particle Photon і Electron

	Photon	Electron
Мікроконтролер	STM32 ARM Cortex M3 з тактовою частотою 120 МГц	
Зв'язок	Wi-Fi 802.11 b/g/n	мережі 2G/3G
Пам'ять	128 КБ флеш-пам'яті для зберігання програмного коду та 1 МБ флеш-пам'яті для зберігання даних	
Вхід/вихід	18 (GPIO), з яких 8 можуть працювати як аналогові входи, 1 пін з підтримкою протоколу I2C, 1 пін з підтримкою протоколу SPI та 1 пін з підтримкою протоколу UART	
Інтерфейси	Інтерфейси: USB 2.0 для підключення до комп'ютера та живлення, JTAG для програмування та налагодження	
Живлення	3,3 В	3,7 В
Розміри	36 x 18 x 4,5 мм	53 x 23 x 10 мм
Підтримка	Є можливість використання хмарної інфраструктури Particle для керування та збору даних з пристрою	

Також значного поширення в сфері розробки пристроїв IoT набули модулі компанії Espressif Systems ESP8266 і ESP32, внаслідок своєї доступності, потужності та вбудованому бездротовому модулю зв'язку. Технічні характеристики ESP8266 наступні:

- Мікроконтролер: Tensilica L106 32-бітний RISC-процесор з тактовою частотою до 80 МГц.
- Зв'язок: Вбудована підтримка Wi-Fi 802.11 b/g/n для бездротового підключення до мережі.
- Пам'ять: 64 КБ інструкційної пам'яті (програмна пам'ять), 96 КБ динамічної пам'яті (RAM).

- Вхід/вихід: 17 цифрових вхідно-вихідних пінів (GPIO), з яких 1 пін підтримує протокол I2C, 1 пін підтримує протокол SPI та 1 пін підтримує протокол UART.

- Інтерфейси: UART, SPI, I2C, ADC, GPIO.

- Живлення: Запит живлення 3,3 В.

- Підтримка: Є можливість використання Arduino IDE або мови програмування Lua для розробки програмного забезпечення для ESP8266.

Платформа ESP32 має такі характеристики

- Мікроконтролер: Dual-core процесор Tensilica LX6 з тактовою частотою до 240 МГц.

- Зв'язок: Вбудована підтримка Wi-Fi 802.11 b/g/n та Bluetooth 4.2/5.0 для бездротового зв'язку.

- Пам'ять: ESP32 має вбудовану флеш-пам'ять розміром від 4 МБ до 16 МБ для зберігання програмного коду та даних.

- Вхід/вихід: ESP32 має 34 цифрових вхідно-вихідних піни (GPIO), з яких 18 пінів можуть працювати як аналогові входи (ADC).

- Інтерфейси: ESP32 підтримує різні інтерфейси, включаючи UART, SPI, I2C, I2S, CAN і Ethernet MAC.

- Живлення: Робоче напруга – 3,3 В. Підтримує живлення від батарей або зовнішнього джерела.

- Підтримка: Є можливість використання Arduino IDE або мови програмування Lua для розробки програмного забезпечення для ESP8266.

Порівнявши їх технічні характеристики можна сказати, що ESP32 потужніша і підтримує більшу кількість технологій. Незважаючи на це, обидва модулі активно застосовуються в розробці IoT пристроїв, адже мають унікальний у своєму ціновому сегменті набір можливостей, а саме, можливість підключення датчиків, потужний процесор, вбудовані безпроводні модулі зв'язку та великий об'єм пам'яті.

РОЗДІЛ 2. РОЗРОБКА ІОТ СИСТЕМ НА ОСНОВІ МІКРОКОНТРОЛЕРА

Концепція IoT розуміє під собою комплекс систем, рішень і засобів, необхідних для забезпечення функціонування розумних речей в єдиній екосистемі. Сьогодні багато компаній розробників та виробників обладнання мають ряд готових рішень IoT, вірніше сказати – платформ для побудови розумних локальних екосистем. На їх основі можна реалізувати самодостатню систему у рамках власного будинку, або господарства, сюди входить контролер, який дозволяє зв'язати розумні речі, і налаштувати їхню коректну взаємодію; набір розумних речей, датчиків та засобів керування; а також шлюз – який дозволяє локальній системі мати доступ в Інтернет, тим самим забезпечуючи можливість віддаленого доступу.

У контексті децентралізованих і гетерогенних середовищ, таких як Інтернет речей, підтримка семантичної взаємодії є досить складним завданням. Це вимагає масштабованості, гнучкості, відкритості, багаторівневої функціональності та здатності автономно реагувати на події навколишнього середовища. Крім того, система має бути прив'язаною до певних технологій, таких як мови програмування, операційні системи або методи транспортування даних.

Можливості створення IoT систем на сьогодні досить обширні, готові рішення мають ряд переваг, таких, як гарантована сумісність пристроїв, документування алгоритмів, можливостей та обмежень системи, проте такі системи можуть мати достатньо високу вартість.

Іншим, але не менш популярним, способом реалізації IoT систем є розробка їх на основі компонентів, які виготовляються серійно і розповсюджені на ринку. Хорошим підходом є використання мікроконтролерних обчислювальних платформ та сумісних із ними плат розширення (датчики, системи керування, відображення і таке інше).

У рамках моєї бакалаврської роботи метою була розробка системи

контролю та керування мікрокліматом і освітленням у приміщенні, а також реалізація функцій автоматизації. На основі мікроконтролерної платформи Arduino було спроектовано систему яка дозволяє реалізовувати функціонал розумного помешкання у рамках однієї кімнати. Система має наступні функціональні можливості:

- увімкнення світла при фіксації руху в кімнаті;
- аудіо сигналізація про наявність задимлення приміщення;
- увімкнення вентиляції при підвищенні температури;
- відкриття шухляди при піднесенні до неї руки.

Враховуючи поставлені задачі було розроблено принципову схему системи автоматизації (Рис. 2.1). До схеми входять такі елементи:

- мікроконтролерна платформа Arduino;
- піроелектричний датчик руху;
- електромеханічне реле;
- газовий датчик;
- датчик температури;
- електричний двигун (у нашому випадку він імітує роботу вентилятора охолодження повітря або ж кондиціонера);
- ультразвуковий далекомір;
- сервопривод;
- п'єзоелектричний динамік (зумер)

Моделі елементів, які використовуються у даній схемі можуть різнитися від умов та потреб, що забезпечується різноманіттям пристроїв з різними робочими характеристиками, та однаковими умовами протоколами роботи і схемами підключення. Це дозволяє позбутися необхідності внесення змін у програмну частину проєкту.

Описати роботу присторою IoT, створеного за схемою із рисунку 2.1 можна так: коли людина входить до робочого сектору датчика руху. Контролер отримує про це сигнал, і вмикає освітлення, останнє реалізовується

увімкненням лампочок, посередництвом електромеханічного реле; при реєстрації температури вище певного порогового рівня, вмикається вентилятор системи кондиціонування, його вимкнення здійснюється також спираючись на дані щодо температури; при реєстрації високої концентрації речовини, до якої чутливий датчик (наприклад CO₂, CO , або метан), буде увімкнено зумер, подачею контролером відповідного сигналу на фізичний порт; при піднесенні руки до ультразвукового далекоміра ближче певної, наперед заданої, відстані, буде задіяний сервопривод відкриття шухляди, при виході із зони дії далекоміра- шухляда буде закрита.

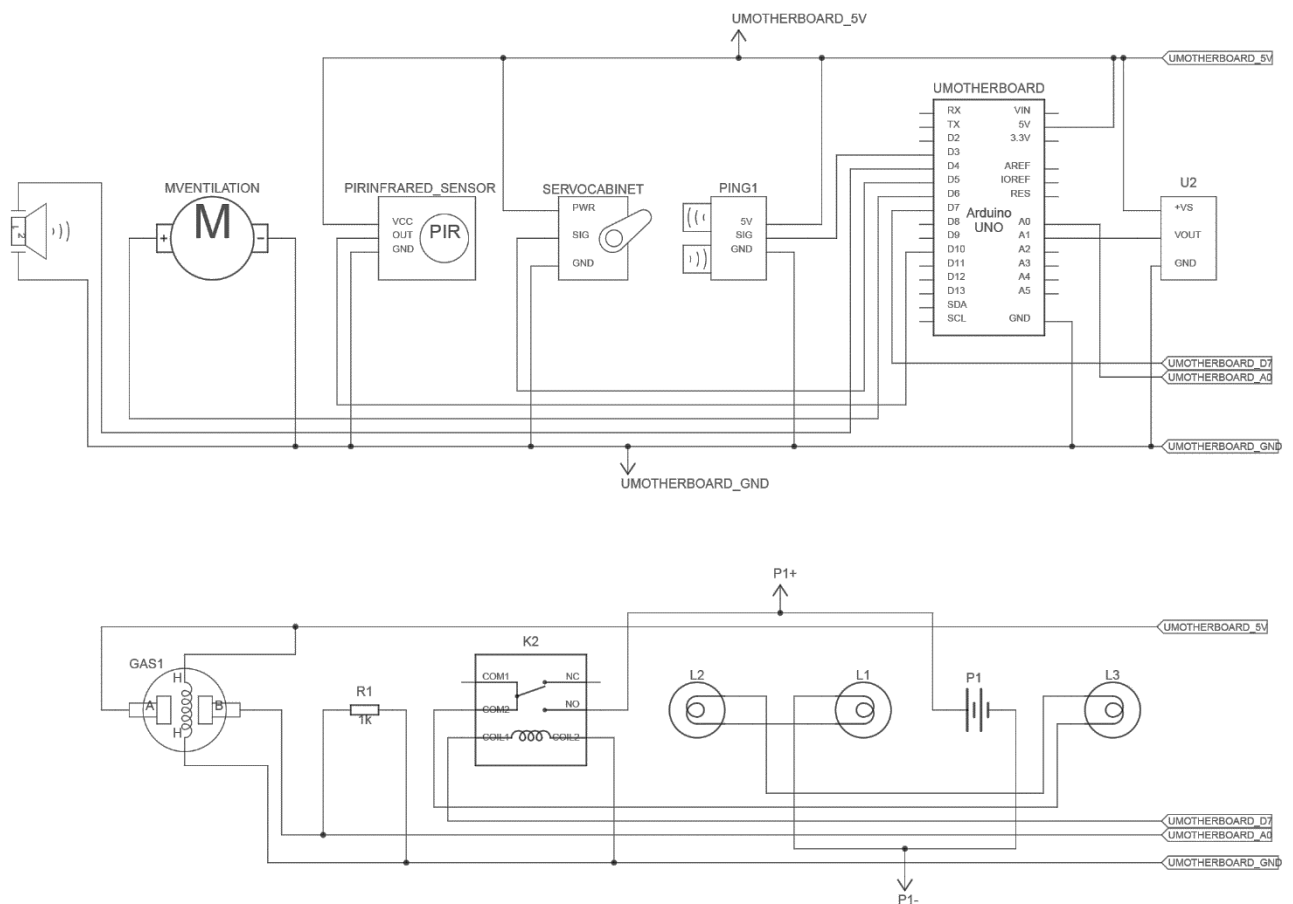


Рисунок 2.1 – Принципова схема системи керування мікрокліматом та освітленням на базі мікроконтролерної платформи Arduino

Моделювання і перевірка працездатності системи здійснювалась у онлайн середовищі Tinkercad (tinkercad.com), приклад схеми створеної у цьому

середовищі, показано на рисунку 2.2 а. Цей інструмент представляє собою систему автоматизованого проектування (САПР) електричних схем із можливістю моделювання їх роботи в режимі реального часу. Також цей САПР має вбудовану підтримку деяких мікроконтролерних платформ, зокрема й Arduino, та дозволяє програмувати їх у вбудованому середовищі розробки.

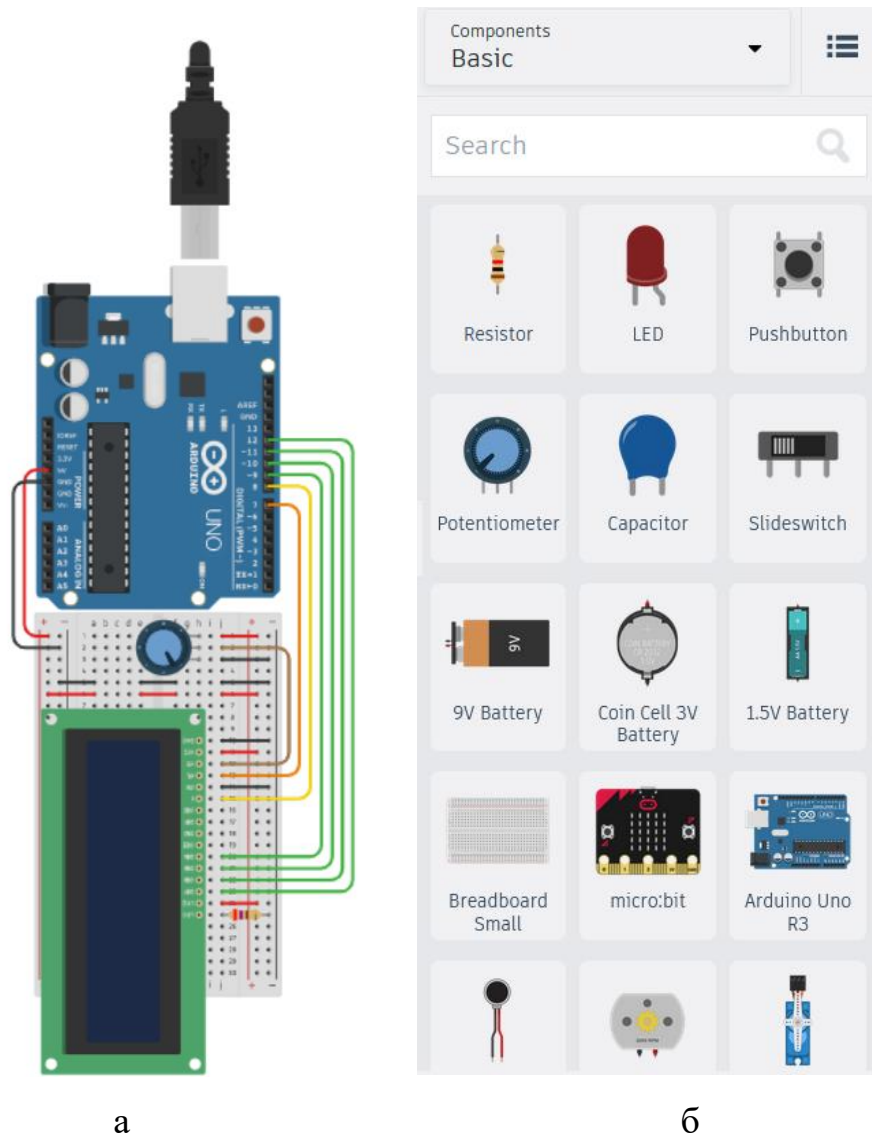


Рисунок 2.2 – Зовнішній вигляд робочого поля (а) та бібліотеки компонентів (б) Tinkercad

Tinkercad має також велику бібліотеку радіоелектронних компонентів (Рис. 2.2 б) та датчиків, що і були використані в розроблюваному проєкті.

РОЗДІЛ 3. КОПОНУВАННЯ ТА ПРОГРАМУВАННЯ ІОТ СИСТЕМ НА ОСНОВІ ARDUINO

3.1. Створення функціональної схеми ІоТ системи

Першим етапом створення автоматизованої системи керування у рамках концепції розумного будинку в середовищі Tinkercad є розміщення на робочому полі усіх необхідних компонентів. Відповідно до розробленої принципової схеми (Рис. 2.1) нам необхідно обрати з бібліотеки наступні: мікроконтролерна платформа Arduino, піроелектричний датчик руху, електромеханічне реле, газовий датчик, датчик температури, електричний двигун; ультразвуковий далекомір, сервопривод, п'єзоелектричний динамік (зумер), лампочки та джерело живлення. Розмістивши елементи на робочому полі об'єднуємо їх відповідно до схеми, у результаті чого отримуємо систему, показану на рисунку 3.1.

Варто зазначити, що створена нами система може вважатися лише частиною ІоТ, оскільки працює автономно, не передаючи і не отримуючі дані із зовні. Оскільки функціональність пристрою ІоТ складається з чотирьох ключових етапів: збір даних, їх обробка та контроль, зберігання а також передача даних і зв'язок. Зважаючи на те, що останній етап не реалізовується, то більш коректно називати створену систему ІоТ модулем. Такий функціонал може бути також реалізований, додаванням до проєкту мережевого модуля зв'язку та модифікацією програмного забезпечення. Також це дало б можливість розробити систему віддаленого моніторингу і керування, наприклад у вигляді застосунку для смартфона чи Web-інтерфейсу.

Також розширити можливості автономного пристрою може дозволити його інтеграція з хмарними сервісами, проте в такому випадку необхідно буде виконати ряд умов онлайн сервісу, та забезпечити повноцінну підтримку її функціональних можливостей.

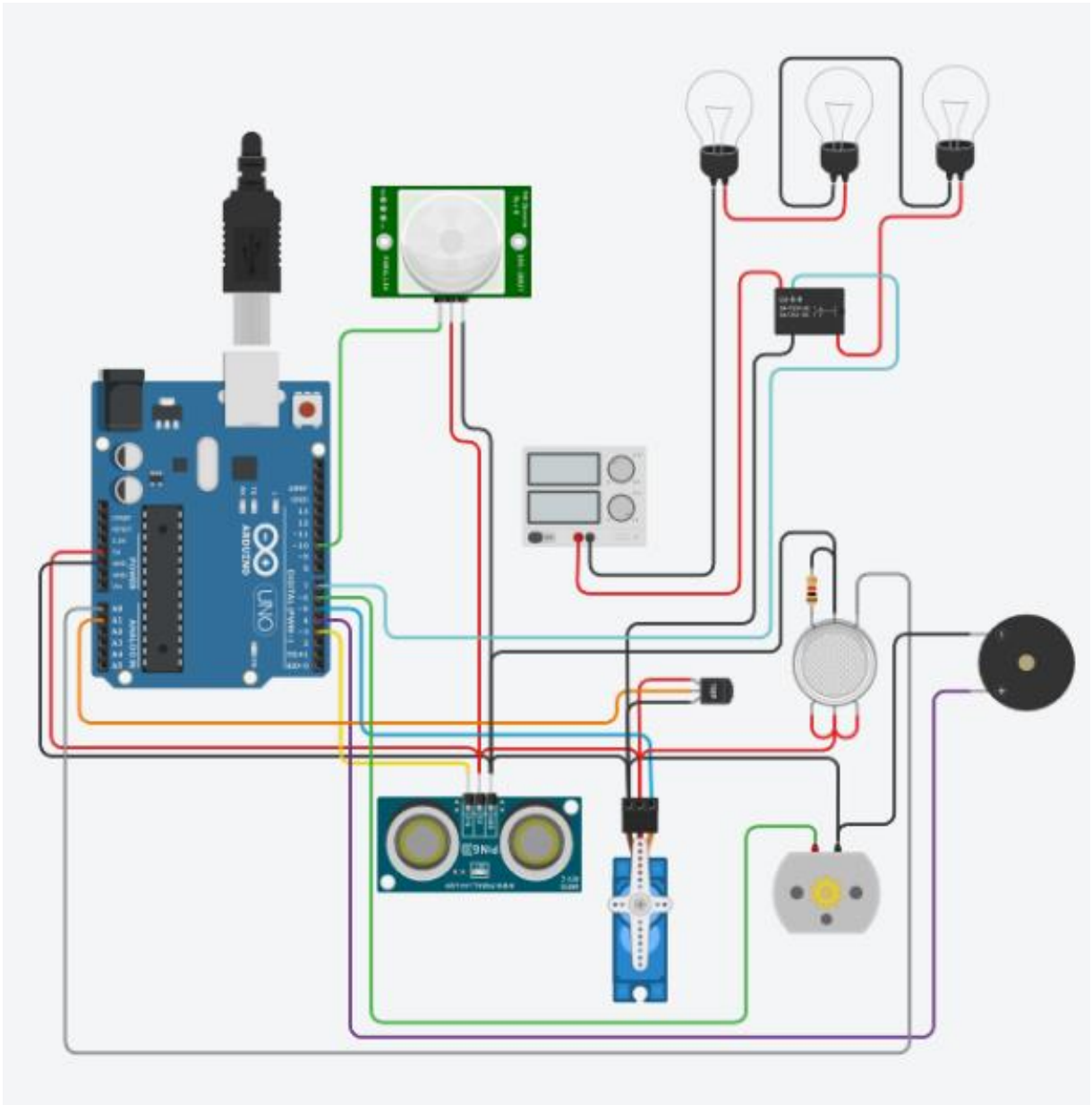


Рисунок 3.1 – Розроблена функціональна схема IoT системи в середовищі Tinkercad

3.2. Створення програмного забезпечення IoT системи

Як контролер, у нашій IoT системі використовується платформа Arduino, побудована на основі мікроконтролера AVR ATmega328, написання програмного коду для якого може здійснюватися багатьма мовами програмування. Середовище розробки Tinkercad дозволяє програмувати контролери Arduino з використанням мов Scratch та C++, останньою ми і

будемо користуватись, оскільки вона за розумного рівня деталізації технічних моментів розробки програмного забезпечення має високий рівень абстракції. Tinkercad має вбудований редактор коду, та не потребує жодного стороннього додатку для його написання.

Розглянемо програмну реалізацію закладених у нашу систему функцій. У мові C++ програма традиційно починається із оголошення директив препроцесора та опису глобальних змінних, ця частина коду має наступний вигляд:

```
#include <Servo.h> //Підключення бібліотеки для
роботи із сервоприводами
int Cabinet = 0;
int PIRS = 0;
int Gass = 0;
int Temps = 0;
```

Він реалізовує підключення бібліотеки для роботи із сервоприводами, а також оголошує глобальні цілочисельні змінні, які необхідні для зберігання поточного стану сенсорів руху, температури, наближення та наявності газу.

Далі нам потрібно описати функцію Setup(), яка описує налаштування портів введення/виведення, ініціалізацію та налаштування периферійного обладнання і мережевих протоколів. Тут ми ініціалізуємо передачу даних за послідовним інтерфейсом передачі даних, що необхідно для відладки програмного забезпечення, налаштуємо параметри роботи сервоприводу і задаємо режим роботи цифрових пінів як входів, чи виходів.

```
void setup()
{
    Serial.begin(9600);
    servo_5.attach(5, 500, 2500);
    pinMode(10, INPUT);
    pinMode(7, OUTPUT);
    pinMode(A1, INPUT);
```

```

pinMode(6, OUTPUT);
pinMode(A0, INPUT);
pinMode(4, OUTPUT);
}

```

Окремо, у вигляді функції, описуємо частину коду для обробки вхідних даних від ультразвукового далекоміра:

```

long  readUltrasonicDistance(int  triggerPin,  int
echoPin)
{
    pinMode(triggerPin, OUTPUT); // Clear the trigger
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    // Sets the trigger pin to HIGH state for 10
microseconds
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);
    pinMode(echoPin, INPUT);
    // Reads the echo pin, and returns the sound wave
travel time in microseconds
    return pulseIn(echoPin, HIGH);
}

```

Далі описуємо код основної програми, який виконуватиметься циклічно. На кожній ітерації головного циклу ми виконуємо по чергове опитування датчиків, обробляємо одержані дані та виконуємо необхідні дії, при настанні певних подій.

Першим опишемо програмну реалізацію увімкнення освітлення, програмний код якої має наступний вигляд:

```

PIRS = digitalRead(10);
Serial.println(PIRS);

```

```

    if (PIRS == HIGH) {
        digitalWrite(7, HIGH);
    } else {
        digitalWrite(7, LOW);
    }

```

Спочатку ми опитуємо цифровий порт 10, до якого підключено датчик руху, далі, якщо в робочому секторі зафіксовано рух (повернуто значення HIGH), ми виставляємо високий логічний рівень на порту 7, інакше – низький логічний рівень. До цифрового порту 7 підключений керуючий вхід електромеханічного реле, при зпрацюванні якого замкнеться коло освітлення і увімкнуться лампочки.

Наступною є реалізація спрацювання сигналізації при фіксації підвищеної концентрації газу:

```

Gass = analogRead(A0);
Serial.println(Gass);
if (Gass >= 220) {
    digitalWrite(4, HIGH);
} else {
    digitalWrite(4, LOW);
}

```

Логіка роботи цієї частини коду схожа із попередньою, завийнятком того, що гадовий датчик має аналоговий сигнал на виході, і програмно його порівнюють із певним наперед заданим значенням, виходячи із градуювальної кривої для конкретного типу датчика. При зправдженні умови спрацювання сенсора, ми подаємо логічну одиницю на порт 4, до якого під'єднаний зумер, за необхідності це може бути й інший пристрій, наприклад світлова сигналізація.

Третім датчиком є сенсор температури, частина програмного коду, що стосується роботи із ним має такий вигляд:

```

Temps = (-40 + 0.488155 * (analogRead(A1) - 20));

```



```

Serial.println(Temps);
if (Temps >= 30) {
    digitalWrite(6, HIGH);
} else {
    digitalWrite(6, LOW);
}

```

На першому етапі ми зчитуємо значення із датчика, та підстановкою в градууювальну функцію отримуємо значення температури в градусах цельсія, далі порівнюємо її із пороговим, наперед заданим значенням температури і у разі його перевищення подаємо високий логічний рівень на цифровий порт 6, до якого підключений двигун постійного струму, що імітує вентилятор системи охолодження.

Останнім датчиком у нашій системі є ультразвуковий далекомір, який фіксує наближення руки до шухляди і відкриває її. Він має наступний вигляд:

```

Cabinet = 0.01723 * readUltrasonicDistance(3, 3);
Serial.println(Cabinet);
if (Cabinet < 15) {
    servo_5.write(90);
    delay(5000); // Wait for 5000 millisecond(s)
} else {
    servo_5.write(0);
}

```

Логічна частина цього коду працює таким чином, що при фіксації перешкоди, ближче певного порогового значення (у нашому випадку 15 см) викликаємо описану вище функцію повороту сервопривода, з параметром відкриття шухлядки, коли перешкода перестає фіксуватись, то після очікування 5 секунд викликаємо функцію знову, але із параметром на закриття шухляди.

ВИСНОВКИ

1. Виконано аналіз літературних джерел за темою IoT. Розглянуто принципи роботи таких мереж їх архітектуру та механізми роботи. Показано, що мережа IoT –це група розумних речей (пристроїв), які об'єднані між собою у самоорганізуючуся структуру, і здатні взаємодіяти між собою.

2. Показано, що локальна IoT мережа може бути складовою частиною глобальної структури, як то хмарний сервіс, та обмінюватись із нею даними. Насьогодні існує велика кількість надавачів хмарних послуг Інтернету речей для підключення до неї власних пристроїв та мереж, наприклад, Arduino IoT Cloud, Google Cloud IoT, AWS, PTC ThingWorx, Kaa IoT, Платформа Azure IoT. Кожен із цих сервісів має унікальні особливості, які забезпечують переваги перед конкурентами.

3. Розглянуто мікроконтролерні платформи, на основі яких можуть бути побудовані пристрої та мережі IoT. Показано, що для цього добре підходять платформи MCS 51 від Intel, ESP8266 і ESP32 від компанії Espressif, MSP430 від Texas Instruments, ARM від ARM Limited, AVR від Atmel та PIC від Microchip.

4. Розроблено схему мережі IoT на основі Arduino та змодельовано її в середовищі Tinkercad для автоматизації процесів у приміщенні. Система забезпечує функції автоматичного увімкнення світла, автоматичного кондиціонування сигналізацію про перевищення концентрації газу та автоматичне відкриття шухляди.

5. Розроблено програмне забезпечення мовою C++, яке описує алгоритм роботи системи автоматичного керування. Програма побудована на основі безкінечного циклу та має структуру із поділом на окремі логічні блоки, кожен з яких відповідає за певний функціонал системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Internet of Things [Електронний ресурс]. – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/internet-veschej-internet-of-things-iot> Дата доступу: 11.05.2023
2. Klaus Schwab. The Fourth Industrial Revolution. Currency. – 2017. – 189 с.
3. Peter Waher. Learning Internet of Things. Packt Publishing. – 2015. – 242 с.
- 4 Cuno Pfister. Getting Started with the Internet of Things: Connecting // Sensors and Microcontrollers to the Cloud (Make: Projects). Make Community, LLC; 1st edition. – 2011. – 255 с.
5. Arduino IoT Cloud [Електронний ресурс]. – Режим доступу: <https://docs.arduino.cc/arduino-cloud/> Дата доступу: 11.05.2023
6. Google Cloud IoT [Електронний ресурс]. – Режим доступу: <https://cloud.google.com/iot-core> Дата доступу: 11.05.2023
7. AWS IoT [Електронний ресурс]. – Режим доступу: <https://aws.amazon.com/ru/iot/> Дата доступу: 11.05.2023
8. PTC ThingWorx [Електронний ресурс]. – Режим доступу: <https://www.ptc.com/en/products/thingworx> Дата доступу: 11.05.2023
9. Kaa IoT Platform [Електронний ресурс]. – Режим доступу: <https://www.kaaiot.com/>
10. Microsoft Azure IoT [Електронний ресурс]. – Режим доступу: <https://azure.microsoft.com/ru-ru/products/iot-hub/> Дата доступу: 11.05.2023
11. ITU-T, “Overview of the Internet of Things,” Recommendation Y.2060, June 2012.
12. Manohar H. L. T. Data Consumption Pattern of MQTT Protocol for IoT Applications. In: Venkataramani G., Sankaranarayanan K., Mukherjee S., Arputharaj K., Sankara Narayanan S. (eds) Smart Secure Systems – IoT and Analytics Perspective. ICIT 2017. Communications in Computer and Information Science, vol 808. Springer, – Singapur, 2018 – P. 97 – 99;
13. Загальні поняття Інтернету речей.[Електронний ресурс] – Режим

- доступу: <http://academicfox.com/lektsiya-1-zahalni-ponyattya-interneturechej/>
Дата доступу: 10.05.2023
14. . Internet of Things Platforms. Postscapes. // [Електронний ресурс.] – Режим доступу: <http://postscapes.com/internet-of-thingsplatforms?order=rhits>
Дата доступу: 14.05.2023
15. Connected Pigs: Google Partner Agosto Deploys IoT Solution // [Електронний ресурс.] – Режим доступу: - <https://www.crn.com/news/internet-of-things/300095681/connected-pigs-googlepartner-agosto-deploys-iot-solution-to-help-farmers-remotely-monitor-livestock.htm> Дата доступу: 11.05.2023
16. A comparative study of LPWAN technologies for large-scale IoT deployment // [Електронний ресурс.] – Режим доступу: <https://www.sciencedirect.com/science/article/pii/S2405959517302953> Дата доступу: 16.05.2023
17. Analytics Perspective. ICIT 2017. Communications in Computer and Information Science, V/ 808. Springer, – Singapur, 2018 – P. 97 – 99.
18. P. Lopez, D. Fernandez, A. J. Jara, and A. F. Skarmeta, “Survey of Internet of Things technologies for clinical environments,” in Proc. 27th Int. Conf. WAINA, 2013, pp. 1349–1354.
19. Internet of Things Platforms. Postscapes. // [Електронний ресурс.] – Режим доступу: <http://postscapes.com/internet-of-thingsplatforms?order=rhits> Дата доступу: 16.05.2023
20. Manohar H. L. T. Data Consumption Pattern of MQTT Protocol for IoT Applications. In: Venkataramani G., Sankaranarayanan K., Mukherjee S., Arputharaj K., Sankara Narayanan S. (eds) Smart Secure Systems – IoT and Analytics Perspective. ICIT 2017. Communications in Computer and Information Science, vol 808. Springer, – Singapur, 2018 – P. 97 – 99;
21. Пелих Р.К., Тищенко К.В. Організація Mesh мереж в розрізі технології IoT / Матеріали Міжнародної науково-технічної конференції студентів та молодих вчених «Фізика, електроніка, електротехніка ФЕЕ-2023». – Суми: СумДУ, 2023. – С.56.