

6. Орієнтовний перелік публікацій

1. Яременко К.М.. Порівняння Netburner, Arduino та Raspbery Pi для використання в IoT / К.М.Яременко. // Міжнародний науковий журнал "Інтернаука". – 2018. – №4..
2. Яременко К.М.. Використання блокчейн технології разом з IoT: перспективи і проблеми на шляху розвитку/ К.М.Яременко. // Міжнародний науковий журнал "Інтернаука". – 2018. – №4..

7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Реалізація стартап-проекту	Кирюша Б.А., доцент		

8. Дата видачі завдання 01.02.2018

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	01.02.2018	
2	Збір інформації та аналіз літератури	15.02.2018	
3	Проведення огляду існуючих засобів для розгортання Інтернету речей	28.02.2018	
4	Аналіз засобів для автоматизації планування розумного дому	10.03.2018	
5	Створення онлайн калькулятора для планування інфраструктури розумного дому	1.04.2018	
6	Оформлення дипломної роботи	30.04.2018	
7	Отримання допуску до захисту та подача роботи в ДЕК	10.05.2018	

Студент

(підпис)

Яременко К.М.

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

Кирюша Б.А.

(ініціали, прізвище)

РЕФЕРАТ

НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ

виконану на тему: Автоматизація планування розумного будинку

студентом: Яременко Костянтином Миколайовичем

Робота виконана на 81 сторінках, містить 14 ілюстрацій, 23 таблиць.

При

підготовці використовувалась література з 14 джерел.

Актуальність теми

В роботі розглядається одна з найактуальніших тенденцій розвитку інформаційних технологій Internet of Things (IoT). Проводиться огляд і аналіз рішень, розроблених відповідно до концепції Інтернету речей. Дана технологія набуває все більшої популярності, оскільки дозволяє об'єднати в єдину мережу прилади для моніторингу та керування елементами оточуючого простору, що в свою чергу надає безліч можливостей для автоматизації як виробничих процесів так і буденних справ кожної людини. Тому дослідження даної технології, а також створення програмного засобу для планування розумного будинку, є актуальним напрямком досліджень саме на сьогоднішній день.

Мета та задачі дослідження

Метою даної роботи є дослідження існуючих засобів для розгортання IoT, а також аналіз комерційних і некомерційних хмарних платформ Інтернету речей. Результатом проведених досліджень є практична частина роботи, що являє собою програмний засіб для автоматизації проектування розумного дому.

Рішення поставлених завдань та досягнуті результати

У даній роботі було проведено аналіз хмарних платформ. Були знайдені відмінності між розглянутими сервісами та виділені сильні та слабкі сторони кожного з них. Було створено онлайн калькулятор для проектування розумного будинку, що дозволяє користувачеві обрати необхідний набір пристроїв та

розгорнути їх в декілька нескладних кроків на базі одноплатового комп'ютера *Raspberry Pi* та відкритої хмарної платформи *ThingSpeak*. Даний програмний засіб можна також визначити, як стартовий комплект для початківців в сфері IoT, адже дозволяє користувачеві власноруч налаштувати зв'язки між пристроями мережі, крім того не обов'язково використовувати дорогі прилади таких відомих компаній як *Nest*, *SmartThings*, *Ecobee*, а й звичані сенсори та прилади, що можна підключити через GPIO інтерфейс *Raspberry Pi*. Також в ході виконання дипломної роботи була побудований робочий прототип розумного будинку з датчиками температури, вологості та освітленості.

Об'єкт досліджень

Інтернет речей.

Предмет досліджень

Способи та засоби автоматизації проектування розумного дому на базі хмарних платформ Інтернет речей.

Методи досліджень

Для вирішення задач в даній роботі використовуються методи аналізу, системного аналізу, порівняння, проектування логічних структур даних, логічного узагальнення результатів.

Наукова новизна

Наукова новизна роботи полягає у створенні нової системи для вирішення задачі автоматизації планування розумного дому на базі хмарних платформ.

Практичне значення одержаних результатів

Отримані результати можуть використовуватись у майбутніх дослідженнях за даним напрямком, враховуючи переваги та недоліки результатів проведеної роботи. Також розроблена програма може бути використана як основа для створення лабораторного практикуму на кафедрі

Системного проектування.

Публікації

Яременко К.М.. Порівняння Netburner, Arduino та Raspberry Pi для використання в IoT / К.М.Яременко. // Міжнародний науковий журнал "Інтернаука". – 2018. – №4..

Яременко К.М.. Використання блокчейн технології разом з IoT: перспективи і проблеми на шляху розвитку/ К.М.Яременко. // Міжнародний науковий журнал "Інтернаука". – 2018. – №4..

Ключові слова

Інтернет речей, MQTT, Raspberry Pi, Node.JS, Python, датчики, хмарна платформа, розумний будинок.

РЕФЕРАТ

НА МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ

На тему: Автоматизация планирования умного дома

студентом: Яременко Константином Николаевичем

Работа выполнен на 81 страницах, содержит 14 иллюстраций, 23 таблиц.

При подготовке использовалась литература с 14 источников.

Актуальность темы

В работе рассматривается одна из самых актуальных тенденций развития информационных технологий Internet of Things (IoT). Проводится обзор и анализ решений, разработка в соответствии с концепцией интернета вещей. Данная технология приобретает все большую популярность, поскольку позволяет объединить в единую сеть приборы для мониторинга и управления элементами окружающего пространства, что в свою очередь предоставляет множество возможностей для автоматизации как производственных процессов так и будничных дел каждого человека. Поэтому исследования данной технологии, а также создания программного средства для планирования умного дома, является актуальным направлением исследований именно на сегодняшний день.

Цель и задачи исследования

Целью данной работы является исследование существующих средств для развертывания IoT, а также анализ коммерческих и некоммерческими облачных платформ интернета вещей. Результатами проведенных исследований является практическая часть работы, что представляет собой программное средство для автоматизации проектирования умного дома.

Решение поставленных задачи и достигнутые результаты

В данной работе был проведен анализ облачных платформ. Были найдены

Различия между Рассмотрены сервисами и выделенных Сильные и слабые стороны кожи из них. Был создан онлайн калькулятор для проектирования умного дома, что позволяет пользователю выбрать необходимый набор устройств и развернуть их в несколько несложных шагов на базе одноплатового компьютера Raspberry Pi и открытой облачной платформы ThingSpeak. Данное программное средство можно определить, как стартовый комплект для начинающих в сфере IoT, поскольку оно позволяет пользователю собственноручно настроить связи между устройствами, кроме того не обязательно использовать дорогие приборы таких известных компаний как Nest, SmartThings, Ecobee и т. д., но и обычные сенсоры и приборы, которые можно подключить через GPIO интерфейс Raspberry Pi. Также в ходе выполнения дипломной работы был построен рабочий прототип умного дома с датчиками температуры, влажности и освещенности.

Объект исследований

Интернет вещей.

Предмет исследований

Способы и средства автоматизации проектирования умного дома на базе облачной платформ Интернета вещей.

Методы исследований

Для решению задач в данной работе Используют методы АНАЛИЗА, системного анализа, сравнения, проектирование логической структуры данных, логического обобщения результатов.

Научная новизна

Научная новизна работы заключается в создании новой системы по разрешению задачи автоматизации планирования умного дома на базе облачной платформы.

Практическое значение полученных результатов

Полученные результаты могут использоваться в будущих исследованиях по данному направлению, включая преимущества и недостатки результатов проведенной работы. Также разработана программа может быть использована в качестве основы для создания лабораторного практикума на кафедре Системного проектирования.

Публикации

Яременко К.М .. Сравнение Netburner, Arduino и Raspbery Pi для использование в IoT / К.М.Яременко. // Международный научный журнал "Интернаука". - 2018. - №4 ..

Яременко К.М .. Использование блокчейн технологии вместе с IoT: перспективы и проблемы на пути развития / К.М.Яременко. // Международный научный журнал "Интернаука". - 2018. - №4 ..

Ключевые слова

Интернет речей, MQTT, Raspbery Pi, Node.JS, Python, датчики, облачная платформа, умный дом.

ABSTRACT

ON MASTER'S THESIS

on topic: Automated home IoT planning

student: Kostiantyn M. Yaremenko

Work carried out on 81 pages containing 14 figures, 23 tables. The paper was written with references to 14 different sources.

Topicality

One of the most topical trends in the development of information technologies Internet of Things (IoT) is considered in this work. The review and analysis of solutions, development in accordance with the concept of the Internet of things is conducted. This technology becomes more and more popular, as it allows to unite in a single network devices for monitoring and controlling elements of the surrounding space, which in turn provides many opportunities for automation of both production processes and everyday affairs of each person. Therefore, the research of this technology, as well as the creation of a software tool for planning a smart home, is the current direction of research precisely for today.

Purpose

The purpose of this paper is to investigate existing tools for deploying IoT, as well as analysis of commercial and non-commercial cloud Internet platforms of things. The results of these studies are the practical part of the work, which is a software tool for automating the design of a smart home.

Solution

Was carried out analysis of cloud platforms. The differences were found between the considered services and the highlighted strengths and weaknesses of the skin of them. An online calculator for designing a smart home was created, which allows the user to select the required set of devices and deploy them in several simple steps based

on the single-board computer Raspberry Pi and the open cloud platform ThingSpeak. This software can be defined as a starter kit for beginners in the IoT area, since it allows the user manually set up communications between devices, besides, it is not necessary to use expensive devices from such well-known companies as Nest, SmartThings, Ecobee, etc., but also sensors and devices, which can be connected via the GPIO interface Raspberry Pi. Also during the graduation work a working prototype of a smart house with temperature, humidity and illumination sensors was built.

The object of research

Internet of things.

The subject of research

The subject of research is the methods and tools for automating the design of a smart home based on the cloud platform of the Internet of things.

Research methods

To solve the problem in this research such methods were used: analysis and synthesis, system analysis, comparison, logical generalization of the results, design logical data structures.

Scientific novelty

The scientific novelty of the work is to create a new system for solving the problem of automating the planning of a smart home based on a cloud platform.

The practical value of research

The obtained results can be used in future studies in this field, including the advantages and disadvantages of the results of an operation. The program can also be used as a basis for creating a laboratory workshop at the Department of System Design.

Publications

Yaremenko K. Comparison of Netburner, Arduino and Raspberry Pi for use in IoT / Kostya Yaremenko. // International scientific journal "Internauka". - 2018. - №4 ..

Yaremenko K. Use blokcheyn technology together with IoT: perspectives and problems on the way of development / Kostya Yaremenko. // International scientific journal "Internauka". - 2018. - №4 ..

Keywords

Internet speeches, MQTT, Raspberry Pi, Node.JS, Python, sensors, cloud platform, smart house.

ЗМІСТ

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ ..	16
ВСТУП.....	17
1 КОНЦЕПЦІ РОЗУМНОГО БУДИНКУ. ІНТЕРНЕТ РЕЧЕЙ ТА ХМАРНІ ОБЧИСЛЕННЯ	19
1.1 Історія розвитку	19
1.2 Інтернет речей та хмарні обчислення.....	20
1.3 Протокол взаємодії хмарного сервера з пристроями розумного будинку	23
1.3.1 SOAP	23
1.3.2 MQTT	23
1.4 Висновки до розділу 1	25
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ДЛЯ СТВОРЕННЯ ІНФРАСТРУКТУРИ РОЗУМНОГО БУДИНКУ	26
2.1 Система домашньої автоматизації Nest.....	26
2.2 Система домашньої автоматизації SmartThings	27
2.3 Система автоматизації освітлення Philips Hue	29
2.4 Система домашньої автоматизації Ecobee	29
2.5 Система автоматизації домашньої електроніки Belkin WeMo	31
2.6 Система домашньої автоматизації Lutron	32
2.7 Висновки до розділу 2	33
3 АНАЛІЗ ІСНУЮЧИХ ЗАСОБІВ ДЛЯ РОЗГОРТАННЯ ІОТ	35
3.1 AWS IoT Core	35
3.1.1 Підключення пристроїв та управління ними.....	36
3.1.2 Безпека підключення і даних	36
3.1.3 Обробка даних пристроїв	36
3.2 Google Cloud IoT	37
3.2.1 Кінцевий рівень безпеки.....	38
3.2.2 Протоколи передачі даних.....	39

3.2.3	Основні моменти	40
3.3	Microsoft Azure IoT Suite.....	40
3.3.1	Безпека Azure IoT Suite.....	42
3.3.2	Захищена підготовка пристроїв і перевірка справжності	43
3.3.3	Захищене з'єднання	43
3.3.4	Безпечна обробка і зберігання в хмарі	44
3.4	IBM Watson IoT.....	45
3.4.1	Шлюзові пристрої.....	45
3.4.2	Управління пристроєм	46
3.4.3	Розширення та інтеграція служб.....	46
3.4.4	Зберігання даних пристрою.....	47
3.4.5	Управління даними	47
3.4.6	Управління ризиками	48
3.4.7	Основні моменти	48
3.5	ThingSpeak Platform	48
3.5.1	Заявлений функціонал сервісу.....	49
3.5.2	Основні моменти	50
3.6	Висновки до розділу 3	50
4	Розробка онлайн калькулятора для планування розумного будинку	52
4.1	Raspberry Pi.....	52
4.2	Node.JS	54
4.3	Python	54
4.4	Вимоги до системи	56
4.5	Веб-застосунок.....	56
4.6.	Висновки до розділу.....	58
5	РЕАЛІЗАЦІЯ СТАРТАП-ПРОЕКТУ «ОНЛАЙН КАЛЬКУЛЯТОР ДЛЯ ПЛАНУВАННЯ ІНФРАСТРУКТУРИ РОЗУМНОГО БУДИНКУ».....	60
5.1	Опис ідеї проекту.....	60
5.2	Технологічний аудит ідеї проекту	61

5.3	Аналіз ринкових можливостей.....	62
5.4	Розробка ринкової стратегії проекту	68
5.5	Розробка маркетингової програми.....	72
5.6	Висновки до розділу 5	75
ВИСНОВОК.....		77
ПЕРЕЛІК ПОСИЛАНЬ		80

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

IoT – Internet of Things

AWS – Amazon Web Services

Node.js – програмна платформа для мережеских застосунків

Express. – фреймворк для розробки веб-додатків для Node.js

Raspberry Pi – одноплатний комп'ютер

PostgreSQL – об'єктно-реляційна система керування базами даних

MQTT – Message Queue Telemetry Transport

SOAP – Simple Object Access Protocol

ВСТУП

Інтернет речей - це сукупність фізичних об'єктів, які використовують мережеву підтримку для обміну даними. Ці об'єкти можуть бути датчиками, програмним забезпеченням, платами тощо. Це цікава екосистема, де програмне забезпечення може бути безпосередньо пов'язане з реальним обладнанням або пристроями. Найбільш відомі платформи, які можна використовувати для IoT проекту, - Arduino і Raspberry Pi.

Інтеграція цих пристроїв з хмарними платформами дозволяє збирати та аналізувати дані та створювати "розумні" об'єкти, якими можна дистанційно керувати. Такі плати, як Arduino або Raspberry Pi, є дешевими, і кожен може експериментувати з проектами IoT[1].

Платформи Cloud IoT допомагають розробникам швидко і легко створювати та тестувати проекти IoT. Вони надають кілька видів послуг, які можуть бути дуже корисними для проекту IoT:

- *Хмара зберігання даних*
- *Логіка подій*
- *Платформи інтеграції*

Хмарні сховища даних дозволяють розробникам зберігати дані, що надходять з різних приладів. Наприклад, зберігати значення, прочитані датчиком. Ця інформація може бути візуалізована за допомогою графіків або аналізується іншими інструментами[1].

Логіка подій - це веб-логіка програмування, яка може використовуватися для запуску деяких операцій під час події. Використовуючи ці типи платформ, можна реалізувати деяку "бізнес-логіку", використовуючи лише веб-інтерфейс, не знаючи багато про плату, яку ми використовуємо для проекту. Зазвичай логіка виглядає як "IF-THEN", наприклад, якщо відбувається подія, виконайте цю

операцію. Подія може бути сигналом, зчитаним датчиком, а операція може бути електронним листом або SMS.

Інтеграція платформи - це набір "адаптерів", які реалізують певні протоколи, завдяки чому можна без написання коду, змішувати різні інтернет-послуги, щоб створити ланцюжок дій. Наприклад, за допомогою Raspberry Pi можна надсилати сповіщення через SMS, коли значення, прочитане з датчика перевищує порогове значення.

1 КОНЦЕПЦІ РОЗУМНОГО БУДИНКУ. ІНТЕРНЕТ РЕЧЕЙ ТА ХМАРНІ ОБЧИСЛЕННЯ

1.1 Історія розвитку

Сьогодні системи «розумний дім» стали для нас такими ж звичними, як і інші технологічні розробки останнього покоління. Сучасні моделі цих пристроїв оснащені новітнім ПО з інтелектуальним керуванням і великою кількістю різноманітних функцій. Однак щоб отримати настільки досконалий продукт, була виконана величезна робота в різний час і різними людьми[1].

Історія «розумного будинку» почалася в 1961 році, коли Джоель і Рут Спіра винайшли і запатентували спеціальний пристрій для плавного регулювання світла - диммер. Саме цей винахід стало приводом для створення всесвітньо відомої сьогодні компанії *Lutron Electronics Company, Inc.* Дана фірма продовжувала працювати над «розумними» технологіями, паралельно впроваджуючи в ужиток такі поняття, як світлова зона і сцена[1].

Значною подією у подальшому розвитку технології «розумного будинку» було створення шведською компанією *Pico Electronics* побутової автоматики в 1975 році, яку вперше почали використовувати для управління музичними програвачами. Вдосконалили домашню автоматику американці Скотт і Рослін Міллер.

Першим повноцінним проектом «розумного будинку» став невеликий житловий будинок на південному березі Англії. В основу його автоматики лягло використання широкосмугового KNX-системи, що відповідає за управління освітленням, сигналізацією, жалюзі, опаленням і дверима гаража. Також в даному будинку був створений басейн, який згодом доповнили LED-системою з оригінальними кольорними ефектами[1].

У 1987 році організація *ASHRAE* створила новий протокол домашньої

автоматизації, який в подальшому був удосконалений групою компаній Berker, Merten, Insta, Gira, Jung і Siemens і перетворений в абсолютно нову модель автоматики European Installation Bus. У 1999 році на її основі було розроблено нове покоління польових шин KNX[1].

Сучасні системи пішли далеко вперед, істотно розширивши свої технічні можливості. Сьогодні в них використовуються вбудовані домашні кінотеатри, об'єднуються всі інженерні системи, застосовується інтелектуальне управління на основі спеціального ПО. Завдяки модульності системи у користувачів з'явилася можливість самостійно вибирати функціонал «розумного будинку»[1].

1.2 Інтернет речей та хмарні обчислення

Про основні переваги хмарних рішень знають практично все і часто навіть відчували їх на практиці. Розподілена система не має єдиної точки відмови, але при цьому, як правило, має одну точку входу (загальний інтерфейс) для доступу до функціональності. За рахунок цього забезпечується надійність, яка включає в себе наступні фактори:

- доступність сервісу,
- надійність зберігання даних,
- зручність його експлуатації за рахунок використання загального інтерфейсу і відсутності необхідності настройки.

Відповідно в цілому можна розділити хмарні сервіси по типу функціональності на хмарні обчислення і хмарні сховища даних. Для інтернету речей інтерес становлять обидва цих рішення.

Як правило доступність сервісу визначається часом в процентному

співвідношенні від загального розрахункового періоду (як правило, року). Також часто застосовується метрика з кількістю дев'яток в відсотку доступності (від 1 до 9), що характеризує те ж саме, але в більш зручній формі. Резервування даних добре описано на прикладі RAID, однак, на практиці застосовують найрізноманітніші системи. Найчастіше рознесені територіально, щоб знизити вплив зовнішніх факторів на датацентри, а також, щоб знизити час доступу з будь-якої точки світу.

Надійність зберігання і доступність сервісу реалізується за рахунок резервування всіх складових: накопичувачів, обчислювачів та каналів зв'язку. Це є кардинальною відмінністю від пористих мереж, де резервуються тільки канали зв'язку[2].

Історично пропоновані хмарні рішення були орієнтовані на дещо інші аспекти застосування, ніж інтернет речей. У той же час ряд компаній пропонує свої IoT орієнтовані рішення. Розглянемо найбільш поширені рішення серед них.

Існує три найбільш поширених моделі обслуговування хмарних рішень:

- IaaS - інфраструктура як послуга (англ. Infrastructure-as-a-Service),
- PaaS - платформа як послуга (англ. Platform-as-a-Service),
- SaaS - програмне забезпечення як послуга (англ. Software-as-a-Service).

Відрізняються вони наданою функціональністю і відповідно можливостями, а також, як наслідок, необхідними ресурсами на розробку повного рішення. Розглянемо їх по зростанню рівня:

Інфраструктура як послуга (IaaS) надається як можливість використання хмарної інфраструктури для самостійного управління ресурсами обробки,

зберігання, мережами та іншими фундаментальними обчислювальними ресурсами. Найнижчий (апаратний) рівень. Найбільша гнучкість, але і найбільш витратний з розглянутих варіант.

Платформа як послуга (PaaS) надається як можливість використання хмарної інфраструктури для розміщення базового програмного забезпечення для подальшого розміщення на ньому нових або існуючих додатків. Середній (системний) рівень.

Програмне забезпечення як послуга (SaaS) надається як можливість використання прикладного програмного забезпечення провайдера, який працює в хмарній інфраструктурі. Найвищий (прикладної) рівень. Найменш гнучке рішення, але найбільш швидкий вихід на ринок.

Більшість рішень пропонують інфраструктуру і платформу як послугу. При цьому в більшості випадків рішення є чимось середнім між означеними класами. Оскільки пропоновані рішення відносяться до IaaS, то по суті відсутнє обмеження на використання тих чи інших мов програмування і стандартів. Кінцевий розробник сам вільний вибирати рішення яке йому зручно і економічно/технічно обгрунтовано. Цікавою є модель, коли рішення, наприклад, для розумного будинку розробляється на локальній системі, а потім переноситься в хмару. В даному випадку з'являється ряд особливостей, таких, як забезпечення каналів зв'язку між хмарою і пристроями розумного будинку, але практично всі вони вирішуються. Особливим випадком на тлі конкурентів є хмара від IBM, тому що воно надає також послугу SaaS. Що може бути цікаво у випадку швидкого виходу на ринок або орієнтації компанія розробника тільки на деякі зі складових IoT при розробці, зокрема апаратні[2].

Серед усіх розглянутих рішень тільки OpenStack є відкритим. Всі інші пропонують модель із закритою ліцензією, але відкритим API для доступу до сервісів. Однак, тому що компанії провайдери хмарних послуг продають рішення в яких цінністю є самі сервіси, а не ПО, то це не є проблемою. OpenStack ж може бути цікавий для тих, хто захоче зробити свою власну хмару.

1.3 Протокол взаємодії хмарного сервера з пристроями розумного будинку

1.3.1 SOAP

SOAP - протокол обміну структурованими повідомленнями в розподіленій обчислювальній середовищі. Спочатку SOAP призначався в основному для реалізації віддаленого виклику процедур (RPC). Зараз протокол використовується для обміну довільними повідомленнями в форматі XML, а не тільки для виклику процедур. SOAP є розширенням протоколу XML-RPC[2].

SOAP може використовуватися з будь-яким протоколом прикладного рівня: SMTP, FTP, HTTP, HTTPS і ін. Однак його взаємодія з кожним із цих протоколів має свої особливості, які повинні бути визначені окремо. Найчастіше SOAP використовується поверх HTTP.

SOAP є одним зі стандартів, на яких базуються технології веб-служб.

1.3.2 MQTT

MQTT (Message Queue Telemetry Transport) - спрощений мережевий протокол, який працює поверх TCP/IP. Використовується для обміну повідомленнями між пристроями за принципом видавець - підписник.

В його основі лежить проста ідея перенесення ресурсозатратної частини

системи на один елемент у якого ці ресурси є. При цьому всі інші учасники системи майже повністю звільнені від роботи, що дозволяє їм економити ресурси.

Реалізовано це наступним чином: пристрої, які хочуть передати повідомлення (під назвою видавці в термінах MQTT), відправляють його серверу (брокеру), пристрої які хочуть отримати повідомлення (підписники), також підключаються до брокера і отримують повідомлення від нього. Всю обробку повідомлень бере на себе MQTT-брокер. Даний підхід показав себе дуже добре і ефективно в таких завданнях, як, наприклад, збір даних, телеметрія і управління простими пристроями[3].

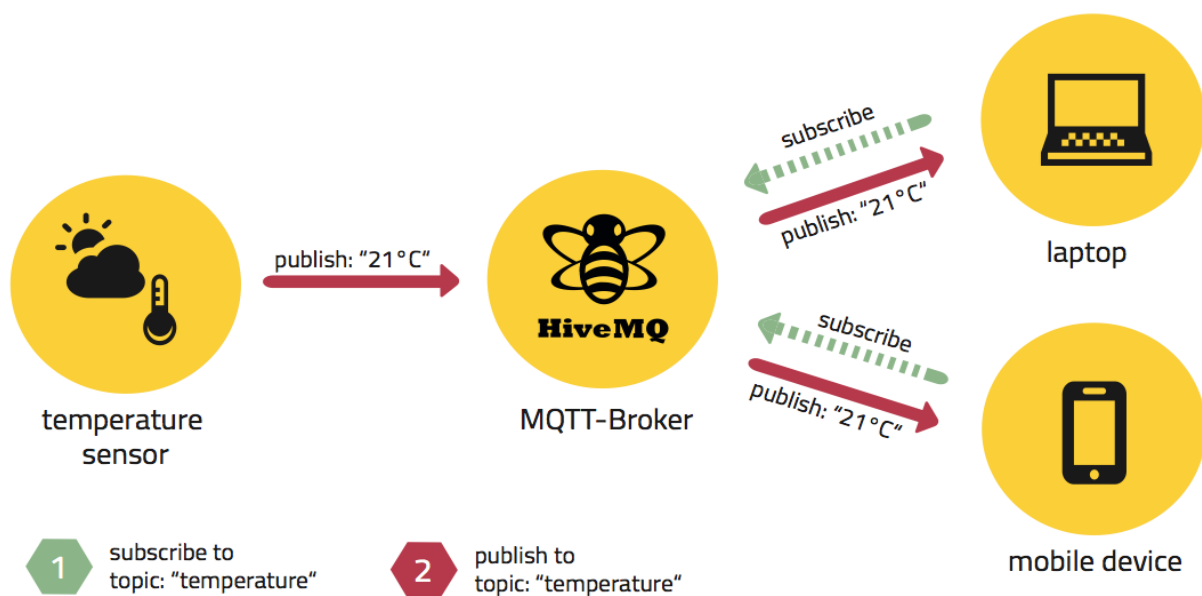


Рисунок 1.1 - Схема роботи MQTT протоколу

Можливості:

- Простий у використанні. Протокол є програмний блок без зайвої функціональності, який може бути легко вбудований в будь-яку складну систему.
- Шаблон проектування видавець-передплатник зручний для більшості рішень з датчиками. Дає можливість пристроям виходити на зв'язок і публікувати повідомлення, які не були заздалегідь відомі або визначені.
- Легкий в адмініструванні.
- Знижено навантаження на канал зв'язку.

- Робота в умовах постійної втрати зв'язку або інших проблем на лінії.
- Немає обмежень на формат переданих даних.

1.4 Висновки до розділу 1

Провівши аналіз історії розвитку та кінцевої концепції розумного будинку та методів передачі даних можна зробити висновок, що саме поняття інтернету речей пройшло довгий та повний оновлень шлях, знайшовши для себе найоптимальнішу концепцію роботи . Але це не крайня межа розвитку. Все більше компаній займаються розробкою пристроїв та сенсорів для розумного дому, щоб повністю автоматизувати його без зайвих зусиль. Саме про них піде мова у наступному розділі.

2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ДЛЯ СТВОРЕННЯ ІНФРАСТРУКТУРИ РОЗУМНОГО БУДИНКУ

2.1 Система домашньої автоматизації Nest

Система домашньої автоматизації Nest першою завоювала популярність, незважаючи на те, що вона не була першою, хто виніс розумний продукт на ринок. Система автоматизації має в своєму комплекті:

1. Nest Learning Thermostat – термостат, що дозволяє ефективно керувати температурою приміщення і тим самим економити електроенергію.
2. Nest Cam IQ – набір камер, що встановлюються як в будинку так і поза ним. Вони не тільки дозволяють фіксувати все, що відбувається за певний проміжок часу і зберігати це в хмарі, але й можуть сповіщати користувача про те хто з жителів помешкання прийшов або покинув будинок, так як Nest сервіс має функцію розпізнавання обличь.
3. Nest Secure Alarm System – складається з набору датчиків та брокера («мізки» системи) Nest Guard, що керує ними. Датчики реагують на рух та відкриття вікон та дверей.
4. Nest Yale – розумний дверний замок, що може бути відкритим введенням секретного коду чи дистанційно за допомогою мобільного додатку.
5. Sensors of Smoke and CO2 Alarm – датчик диму та двоокису вуглецю, може сповістити про небезпеки за допомогою вбудованого динаміка чи через додаток.



Рисунок 2.1 - Стартовий набір системи домашньої автоматизації Nest

Всі елементи системи автоматизації підключаються до Nest сервісу з яким ви можете взаємодіяти через мобільний додаток (отримувати сповіщення, вмикати/вимикати прилади та налаштовувати їх). Система керування має широкий функціонал, що регулярно оновлюється і розширяється. Перевагою використання сервісу є те що, він забезпечує безпеку інтернету речей користувача. Користування сервісом є платним, щомісячна абонплата сягає 50\$.

Nest має відкритий API, що дає можливість налаштовувати взаємодію Nest датчиків з «розумними» речами інших компаній.

2.2 Система домашньої автоматизації SmartThings

Система домашньої автоматизації SmartThings від компанії Samsung.

SmartThings має декілька корисних інструментів: «розумна» розетка, датчик руху (визначає рух), датчик прибуття (виявляє присутність у зоні), датчик витоку води та багатоцільовий датчик (стежить за вібрацією, кутом нахилу та температурою), «розумний» дверний замок, спікер і термостат.



Рисунок 2.2 - Стартовий набір системи домашньої автоматизації SmartThings

Комплект моніторингу Home Monitoring SmartThings - це найдешевший спосіб розпочати роботу. Він поставляється з концентратором, «розумною» розеткою, датчиком руху та двома багатоцільовими датчиками, а також парою аксесуарі.

Взаємодія з компонентами домашньої автоматизації SmartThings відбувається за допомогою концентратора (іноді називається "шлюз" або "домашній контролер"), хмарної платформи та клієнтських програм. SmartThings, як і Nest має відкритий API. Обмін даними між хабом і сенсорами відбувається через безпроводний інтерфейс Z-Wave.

2.3 Система автоматизації освітлення Philips Hue



Рисунок 2.3 - Стартовий набір системи автоматизації освітлення Philips

Philips Hue система була випущена в продаж як перша система освітлення, керована iOS. Розумна лампа обмінюється даними з хабом через протокол Zigbee, який в свою чергу може взаємодіяти з мобільним додатком через Wi-Fi. У початковій системі були лампи, здатні виробляти до 600 люмен, а в нових системах – лампочки виробляють до 800 люмен. Hue може працювати у поєднанні з такими інструментами, як IFTTT або іншими розумними домашніми продуктами. Нова система також має сумісність з HomeKit.

2.4 Система домашньої автоматизації Ecobee

Компанія Ecobee дуже схожа на Nest, тому що її головний продукт «розумний» термостат. В набір домашньої атоматизації входять:

1. Ecobee Switch+ - «розумний» вимикач світла зі вбудованим голосовим сервісом Amazon Alexa.

2. Ecobee Room Sensor – сенсор температури та наявності людей в заданій області, працює разом з Ecobee Thermostat.
3. Ecobee4 – «розумний» термостат із вбудованим голосовим сервісом Amazon Alexa
4. EMS Si - єдиний веб-портал для керування всіма термостатами в будівлі, що під'єднаний до мережі.



Рисунок 2.4 - Стартовий набір системи домашньої автоматизації Ecobee

Ecobee має відкритий API заснований на розширенні фреймворку OAuth 2.0. OAuth 2.0 - нова галузева стандартна схема авторизації, яка забезпечує гарну базу для спільної системи. Сьогодні його використовують деякі з найбільших і найпопулярніших інтернет-сайтів. Специфікація 2.0 все ще розвивається та вдосконалює оригінальну специфікацію, однак це не стосується автентифікації пристроїв. У результаті на багатьох сайтах реалізовано користувальницькі розширення, які використовують механізми розширення, що надаються фреймворком. API Ecobee використовує такі розширення для мінімізації зусиль розробників, але все ще забезпечує глибокий рівень безпеки для кінцевих користувачів.

2.5 Система автоматизації домашньої електроніки Belkin WeMo

WeMo - це серія продуктів від компанії Belkin, яка дозволяє користувачам дистанційно керувати домашньою електронікою. Комплект продукту включає електричні вилки, датчики руху, світлові вимикачі, камери, лампи та мобільні додатки.

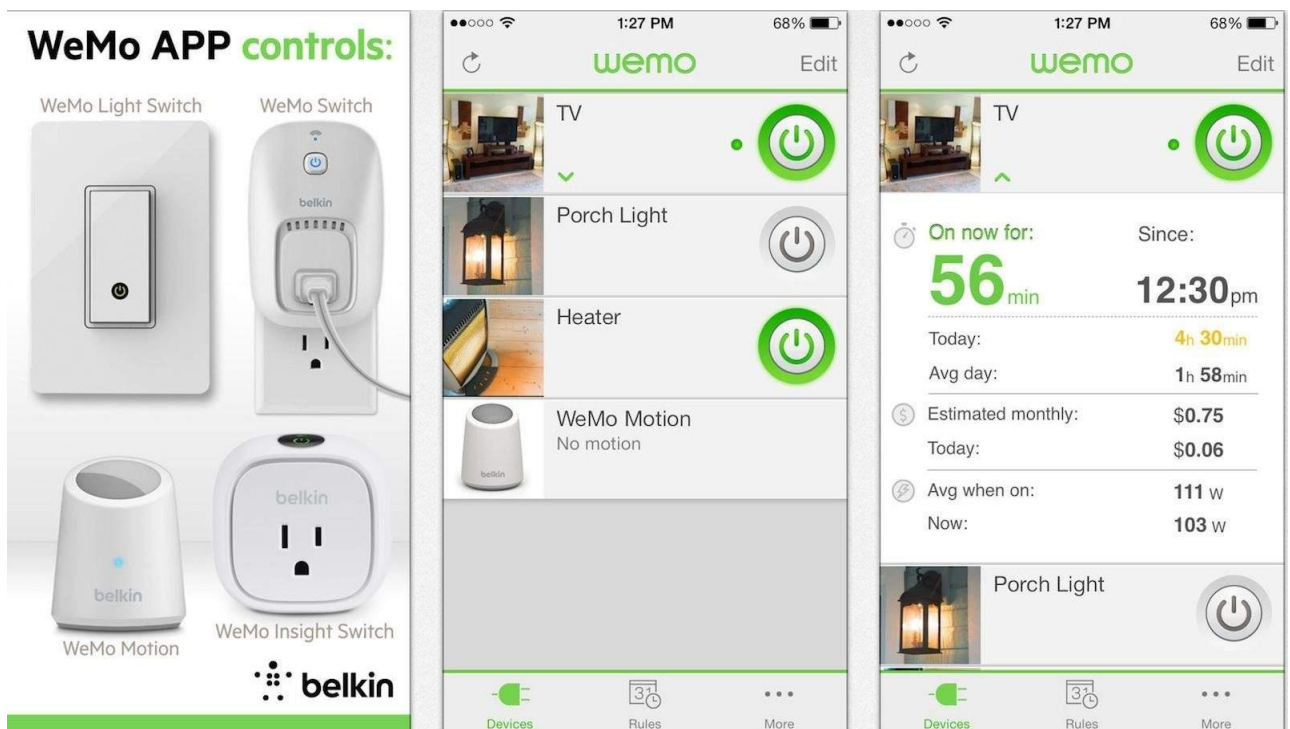


Рисунок 2.5 - Стартовий набір системи домашньої автоматизації WeMo

Перемикач WeMo можна підключити до будь-якої домашньої розетки, якою потім можна керувати зі смартфона iOS або Android, запустивши програму WeMo через домашній Wi-Fi або мережу мобільних телефонів.

Датчик руху WeMo можна розміщувати в будь-якому місці, якщо він може отримати доступ до тієї ж мережі Wi-Fi, що і пристрої WeMo хаб. Після цього він може вмикати та вимикати будь-які пристрої WeMo, підключені до мережі Wi-Fi, реагуючи на наявність мешканців в заданій області.

Перемикач WeMo Insight надає інформацію про використання енергії

приладами, що підключені до нього.

Світловий вимикач WeMo призначений для використання там, де світло керується одним перемикачем світла. Багатофункціональне перемикання не підтримується в даний час, але його можна наблизити, встановивши перемикач WeMo на кожному місці.

Програма WeMo контролює пристрої WeMo з будь-якої точки світу, якщо бездротова мережа WeMo пристроїв підключена до Інтернету. Приладами WeMo також можна керувати за допомогою технології IFTTT. WeMo-пристроями також можна керувати голосом через Amazon Echo, Google Assistant, і Apple Siri (за допомогою Bridge WeMo).

Дана система автоматизації також має відкритий API.

2.6 Система домашньої автоматизації Lutron

Систма автоматизації від компанії Lutron має найбільший вибір девайсів серед всіх вище описаних.

В даний час компанія продає дві версії хаба - Smart Bridge і Smart Bridge Pro, які відрізняються кількістю підтримуваних пристроїв і ціною.



Рисунок 2.6 - Стартовий набір системи домашньої автоматизації Lutron

Smart Bridge одночасно підтримує до 50 пристроїв і сумісний з такими продуктами Lutron, як бездротові диммери Caseta (вбудовані і розеткові), пульти дистанційного керування Pico, а також віддалено керовані штори Serena і Sivola QS Triathlon (версія Smart Bridge Pro). Крім того, новий хаб може працювати з світлодіодними "розумними" лампами GE Telligent, бездротовими термостатами Honeywell, продуктами iPort і Savant. Система може взаємодіяти з професійними рішеннями автоматизації будівель таких компаній, як Remote Technologies Incorporated (RTI), Universal Remote Control (URC) і Control4, а також інтегрується з рядом систем безпеки, зокрема додатком Alarm.com і ELK.

Замість протоколів типу Z-Wave або Zigbee, або навіть WiFi, Smart Bridge використовує протокол Caseta, варіацію власної бездротової системи Clear Connect. Система має суттєві обмеження - всі керовані пристрої повинні знаходитися не далі 9 метрів від хаба.

2.7 Висновки до розділу 2

В даному розділі було проведено оглядовий аналіз існуючих рішень для

створення інфраструктури розумного дому. Розглянуті компанії займають лідерські позиції на ринку Інтернету речей і мають ряд переваг. Кожна з них надає власний веб-сервіс для взаємодії користувача з Інтернет речами і їх обслуговування, що є значною перевагою, але це виключає можливість підключення Інтернет речей від інших виробників. Для рішення цієї проблеми можна використати хмарну IoT платформу.

3 АНАЛІЗ ІСНУЮЧИХ ЗАСОБІВ ДЛЯ РОЗГОРТАННЯ ІОТ

3.1 AWS IoT Core

AWS IoT Core - це керована хмарна платформа, яка дозволяє підключеним пристроям просто і безпечно взаємодіяти з хмарними додатками та іншими пристроями. AWS IoT Core підтримує роботу з мільярдами пристроїв і трильйонами повідомлень, дозволяючи надійно і безпечно обробляти і направляти ці повідомлення до адрес AWS і інших пристроїв[4].

AWS IoT Core спрощує роботу з такими сервісами AWS, як AWS Lambda, Amazon Kinesis, Amazon S3, Amazon Machine Learning, Amazon DynamoDB, Amazon CloudWatch, AWS CloudTrail і Amazon Elasticsearch Service з вбудованими можливостями інтеграції з платформою Kibana, що дозволяє створювати додатки Інтернету речей для збору, обробки, аналізу даних, що генеруються підключеними пристроями, а також виконання дій на основі цих даних без необхідності керувати будь-якою інфраструктурою[4].

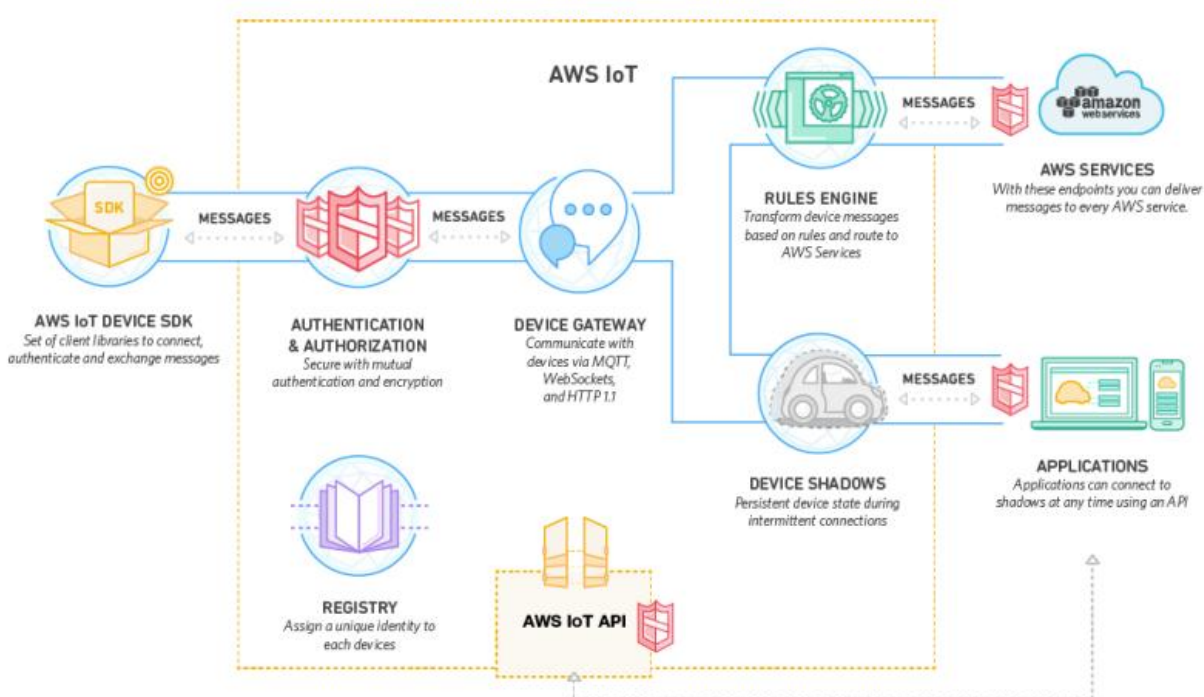


Рисунок 3.1 - Схема принципу роботи сервісу AWS IoT Core

3.1.1 Підключення пристроїв та управління ними

AWS IoT Core дозволяє легко підключати пристрої до хмари або один до одного. AWS IoT Core підтримує протоколи HTTP, WebSockets і спрощений протокол зв'язку MQTT, спеціально спроектований для підтримки нестабільного підключення, скорочення обсягу коду, переданого пристрою, і роботи в мережах з низькою пропускнуою здатністю. AWS IoT Core також підтримує інші стандартні і призначені для користувача протоколи, забезпечуючи взаємодію між пристроями навіть в разі використання ними різних протоколів[4].

3.1.2 Безпека підключення і даних

AWS IoT Core забезпечує аутентифікацію і наскрізне шифрування між усіма точками підключення. Таким чином, будь-який обмін даними між пристроями і AWS IoT Core відбувається тільки після перевірки ідентифікації. Крім того, можна забезпечити безпечний доступ до своїх пристроїв і додатків, застосовуючи політики з точним настроюванням дозволів.

3.1.3 Обробка даних пристроїв

За допомогою AWS IoT Core можна фільтрувати дані пристроїв, перетворювати їх на льоту і виконувати різні пов'язані дії на основі певних користувачем правил роботи. Ці правила можна оновити в будь-який момент з урахуванням нових пристроїв або можливостей програми. AWS IoT Core дозволяє з легкістю створювати ще більш потужні IoT-додатки з використанням таких сервісів AWS, як AWS Lambda, Amazon Kinesis, Amazon S3, Amazon Machine Learning, Amazon DynamoDB, Amazon CloudWatch і Amazon Elasticsearch Service.

AWS IoT Core зберігає останній стан пристрою, що дозволяє в будь-який момент зчитати або задати цей стан. Завдяки цьому додатку пристрій визначають так, як ніби він постійно підключений. Це означає, що додаток може зчитувати стан

пристрою, навіть коли він працює автономно, а також дозволяє задати стан пристрою, який буде застосовано при наступному його підключенні.

Основні моменти:

- Підтримка HTTP, WebSockets та MQTT.
- Правила Engine може відправляти повідомлення до кінцевих точок AWS, включаючи AWS Lambda, Amazon Kinesis, Amazon S3, Amazon Machine Learning, Amazon DynamoDB, Amazon CloudWatch та службу Amazon Elasticsearch з вбудованою інтеграцією Kibana.
- Створіть стійку, віртуальну або "тіньову" версію кожного пристрою, що включає останній стан пристрою.

3.2 Google Cloud IoT

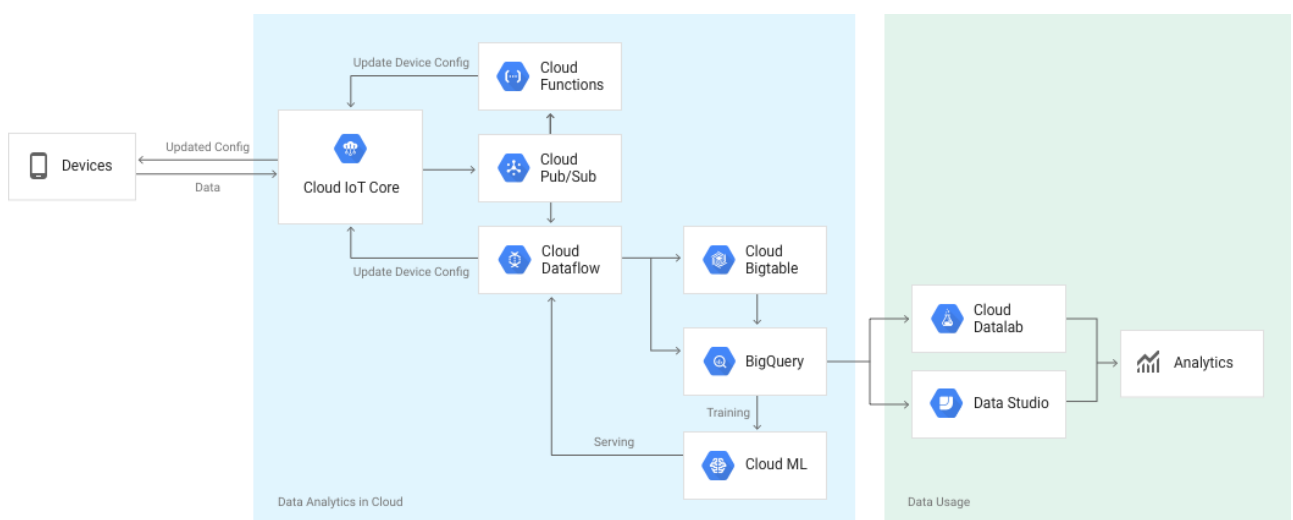


Рисунок 3.2 - Схема принципу роботи сервісу Google Cloud IoT Core

Cloud IoT Core має дві основні компоненти: диспетчер пристроїв і міст протоколу[5].

Диспетчер пристроїв дозволяє окремо налаштувати та керувати окремими

пристроями грубозернистим способом, управління може бути зроблено через консоль або програмно. Менеджер пристроїв встановлює ідентичність пристрою та забезпечує механізм автентифікації пристрою під час підключення. Він також підтримує логічну конфігурацію кожного пристрою і може використовуватися для дистанційного керування пристроєм з хмари[5].

Міст протоколу забезпечує кінцеві точки з'єднання для протоколів з автоматичним балансуванням навантаження для всіх з'єднань пристроїв. Міст протоколу має власну підтримку для безпечного з'єднання за галузевими стандартними протоколами, такими як MQTT та HTTP. Міст протоколу публікує всю телеметрію пристрою Cloud Pub / Sub, яка потім може послідовно споживатися аналітичними системами.

3.2.1 Кінцевий рівень безпеки

Безпека є серйозною проблемою при розгортанні та управлінні пристроями IoT. Cloud IoT Core пропонує наступні функції безпеки:

- Аутентифікація загальнодоступного / приватного ключа за допомогою JSON Web Tokens (JWTs, RFC 7519). Це обмежує площу зони атаки, оскільки компромісний ключ вплине лише на один пристрій, а не на всю систему.
- Підтримка алгоритмів RSA або еліптичних кривих для перевірки підписів, із застосуванням для надійних розмірів ключів.
- Підтримка обороту ключів в пристрої, що дозволяє зареєструвати паралельні ключі та підтримувати термін придатності для кожного облікового запису.
- TLS 1.2 підключення, використовуючи кореневі сертифікати (потрібні для MQTT).

- Доступ до Cloud IoT Core API здійснюється за допомогою правил і дозволів Cloud Identity and Access Management (IAM)[5].

3.2.2 Протоколи передачі даних

Cloud IoT Core підтримує два протоколи для з'єднання та зв'язку пристрою: MQTT та HTTP. Пристрої взаємодіють із Cloud IoT Core через "міст" - це міст MQTT або міст HTTP. Міст MQTT / HTTP є центральним компонентом Cloud IoT Core.

Коли користувач створює реєстр пристроїв, він може обрати протоколи для активації: MQTT, HTTP або обидва.

MQTT - це стандартний протокол опублікування / підписки, який часто використовується та підтримується вбудованими пристроями, а також часто зустрічається у взаємодії між машиною та комп'ютером[5].

HTTP - це протокол "без з'єднання": пристрої не підтримують зв'язок з Cloud IoT Core через HTTP міст. Замість цього вони надсилають запити та отримують відповіді. Cloud IoT Core підтримує тільки HTTP 1.1 (не 2.0).

У наведеній нижче таблиці порівнюється, як два протоколи працюють в Cloud IoT Core.

Таблиця 2.1 Протоколів передачі даних

MQTT	HTTP
Підтримується підключення пристроїв	Бездротовий зв'язок (запит / відповідь)
Повнодуплексне TCP з'єднання	Напівдуплексне TCP з'єднання
JWT надсилається у поле пароля в повідомленні про підключення	JWT надсилається у заголовок HTTP запиту
Події телеметрії записуються до Cloud Pub / Sub	Події телеметрії записуються до Cloud Pub / Sub
Повідомляє про статус підключення пристрою	Не повідомляє про статус підключення пристрою
Конфігурації пристрою поширюються через підписку	Необхідний запит про конфігурації приладу
Найновіша конфігурація завжди	Пристрої можуть вказати, що можуть

приймається пристроями підписки	бути отримані лише нові конфігурації
Конфігурації пристрою визнаються (ACKed) при використанні QoS 1	Немає явного АСК для конфігурації пристрою
Зберігаються останні дані, що були надіслані пристроєм	Не зберігаються останні дані, що були надіслані пристроєм

3.2.3 Основні моменти

- Спадщини Google у галузі веб-масштабу обробки, аналітики та машинного розвідки.
- Використовує глобальну волоконну мережу Google (70 точок присутності в 33 країнах) для наднизької затримки.

3.3 Microsoft Azure IoT Suite

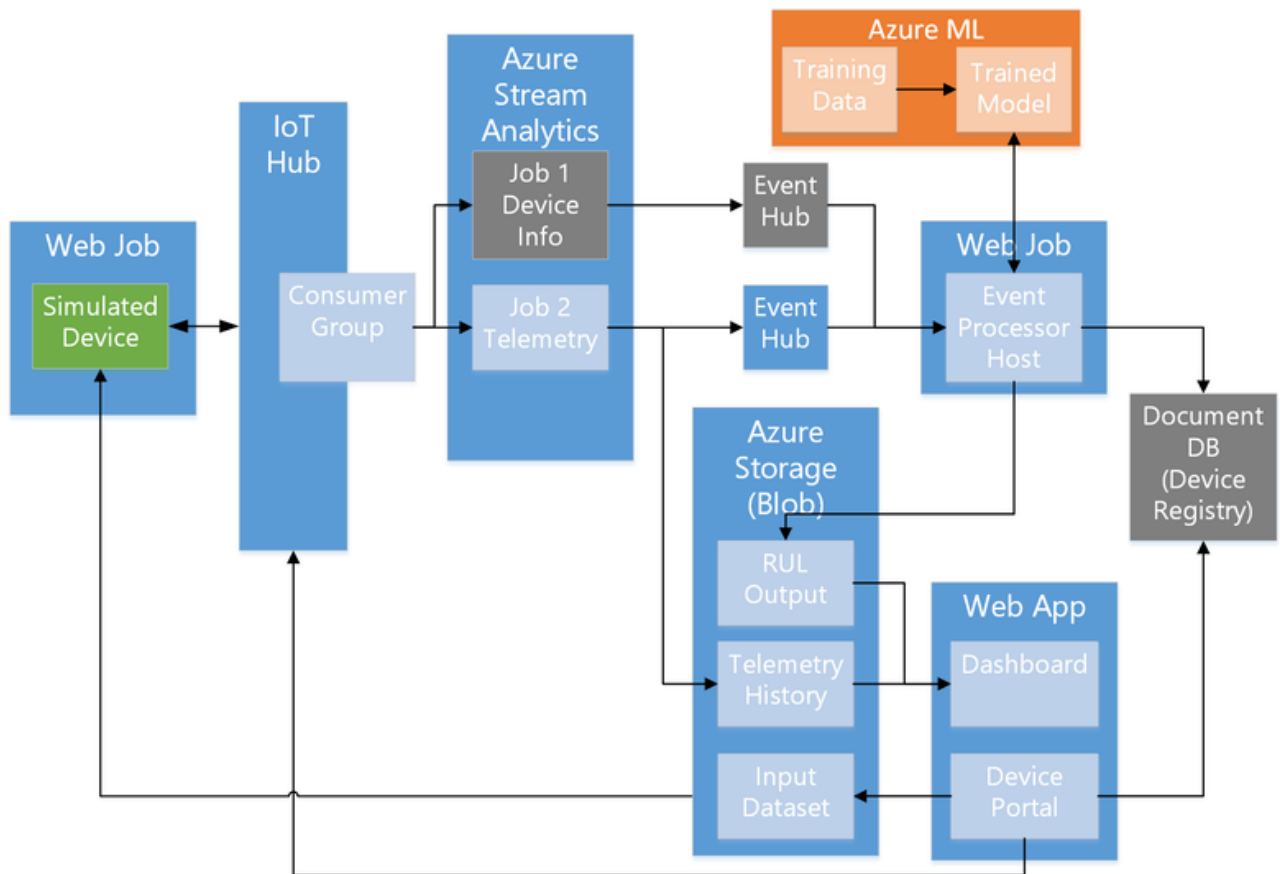


Рисунок 3.3 - Схема принципу роботи сервісу Azure IoT Suite

Рішення являє собою набір хмарних сервісів, інтегрованих із середовищем Azure і дозволяють будь-яким компаніям запускати власні IoT-проекти. Azure IoT Suite дозволяє підключити до хмари Microsoft Azure найрізноманітніші електронні вироби з мережевою функціональністю, отримувати від них підлягають управлінню, зберігання та аналізу дані, які можуть бути використані для прийняття важливих бізнес-рішень і автоматизації процесів. Набір Microsoft Azure IoT - це рішення корпоративного рівня, яке дозволяє швидко приступити до роботи за допомогою набору розширюваних попередньо налаштованих рішень. Ці рішення підходять для поширених сценаріїв Інтернету речей, таких як віддалений моніторинг та прогнозне (випереджаюче) обслуговування[6].

Попередньо налаштовані рішення є повними, які працюють, комплексними рішеннями, які включають в себе наступне:

- віртуальні пристрої, необхідні для початку роботи;
- попередньо налаштовані служби Azure;

- консолі управління певного рішення.

Попередньо налаштовані рішення містять перевірений, готовий до роботи код, який можна налаштовувати і розширювати для реалізації певних сценаріїв IoT.

Основою пакету Azure IoT Suite є служба Центр Інтернету речей Azure. Ця служба надає можливість обміну повідомленнями між пристроями і хмарою, а також виступає в якості шлюзу для хмари і інших ключових служб IoT Suite. Служба дозволяє отримувати повідомлення від пристроїв в реальному часі і відправляти команди на пристрої. Також служба дозволяє управляти пристроями[6].

Azure Stream Analytics забезпечує оперативний аналіз даних: обробка вхідних даних телеметрії, агрегування і виявлення подій. Попередньо налаштовані рішення також використовують Stream Analytics для обробки інформаційних повідомлень, що містять, наприклад, метадані або відповіді на команди від пристроїв. Ці рішення використовують Stream Analytics для обробки повідомлень від пристроїв і доставки цих повідомлень іншим службам.

Можливості зберігання даних надаються службою Azure DocumentDB. Для зберігання даних телеметрії і їх підготовки до аналізу попередньо налаштовані рішення використовують сховище BLOB-об'єктів. Ці рішення використовують DocumentDB для зберігання метаданих пристрою і включають можливості керування пристроями для рішень.

Можливості візуалізації даних надаються веб-додатками Microsoft Power BI. Гнучкість Power BI дозволяє швидко створювати інтерактивні панелі моніторингу на основі даних з пакета IoT Suite[6].

3.3.1 Безпека Azure IoT Suite

Microsoft Azure IoT Suite містить вбудовані компоненти забезпечення безпеки і конфіденційності платформи Azure, а також процеси SDL і OSA - все це використовується для безпечної розробки та експлуатації програмного забезпечення Microsoft. За допомогою цих процедур реалізовані захист інфраструктури і мережі, а

також функції ідентифікації та управління, що мають ключове значення для забезпечення безпеки в рамках будь-якого рішення.

Компонент Azure IoT Hub в складі IoT Suite - це повністю керована служба, яка забезпечує надійний захист двостороннього обміну даними між пристроями IoT і службами Azure (наприклад, службами машинного навчання Azure і Azure Stream Analytics) за рахунок використання індивідуальних облікових даних для кожного конкретного пристрою і контролю доступу до нього[6].

Щоб найбільш повно представити вбудовані компоненти забезпечення безпеки і конфіденційності Azure IoT Suite, розглядаємо програмний комплекс в розрізі трьох головних областей безпеки.

3.3.2 Захищена підготовка пристроїв і перевірка справжності

Azure IoT Suite забезпечує захист пристроїв на час їх роботи на місцях. Для цього на кожному пристрої налаштований унікальний ключ посвідчення, який інфраструктура використовує для зв'язку з працюючим пристроєм. Налаштування цієї процедури не відрізняється складністю і не забирає багато часу. На базі ключа з ідентифікатором пристрою, заданим користувачем, створюється токен, який згодом використовується у всіх сенсах передачі даних між пристроєм і службою Azure IoT Hub.

3.3.3 Захищене з'єднання

Безперервний і стійкий обмін даними має критично важливе значення для будь-якого рішення IoT. Особливо важлива безперервність доставки команд і отримання даних від пристрою, враховуючи, що пристрої IoT підключаються через Інтернет або інші аналогічні мережі, і таке підключення може виявитися ненадійним. Використовуючи систему відправки підтверджень для отриманих повідомлень, Azure IoT Hub забезпечує необхідну стійкість при обміні даними між хмарної середовищем і пристроєм. Завдяки кешування повідомлень в службі IoT

Hub підвищується стійкість при обміні повідомленнями. Кешовані повідомлення зберігаються протягом семи (телеметрія) або двох днів (команди).

Вкрай важливо забезпечити ефективність при консервації ресурсів і роботі в середовищі з обмеженими ресурсами. HTTPS (HTTP Secure) - це відповідна галузевим стандартам захищена версія популярного протоколу HTTP. У Azure IoT Hub протокол HTTPS відповідає за максимально ефективний обмін даними. Протоколи AMQP і MQTT, які також підтримуються службою Azure IoT Hub, не тільки забезпечують ефективність в плані споживання ресурсів, а й гарантують надійну доставку повідомлень. [1]

3.3.4 Безпечна обробка і зберігання в хмарі

Azure IoT Hub гарантує комплексний захист на всіх рівнях передачі даних - від шифрування до обробки в хмарі. Служба забезпечує належну гнучкість і адаптивність, дозволяючи застосовувати додаткові методи шифрування ключів безпеки і розширені функції управління ключами. Для перевірки справжності та авторизації користувачів в Azure IoT Hub використовується Azure Active Directory (AAD). У Azure IoT Suite реалізована модель авторизації даних в хмарі на основі політик, яка спрощує управління доступом і надає можливість перевірки та аудиту. Вона дозволяє практично миттєво відкликати дозволи на доступ до даних в хмарі і блокувати пристрої, підключені до Azure IoT Suite. [6]

Передані в хмару дані можна обробляти і зберігати в будь-якому робочому процесі, визначеному користувачем. Azure Active Directory управляє доступом до різних розділів даних в залежності від використовуваної служби зберігання даних.

Основні моменти

- Легко інтегрується Azure IoT Suite із системами та додатками користувача, включаючи Salesforce, SAP, Oracle Database та Microsoft Dynamics.
- Azure IoT Suite комплектує послуги Azure IoT з попередньо налаштованими рішеннями.

- Підтримка протоколу HTTP, Advanced Message Queuing Protocol (AMQP) та MQ Telemetry Transport (MQTT).
- SDK Gateway.

3.4 IBM Watson IoT

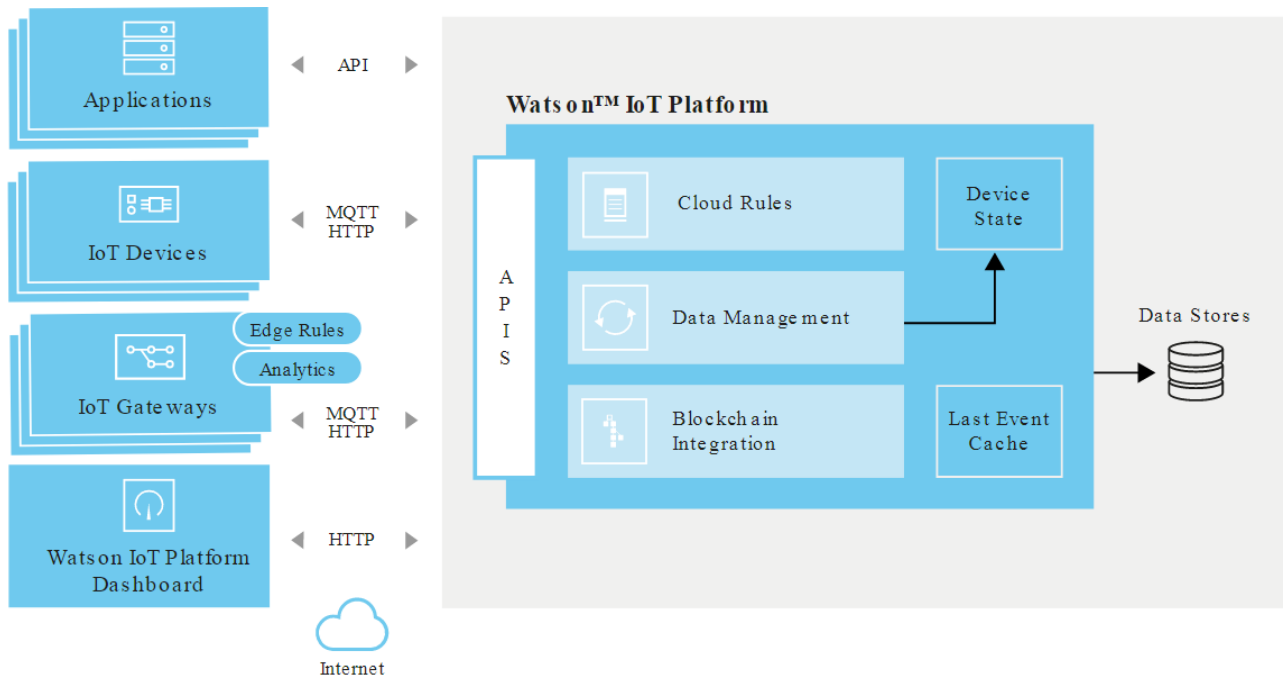


Рисунок 3.4 - Схема принципу роботи сервісу IBM Watson IoT

Watson IoT Platform Connect - це відправна точка для будь-якої служби Watson IoT Platform. Підключення пристроїв, створення додатків, керування пристроями та взаємодія з сторонніми сервісами доступні в Watson IoT Platform Connect[7].

3.4.1 Шлюзові пристрої

Використовуючи шлюз, ви можете підключати пристрої до Watson IoT Platform, які в іншому випадку не зможуть підключитися до Інтернету. Шлюзи мають всі функції пристрою, але також можуть публікувати та підписатись від імені підключених до них пристроїв. Пристрої шлюзу дозволяють групам датчиків, які не можуть підключитися до Інтернету, підключитись до платформи Watson IoT, відправляючи свої дані на шлюз[7].

Шлюзи підключаються до Watson IoT Platform за допомогою протоколу MQTT або протоколу HTTP для обміну повідомленнями. Клієнтські бібліотеки для розробки шлюзів, які можуть підключатися до платформи Watson IoT, доступні на таких мовах:

- C++
- Embedded C
- Java
- Mbed C++
- Node.js
- Node-RED
- Python

3.4.2 Управління пристроєм

Можливості керування пристроями забезпечуються через API керування пристроєм та агентом керування пристроєм, який встановлений на пристроях. Пристрої з агентом керування можуть виконувати дії, які можна викликати через основну інформаційну панель Watson IoT Platform або API керування пристроєм. Дії з керування пристроєм включають перезавантаження, завантаження та встановлення оновлень прошивки та відновлення пристроїв до заводських налаштувань. Управління пристроями також може бути розширено, щоб включати власні дії керування пристроєм[7].

3.4.3 Розширення та інтеграція служб

Розширення та інтеграція служб включають як зовнішні служби, так і визначені користувачем розширення основних служб, які слід додати до Watson IoT Platform. Зовнішні служби, які можуть бути інтегровані з платформою Watson IoT, включають служби погодних умов розташування, що дозволяє знаходити поточну

погоду на місці пристрою та дані SIM-карти Jasper[7].

3.4.4 Зберігання даних пристрою

Використовуючи Watson IoT Platform Last Event Cache API, можна отримати дані про останню подію, надіслану пристроєм. Це дозволяє визначити, чи є пристрій в режимі онлайн або офлайн, що дозволяє відслідковувати стан пристрою незалежно від фізичного місцезнаходження пристрою. Останню інформацію про подію, надіслану пристроєм можна отримати для будь-якої конкретної події, яка сталася протягом останніх 45 днів.

Дані про події пристрою можна зберігати для подальшого використання. Зберігання даних є важливим першим кроком на шляху до проникливої аналітики, щоб отримати статистику з цих даних. Наприклад, відстеження змін протягом більш тривалого періоду часу та зберігання наборів даних для використання з потужними інструментами аналітики, включаючи використання Watson API та когнітивні обчислення.

3.4.5 Управління даними

Різні макети та моделі пристроїв публікують дані в різних форматах. Функція керування даними дозволяє перетворити та нормалізувати ці дані в єдину логічну схему, яка називається стан пристрою. Використання функції керування даними значно спрощує розробку додатків, оскільки програмі більше не потрібно розуміти різні формати даних події, які надсилаються з кожного пристрою. Коли пристрої публікують події на Watson IoT Platform, вміст подій може бути зіставлений з визначеними користувачем властивостями держави, використовуючи відображення. Якщо вхідна подія призводить до зміни стану пристрою, значення властивостей пристрою оновлюються та зберігаються у платформі Watson IoT. Ці значення доступні для програми за запитом за допомогою HTTP API або підписавшись на тему[7].

3.4.6 Управління ризиками

Пристрої підключаються до платформи Watson IoT, використовуючи комбінацію ідентифікатора клієнта та токена автентифікації, які знає лише користувач. Після реєстрації пристроїв або створення ключів API, маркер автентифікації хешується для забезпечення безпеки облікових даних.

Додаток для управління ризиком та безпекою дає змогу покращити безпеку Watson IoT Platform, щоб забезпечити автентичність усіх точок з'єднання між сервером та пристроями за допомогою перевірених облікових даних. За допомогою цього додатка виконується автентифікація сертифікатів та захист транспортного рівня (TLS). Під час обміну даними між пристроями та сервером будь-які пристрої, які не мають дійсних сертифікатів із доступом до сервера, і налаштовуються у додатку управління ризиком та безпекою, не мають доступу, навіть якщо вони використовують дійсні ідентифікатори та паролі.

3.4.7 Основні моменти

- Machine Learning - автоматизація обробки даних та ранжування даних на основі вивчених пріоритетів.
- Raspberry Pi Support – можливість розробки IoT додатків для Raspberry Pi, API.
- Real-Time Insights - контекстуально аналіз IoT даних в режимі реального часу.

3.5 ThingSpeak Platform

ThingSpeak - це платформа Internet of Things (IoT), яка дозволяє збирати та зберігати дані датчиків у хмарі та розробляти IoT програми. Платформа ThingSpeak IoT забезпечує додатки, які дозволяють аналізувати та візуалізувати ваші дані в MATLAB. Дані датчика можуть бути відправлені до ThingSpeakплатформи від Arduino, Raspberry Pi, BeagleBone Black та інших апаратних засобів[8].

Датчики, або речі, реєструють дані і, як правило, діють локально. ThingSpeak дає змогу датчикам та іншим приладам надсилати дані в хмару, де вони зберігається як в приватних, так і в загальнодоступних каналах. За замовчуванням ThingSpeak зберігає дані на приватних каналах, однак публічні канали можуть використовуватися для обміну даними з іншими. Коли дані перебувають у каналі ThingSpeak, користувач може проаналізувати та візуалізувати їх, обчислити нові дані або взаємодіяти з соціальними мережами, веб-службами та іншими пристроями.

3.5.1 Заявлений ункціонал сервісу

Збір даних, тобто можливість "прив'язки" пристрою до сервісу, пристрій відправляє дані на сервер ThingSpeak, за допомогою відкритого API, сервер зберігає дані в базу.

Аналіз і візуалізація даних. Всі дані, отримані від пристроїв піддаються аналізу. Користувач може виявити залежності, шаблони і тренди в даних. За інформацією, отриманою від пристроїв можуть генеруватися нові дані. Дані візуалізуються за допомогою графіків, діаграм, а також елементів інтерфейсу у вигляді вимірювальних приладів (наприклад, термометр і т.д).

ThingSpeak надає користувачеві доступ до функціональності MATLAB, завдяки чому користувач може конвертувати і обробляти отримані дані, генерувати з них нових дані, планувати обчислення і аналіз даних, комбінувати дані з декількох каналів, візуалізувати даніх[8].

Реагування. Користувач може задати правило реагування на дані, отримані від пристрою. Наприклад, якщо температура на термометрі перевищить 70 градусів, користувач отримає твіт від системи, що попереджає його про це.

Користувач може також будувати більш складні правила реагування і дій при настанні певної ситуації. Наприклад, користувач вказує необхідність відключення мотора тоді, коли рівень води впаде нижче певного порогу. ThingSpeak надає можливість для "реагування" на отримані або ж уже на оброблені дані, можливість

вибудовування ланцюжка дій для будь-якого пристрою[8].

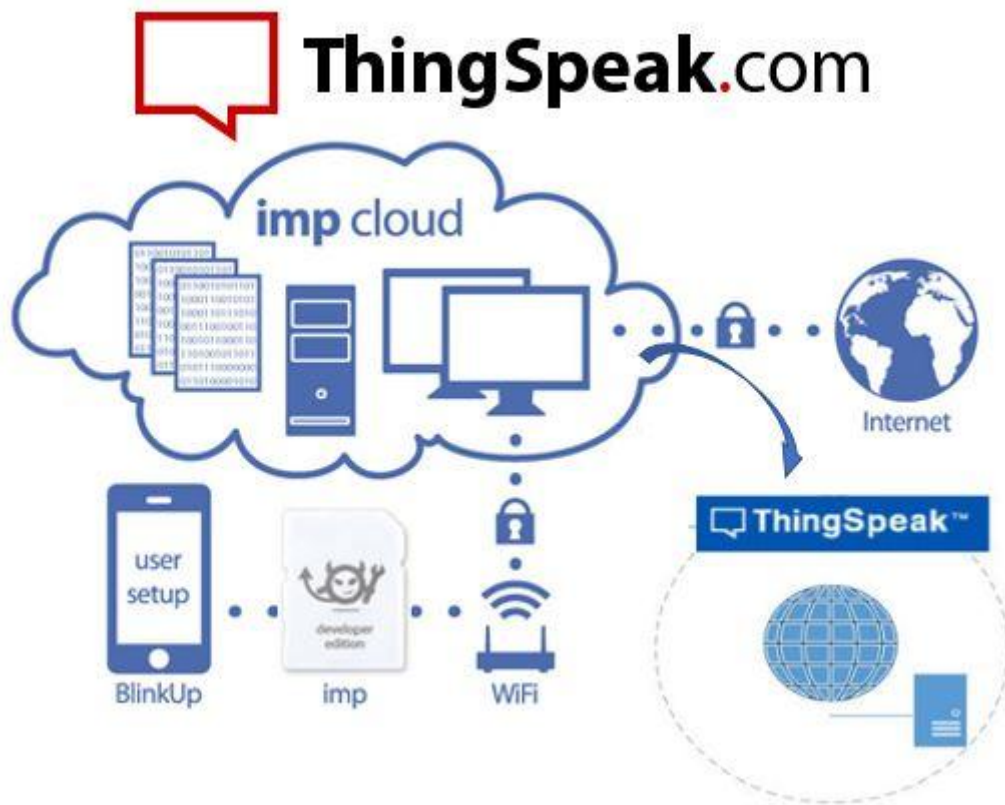


Рисунок 3.5 - марна платформа ThingSpeak

3.5.2 Основні моменти

- Безкоштовний реєстрація
- Аналіз та візуалізація даних за допомогою MATLAB, RESTful та MQTT API.
- Підтримка Arduino, Raspberry Pi, BeagleBone Black та інших апаратних засобів.

3.6 Висновки до розділу 3

В даному розділі були розглянуті хмарні платформи керування розумним будинком, такі як: Microsoft Azure IoT Suite, AWS IoT, ThingSpeak, Google Cloud IoT, IBM Watson IoT.

В аспекті безпеки передачі даних розглянуті платформ являються однаково

надійними. В них реалізовані передові технології аутентифікації користувачів і пристроїв, а також шифруванні даних. В аспекті візуалізації, моніторингу і та аналізу даних всі п'ять платформ надають подібний набір засобів для адміністрування та налаштування датчиків.

Також всі висвітлені платформи надають широкий вибір пристроїв, що можуть бути підключеними.

Тому ключовим фактором для вибору конкретної платформи є достатньо вільний доступ до ресурсів платформи, де лідирує ThingSpeak.

4 Розробка онлайн калькулятора для планування розумного будинку

В попередньому розділі були проаналізовані одні з найпоширеніших хмарних платформ Інтернету речей, і на основі нього для реалізації поставленої задачі, було обрано ThingSpeak. А базою для безпосереднього розгортання IoT в будинку було обрано Raspberry Pi 3 B+. Для створення веб-додатку було прийнято рішення використовувати платформу Node.JS та фреймворк Express, а використовуватися при написанні конфігураційних скриптів для обладнання.

4.1 Raspberry Pi

Хоча Raspberry Pi настільки ж маленька, як і розмір кредитної картки, вона працює як звичайний комп'ютер за відносно низькою ціною. можна працювати як недорогий сервер для обробки легкого внутрішнього або веб-трафіку. Використання Raspberry Pi для роботи в якості сервера є більш економічним, ніж звичайний сервер. Якщо всі легкі сервери трафіку будуть замінені на Raspberry Pi, це може мінімізувати бюджет підприємства[9].

Переваги Raspberry Pi:

- Вартість.
- Доступність.
- Досить велика база знань для RPi.
- Багато додатків вже доступні для RPi.
- Розмір і портативність.
- Безперервне вдосконалення і здатність завантажувати звичайну ОС Linux і існує багато дистрибутивів Linux виключно для RPi і Windows 10 IoT Core також можна завантажити в RPi

Характеристики Raspberry Pi:

- Процесор: Broadcom BCM2837
- Кількість ядер процесора: 4
- Частота процесора: 1200 МГц
- Тип оперативної пам'яті: SDRAM
- Розмір оперативної пам'яті: 1 Гб
- Відеочіп: Broadcom Videocore 4
- Обсяг відеопам'яті: виділяється з оперативної
- Відео інтерфейси: HDMI, MIPI (CSI) вхід
- Аудіо інтерфейси: 3.5 мм jack (аудіо)
- Інтерфейси периферії: USB 2.0 x4, GPIO, micro USB, micro SD
- Вид доступу в Інтернет: Ethernet (RJ-45), Wi-Fi
- Швидкість мережевого адаптера: 100 Мбіт / с
- Додаткове вбудоване обладнання: Bluetooth 4.1[10].

01	3.3v DC Power			DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)			DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)			Ground	06
07	GPIO04 (GPIO_GCLK)			(TXD0) GPIO14	08
09	Ground			(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)			(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)			Ground	14
15	GPIO22 (GPIO_GEN3)			(GPIO_GEN4) GPIO23	16
17	3.3v DC Power			(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)			Ground	20
21	GPIO09 (SPI_MISO)			(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)			(SPI_CE0_N) GPIO08	24
25	Ground			(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)			(I ² C ID EEPROM) ID_SC	28
29	GPIO05			Ground	30
31	GPIO06			GPIO12	32
33	GPIO13			Ground	34
35	GPIO19			GPIO16	36
37	GPIO26			GPIO20	38
39	Ground			GPIO21	40

Рисунок 4.1 - Схема інтерфейсу GPIO для Raspberry Pi 3 B+

4.2 Node.JS

Node.js дозволяє створювати веб-сервери та мережеві інструменти, використовуючи JavaScript і набір "модулів", які обробляють різні функціональні можливості. Модулі передбачають введення / виводу файлової системи, мережеві (DNS, HTTP, TCP, TLS / SSL або UDP), двійкові дані (буфери), функції криптографії, потоки даних та інші основні функції. Модулі Node.js використовують API, призначений для зменшення складності написання серверних додатків[11].

Програми Node.js можуть працювати на серверах Linux, macOS, Microsoft Windows, NonStop та Unix. Крім того, вони можуть бути написані з CoffeeScript (альтернатива JavaScript), Dart або TypeScript (строго типізовані форми JavaScript) або будь-яка інша мова, яку можна скомпілювати в JavaScript[11].

Node.js використовується, перш за все, для побудови мережевих програм, таких як веб-сервери. Найбільша різниця між Node.js і PHP полягає в тому, що більшість функцій в блоці PHP до завершення (команди виконуються лише після завершення попередніх команд), а функції Node.js не блокуються (команди виконуються одночасно або навіть паралельно, і використовують зворотню відповідь завершення сигналу) [11].

4.3 Python

Python - це багатопарадигмальна мова програмування. Об'єктно-орієнтоване програмування та структуроване програмування повністю підтримуються, і багато його функцій підтримують функціональне програмування та аспектно-орієнтоване програмування (у тому числі метапрограмування та метаоб'єкти. Багато інших парадигм підтримуються через розширення, включаючи дизайн за контрактом та логічне програмування[12].

Python використовує динамічну типізацію та в комбінації підрахунок посилань та збирач сміття для керування пам'яттю. Він також має динамічне розпізнавання імен

(late binding), який пов'язує імена методів та імена змінних під час виконання програми.

Дизайн Python пропонує деяку підтримку функціонального програмування в традиції Lisp. Вона має функції `filter ()`, `map ()` і `reduce ()`; перелік виразів, словників та наборів; і вирази генератора. Стандартна бібліотека має два модулі (`itertools` and `functools`), які реалізують функціональні інструменти, запозичені з Haskell і Standard ML.

Основна філософія мови узагальнена в документі Zen of Python (PEP 20), що включає афоризми, такі як:

- Красивий краще, ніж потворний
- Явний є кращим, ніж неявний
- Простий краще, ніж складний
- Комплекс краще, ніж складний
- Підрахунок читабельності

Замість того, щоб всі свої функціональні можливості були вбудовані в ядро, мова програмування Python була розроблений так, щоб бути дуже розширюваною. Ця компактна модульність зробила її особливо популярною як засіб для додавання програмованих інтерфейсів до існуючих програм. Погляд Ван Россума на маленьку основну мову з великою стандартною бібліотекою та легко розширюваним інтерпретатором впливав із його розладів з ABC, що супроводжувала протилежний підхід[12].

Пропонуючи вибір методології кодування, філософія Python відхиляє надмірний синтаксис (наприклад, Perl) на користь простішої, менш захащеної граматики. Як зазначив Олексій Мартеллі: "Охарактеризувати щось" розумне "не вважається компліментом у культурі Пітона". Філософія Python відкидає Perl "є більш ніж один спосіб це зробити" підхід до мовного дизайну на користь "повинен бути один і, бажано, лише один-очевидний спосіб зробити це"[12].

Розробники Python прагнуть уникати передчасної оптимізації та відмовляються від патчів до некритичних частин CPython, що забезпечить граничне збільшення

швидкості за ціною прозорості. Коли важлива швидкість, програміст Python може переміщати критично важливі функції до модулів розширень, написаних мовами, такими як C, або використовувати PyPy, компілятор, який працює у режимі реального часу. Також доступний Cython, який перетворює скрипт Python на C і робить прямі виклики C-рівня в інтерпретаторі Python[12].

4.4 Вимоги до системи

Система має являти собою онлайн калькулятор для планування інфраструктури розумного будинку і реалізована як веб-застосунок. Початкову задачу можна розбити на три етапи: створення веб-інтерфейсу для користувача, створення бази даних Інтернет речей та створення системи для автоматизованого розгортання інфраструктури розумного будинку.

В попередньому розділі були проаналізовані одні з найпоширеніших хмарних платформ Інтернету речей, і на основі нього для реалізації поставленої задачі, було обрано ThingSpeak.

А базою для безпосереднього розгортання IoT в будинку було обрано Raspberry Pi.

Основні переваги

Загальний функціонал веб-застосунку:

- *Створення облікового запису користувача.*
- *Можливість додавання до бази даних нових пристроїв.*
- *Реєстрація каналів (умовна назва кімнати) за якими будь закріплені обрані пристрої.*
- *Можливість вибору користувачем необхідних пристроїв.*
- *На базі обраних пристроїв, генерація виконуючого коду для їх встановлення на Raspberry Pi, а також налаштування каналів MQTT брокеру на хмарі ThingSpeak.*

4.5 Веб-застосунок

Було прийнято рішення об'єднати бізнес-логіку та сервер сайту на одному сервері через невелику завантаженість застосунку. В якості сервера був використаний сервер Node.JS з підключеною до нього бібліотекою Express[13]. Так як мова фронт-енд частини веб-додатку була написана із застосуванням мови JavaScript, то вважається цілком обгрунтованим вибір серверу через швидкість написання та його ефективність роботи.

Було виділено окремий віртуальний сервер EC2 на ОС Ubuntu у хмарному сервісі AWS, на який було встановлено веб-сервер Node.JS, а також була розгорнута система управління базами даних PostgreSQL[14]. PostgreSQL – потужна СУБД, що цілком відповідає потреба веб-застосунку для збереження інформації про інтернет речі, а також користувацьких даних.

Кожному користувачу необхідно мати обліковий запис на хмарній IoT платформі ThingSpeak, щоб автоматично розгорнути інфраструктуру розумного дому. Для цього використовуючи ThingSpeak API, було написано скрипт для налаштування MQTT брокера під задані пристрої.

Для демонстрації проведеної роботи був зібраний прототип на базі Raspberry Pi 3 та декількох сенсорів: температури, вологості, освітленості, та витoku природного газу. Та підключений до хмарної платформи ThingSpeak. Нижче на рисунку зображена схема з'єднання всіх компонентів системи.

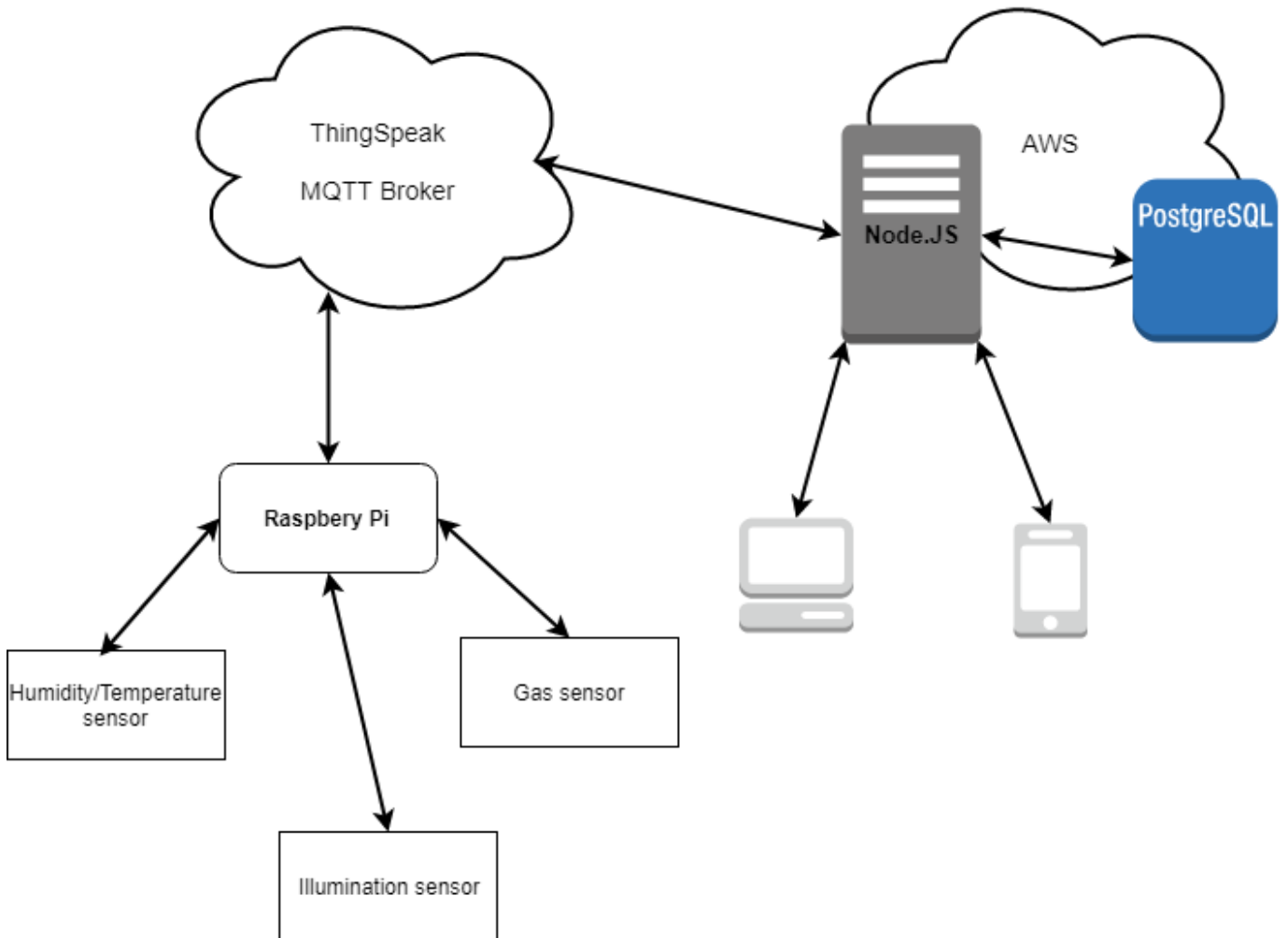


Рисунок 4.2 - Схема взаємодії брокера ThingSpeak, AWS сервіса і Raspberry Pi

Для моніторингу за підключеними сенсорами використовувався мобільний додаток ThingSpeak (IoT).

4.6. Висновки до розділу

На основі аналізу хмарних платформ і засобів для розгортання Інтернету речей був спроектований та програмно реалізований онлайн калькулятор для планування розумного будинку. В основі веб-додатку лежить платформа для мережеских застосунків Node.js, система управління базами даних PostgreSQL та ThingSpeak API. Створена систма передбачає наявність у користувача одноплатового комп'ютера Raspberry Pi, що виконує роль хабу, тобто пересилає дані підключених приладів в ThingSpeak хмару. Варто зазначити, що

користувачу не потрібно налаштовувати ThingSpeak сервіси для реєстрації хабу з всіма пристроями, що до нього підключені, це все виконує онлайн калькулятор.

5 РЕАЛІЗАЦІЯ СТАРТАП-ПРОЕКТУ «ОНЛАЙН КАЛЬКУЛЯТОР ДЛЯ ПЛАНУВАННЯ ІНФРАСТРУКТУРИ РОЗУМНОГО БУДИНКУ»

5.1 Опис ідеї проекту

У даному розділі описано економічне обґрунтування реалізації стартап-проекту на тему «Онлайн калькулятор для планування інфраструктури розумного будинку».

Таблиця 5.1 - Опис ідеї стартап-проекту

<i>Зміст ідеї</i>	<i>Напрямки застосування</i>	<i>Вигоди для користувача</i>
Ідея полягає в тому, щоб створити сервіс, який можна було б використовувати для планування розумного дому.	1. Можливість вибору конкретних датчиків та приладів із обширного набору.	При плануванні інфраструктури будинку для користувача доступна велика колекція приладів з яких він може обрати необхідні пристрої.
	2. Можливість автоматично налаштувати обрані пристрої.	Користувачу не потрібно налаштовувати хмарні сервіси і прилади для їх підключення.

Таблиця 5.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/ п	Техніко- економічні характерис- тики ідеї	(Потенційні) товари/концепції конкурентів				W (слабка сторон а)	N (нейтр альна сторон а)	S (сильна сторон а)
		Мій проект	Конкур ент1	Конкур ент2	Конкур ент3			
1.	Форма виконання	Сер- віс	До- даток	Веб- сервіс	Веб- сервіс			+
2.	Собівартість	Ни- зька	Висо- ка	Сере- дня	Сере- дня			+
3.	Наявність адміністрато ра для налаштуван- ня	Не треба	Не треба	Пот- рібно	Потріб но			+
4.	Наявність інтернету	Не треба	Треба	Треба	Не треба		+	
5.	Крос- платформені сть	Ні	Так	Ні	Ні	+		

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

5.2 Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару).

Таблиця 5.3 - Технологічна здійсненність ідеї проекту

<i>№ n/n</i>	<i>Ідея проекту</i>	<i>Технології її реалізації</i>	<i>Наявність технології</i>	<i>Доступність технології</i>
1.	Створення сервісу	JavaScript	Наявна	Безкоштовна, доступна
		Express.js	Наявна	Безкоштовна, доступна
Обрана технологія реалізації ідеї проекту: для створення сервісу обрана технологія JavaScript (Express.js), яка є безкоштовною та доступною.				

5.3 Аналіз ринкових можливостей

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку проводимо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку.

Таблиця 5.4 - Попередня характеристика потенційного ринку стартап-проекту

<i>№ n/n</i>	<i>Показники стану ринку (найменування)</i>	<i>Характеристика</i>
1.	Кількість головних гравців, од	3
2.	Загальний обсяг продаж, грн/ум.од	20000 грн./ум.од
3.	Динаміка ринку (якісна оцінка)	Зростає/спадає/стагнує
4.	Наявність обмежень для входу (вказати характер обмежень)	Немає
5.	Специфічні вимоги до стандартизації та сертифікації	Немає
6.	Середня норма рентабельності в галузі (або по ринку), %	$R = (3000000 * 100) / (1000000 * 12) = 25\%$

Таблиця 5.5 - Характеристика потенційних клієнтів стартап-проекту

<i>№ n/n</i>	<i>Потреба, що формує ринок</i>	<i>Цільова аудиторія (цільові сегменти ринку)</i>	<i>Відмінності у поведінці різних потенційних цільових груп клієнтів</i>	<i>Вимоги споживачів до товару</i>
1.	Сервіс для автоматичного проектування інфраструктури розумного дому	Будь-яка особа, що бажає спроектувати розумний будинок без залучення додаткових спеціалістів у галузі IoT	Відмінностей між групами не має, сервіс можуть використовувати за наявності необхідного обладнання	Рішення повинне бути зручним у користуванні, надійним, швидким

Таблиця 5.6 - Фактори загроз

<i>№ n/n</i>	<i>Фактор</i>	<i>Зміст загрози</i>	<i>Можлива реакція компанії</i>
1.	Конкуренція	Вихід на ринок великої компанії	<ul style="list-style-type: none"> ● вихід з ринку; ● запропонувати великій компанії поглинути себе; ● передбачити додаткові переваги власного сервісу для того, щоб повідомити про них саме після виходу міжнародної компанії на ринок
2.	Зміна потреб користувачів	Користувачам необхідний сервіс з іншим функціоналом	Передбачити можливість додавання нового функціоналу до створюваного додатку

Таблиця 5.7 - Фактори можливостей

<i>№ n/n</i>	<i>Фактор</i>	<i>Зміст можливості</i>	<i>Можлива реакція компанії</i>
1.	Зростання можливостей потенційних покупців	Зростання фінансових можливостей потенційних покупців	Запропонувати їм свої послуги
2.	Зниження довіри до конкурента 1	У додатку конкурента 1 нещодавно стався збій і протягом тижня система не правильно функціонувала	При виході на ринок звертати увагу покупців на надійність нашого сервісу

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку.

Таблиця 5.8 - Ступеневий аналіз конкуренції на ринку

<i>Особливості конкурентного середовища</i>	<i>В чому проявляється дана характеристика</i>	<i>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</i>
1. Вказати тип конкуренції: - досконала	Існує 3 фірми-конкуренти на ринку	Врахувати ціни конкурентних компаній на початкових етапах створення бізнесу, реклама (вказати на конкретні переваги перед конкурентами)
2. За рівнем конкурентної боротьби: - міжнародний	Одна з компаній – з іншої країни, дві – з України	Додати можливість вибору мови ПЗ, щоб легше було у майбутньому вийти на міжнародний ринок
3. За галузевою ознакою: - внутрішньогалузева	Конкуренти мають ПЗ, який використовується лише всередині даної галузі	Створити основу ПЗ таким чином, щоб можна було легко його переробити для

		використання у інших галузях
4. Конкуренція за видами товарів: - товарно-видова	Види товарів є однаковими, а саме - програмне забезпечення	Створити ПЗ, враховуючи недоліки конкурентів
5. За характером конкурентних переваг: - нецінова	Вдосконалення технології створення ПЗ, щоб собівартість була нижчою	Використання менш дорогих технологій для розробки, ніж використовують конкуренти
6. За інтенсивністю: - не марочна	Бренди відсутні	-

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

<i>Складові аналізу</i>	<i>Прямі конкуренти в галузі</i>	<i>Потенційні конкуренти</i>	<i>Постачальники</i>	<i>Клієнти</i>	<i>Товари-замінники</i>
	<i>Навести перелік прямих конкурентів</i>	<i>Визначити бар'єри входження в ринок</i>	<i>Визначити фактори сили постачальників</i>	<i>Визначити фактори сили споживачів</i>	<i>Фактори загроз з боку замінників</i>
Висновки	Існує 3 конкуренти на ринку. Найбільш схожим за виконанням є конкурент 2, так як його рішення також представлене у вигляді Веб сервісу.	Так, можливості для входу на ринок є, бо наше рішення спрощує та пришвидшує роботу спеціаліста.	Постачальники відсутні	Важливим для користувача є зручність у користуванні	Товари-замінники можуть використати більш дешеву технологію створення ПЗ та зменшити собівартість товару

На основі аналізу конкуренції із урахуванням характеристик ідеї проекту, вимог споживачів до товару та факторів маркетингового середовища визначається та обґрунтовується перелік факторів конкурентоспроможності (табл. 4.10).

Таблиця 5.10 - Обґрунтування факторів конкурентоспроможності

<i>№ n/n</i>	<i>Фактор конкурентоспроможності</i>	<i>Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)</i>
1.	Використання ПЗ у вигляді веб-сервісу	Дозволяє наочно побачити роботу ПЗ і правильність роботи.
2.	Простота інтерфейсу користувача	Користувач має лише завантажити фото.

За визначеними факторами конкурентоспроможності (табл. 4.10) проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 4.11). Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін.

Таблиця 5.11 - Порівняльний аналіз сильних та слабких сторін проекту

<i>№ n/n</i>	<i>Фактор конкурентоспроможності</i>	<i>Бали 1-20</i>	<i>Рейтинг товарів-конкурентів у порівнянні з нашим підприємством</i>						
			-3	-2	-1	0	+1	+2	+3
1.	Використання ПЗ у вигляді веб-сервісу	15			+				
2.	Простота інтерфейсу користувача	20	+						

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 5.12 – SWOT-аналіз стартап-проекту

Сильні сторони: простий користувацький інтерфейс, використання технологій	Слабкі сторони: потрібно мати обладнання тестування системи
Можливості: у конкурента 1 виявлена проблема із надійністю ПЗ, додаткове фінансування для розповсюдження даної технології	Загрози: конкуренція, зміна потреб користувачів

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів.

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

<i>№ п/п</i>	<i>Альтернатива (орієнтовний комплекс заходів) ринкової поведінки</i>	<i>Ймовірність отримання ресурсів</i>	<i>Строки реалізації</i>
1.	Створення ПЗ використовуючи ThingSpeak	80%	6 місяців
2.	Створення ПЗ на основі AWS IoT	30%	12 місяців

З означених альтернатив обирається та, для якої: а) отримання ресурсів є більш простим та ймовірним; б) строки реалізації – більш стислими. Тому обираємо альтернативу 1.

5.4 Розробка ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Таблиця 5.14 - Вибір цільових груп потенційних споживачів

<i>№ п/п</i>	<i>Опис профілю цільової групи потенційних клієнтів</i>	<i>Готовність споживачів сприйняти продукт</i>	<i>Орієнтовний попит в межах цільової групи (сегменту)</i>	<i>Інтенсивність конкуренції в сегменті</i>	<i>Простота входу у сегмент</i>
1.	Пересічний користувач	Можливість автоматичної налаштування системи з обраними датчиками і приладами	Великий	Існує 3 конкуренти, які надають схожі, але менш швидкі рішення.	Швидкодія, зручний користувачський інтерфейс
Які цільові групи обрано: Пересічний користувач					

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку.

Для роботи в обраних сегментах ринку необхідно сформувавши базову стратегію розвитку. За М. Портером, існують три базові стратегії розвитку, що відрізняються за ступенем охоплення цільового ринку та типом конкурентної переваги, що має бути реалізована на ринку (за витратами або визначними якостями товару).

Стратегія лідерства по витратах передбачає, що компанія за рахунок чинників внутрішнього і/або зовнішнього середовища може забезпечити більшу, ніж у

конкурентів маржу між собівартістю товару і середньоринковою ціною (або ж ціною головного конкурента).

Стратегія диференціації передбачає надання товару важливих з точки зору споживача відмітних властивостей, які роблять товар відмінним від товарів конкурентів. Така відмінність може базуватися на об'єктивних або суб'єктивних, відчутних і невідчутних властивостях товару(у ширшому розумінні – комплексі маркетингу), бути реальною або уявною. Інструментом реалізації стратегії диференціації є ринкове позиціонування.

Стратегія спеціалізації передбачає концентрацію на потребах одного цільового сегменту, без прагнення охопити увесь ринок. Мета тут полягає в задоволенні потреб вибраного цільового сегменту краще, ніж конкуренти. Така стратегія може спиратися як на диференціацію, так і на лідерство по витратах, або і на те, і на інше, але тільки у рамках цільового сегменту. Проте низька ринкова доля у разі невдалої реалізації стратегії може істотно підірвати конкурентоспроможність компанії.

Таблиця 5.15 – Визначення базової стратегії розвитку

<i>№ n/n</i>	<i>Обрана альтернатива розвитку проекту</i>	<i>Стратегія охоплення ринку</i>	<i>Ключові конкурентоспроможні позиції відповідно до обраної альтернативи</i>	<i>Базова стратегія розвитку</i>
1.	Створення ПЗ використовуючи хмарну платформу Інтернету речей ThingSpeak	Ринкове позиціонування	Швидкодія, простота у користуванні	Диференціація

Наступним кроком є вибір стратегії конкурентної поведінки.

Стратегія лідера. Залежно від міри сформованості товарного(галузевого) ринку, характеру конкурентної боротьби компанії-лідери обирають одну з трьох

стратегій: розширення первинного попиту, оборонну або наступальну стратегію або ж застосувати демаркетинг або диверсифікацію.

Стратегія розширення первинного попиту доцільна у разі, якщо фірмі-лідеріві недоцільно розмінюватися на боротьбу з невеликими конкурентами, вона може отримати велику економічну віддачу від розширення первинного рівня попиту. В цьому випадку компанія займається реалізацією заходів по формуванню попиту(навчанню споживачів користуванню товаром, формування регулярного попиту, збільшення разового споживання), також пропаганду нових напрямів застосувань існуючих товарів, виявлень нових груп споживачів.

У міру зростання ринку, його становлення позиції компанії-новатора починають атакувати конкуренти-імітатори. В цьому випадку, компанія може вибрати оборонну стратегію, метою якої є захист власної ринкової долі.

Наступальна стратегія припускає збільшення своєї частки ринку. При цьому переслідувана мета полягає в подальшому підвищенні прибутковості роботи компанії на ринку за рахунок максимального використання ефекту масштабу.

Якщо фірма потрапляє під дію антимонопольного законодавства, вона може удатися до стратегії демаркетинга, що припускає скорочення своєї частки ринку, зниження рівня попиту на деяких сегментах за рахунок підвищення ціни. При цьому ставиться завдання недопущення на ці сегменти конкурентів, а компенсація втрат прибутку через зменшення обсягів виробництва компенсується встановленням надвисоких цін.

Стратегія виклику лідера. Стратегію виклику лідеріві найчастіше вибирають компанії, які є другими, третіми на ринку, але бажають стати лідером ринку. Теоретично, ці компанії можуть прийняти два стратегічні рішення: атакувати лідера у боротьбі за частку ринку або ж йти за лідером.

Рішення атакувати лідера є досить ризикованим. Для його реалізації потрібні значні фінансові витрати, know - how, краще співвідношення «ціна- якість», переваги в системі розподілу і просування і т. д. У разі не реалізації цієї стратегії, компанія може бути відкинута на аутсайдерські позиції на досить довгий час.

Залежно від цього компанія може вибрати одну з альтернативних стратегій: фронтальної або флангової атаки.

Стратегія наслідування лідеру. Компанії, що приймають слідування за лідером – це підприємства з невеликою часткою ринку, які вибирають адаптивну лінію поведінки на ринку, усвідомлюють своє місце на ній і йдуть у фарватері фірм-лідерів. Головна перевага такої стратегії - економія фінансових ресурсів, пов'язаних з необхідністю розширення товарного(галузевого) ринку, постійними інноваціями, витратами на утримання домінуючого положення.

Стратегія заняття конкурентної ніші. При прийнятті стратегії зайняття конкурентної ніші (інші назви - стратегія фахівця або нішера) компанія в якості цільового ринку вибирає один або декілька ринкових сегментів. Головна особливість - малий розмір сегментів/сегменту. Ця конкурентна стратегія являється похідною від такої базової стратегії компанії, як концентрація. Головне завдання для компаній, що вибирають стратегію нішера або фахівця, - це постійна турбота про підтримку і розвиток своєї конкурентної переваги, формування лояльності і прихильності споживачів, підтримка вхідних бар'єрів.

Таблиця 5.16 - Визначення базової стратегії конкурентної поведінки

<i>№ п/п</i>	<i>Чи є проект «першопрохідцем» на ринку?</i>	<i>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?</i>	<i>Чи буде компанія копіювати основні характеристики товару конкурента, і які?</i>	<i>Стратегія конкурентної поведінки</i>
1.	Ні	Так	Буде, а саме: основною задачею є розробка ПЗ з використанням нейронних мереж (конкуренти 1, 2, 3), простий інтерфейс	Зайняття конкурентної ніші

			користувача (конкурент 2)	
--	--	--	------------------------------	--

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробляється стратегія позиціонування, що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 5.17 - Визначення стратегії позиціонування

<i>№ n/n</i>	<i>Вимоги до товару цільової аудиторії</i>	<i>Базова стратегія розвитку</i>	<i>Ключові конкурентоспроможні позиції власного стартап-проекту</i>	<i>Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту (три ключових)</i>
1.	Простота інтерфейсу, швидкодія	Диференціація	Простота користувацького інтерфейсу дозволить отримувати необхідні дані і відслідковувати події в режимі реального часу	Швидкодія, безпека, простота

5.5 Розробка маркетингової програми

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл. 4.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.18 - Визначення ключових переваг концепції потенційного товару

<i>№ n/n</i>	<i>Потреба</i>	<i>Вигода, яку пропонує товар</i>	<i>Ключові переваги перед конкурентами (існуючі</i>
--------------	----------------	-----------------------------------	-----------------------------------------------------

			<i>або такі, що потрібно створити)</i>
1.	Швидкодія	ПЗ працює досить швидко, результат можна отримати до 10 мс	Перевага у швидкості
2.	Простота користувацького інтерфейсу	Простота роботи додатку	Користувачі мають зручний інтерфейс для взаємодії з ПЗ

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.19).

1-й рівень. При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень. Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень. Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін).

Таблиця 5.19 - Опис трьох рівнів моделі товару

<i>Рівні товару</i>	<i>Сутність та складові</i>		
I. Товар за задумом	ПЗ дає можливість автоматизувати планування розумного будинку		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Зручність та простота користувацького інтерфейсу 2. Швидкість роботи	1.Нм 2.Нм 3.Нм	1.Нм 2.Нм 3.Нм 4.Нм

	3. Безпека згідно до світових стандартів		
	Якість: згідно до стандарту ISO 4444 буде проведено тестування		
	Маркування відсутнє		
	Моя компанія: “Automotized Smart House”		
III. Товар із підкріпленням	1-місячна пробна безкоштовна версія		
	Постійна підтримка для користувачів		
За рахунок чого потенційний товар буде захищено від копіювання: патент.			

Після формування маркетингової моделі товару слід особливо відмітити – чим саме проект буде захищено від копіювання. Захист може бути організовано за рахунок захисту ідеї товару (захист інтелектуальної власності), або ноу-хау, чи комплексне поєднання властивостей і характеристик, закладене на другому та третьому рівнях товару.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (табл. 4.20). Аналіз проводиться експертним методом.

Таблиця 5.20 - Визначення меж встановлення ціни

<i>№ п/п</i>	<i>Рівень цін на товари-замінники</i>	<i>Рівень цін на товари-аналоги</i>	<i>Рівень доходів цільової групи споживачів</i>	<i>Верхня та нижня межі встановлення ціни на товар/послугу</i>
1.	25000 грн	30000 грн	200000 грн	20000 грн

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.21).

Таблиця 5.21 - Формування системи збуту

<i>№ п/п</i>	<i>Специфіка закупівельної поведінки цільових клієнтів</i>	<i>Функції збуту, які має виконувати постачальник товару</i>	<i>Глибина каналу збуту</i>	<i>Оптимальна система збуту</i>
1.	Купують підписку та роблять щорічні внески для подовження ліцензії	Продаж	0(напрямую), 1(через одного посередника)	Власна та через посередників

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 5.22 - Концепція маркетингових комунікацій

<i>№ п/п</i>	<i>Специфіка поведінки цільових клієнтів</i>	<i>Канали комунікацій, якими користуються цільові клієнти</i>	<i>Ключові позиції, обрані для позиціонування</i>	<i>Завдання рекламного повідомлення</i>	<i>Концепція рекламного звернення</i>
1.	Використання за допомогою сайту	Інтернет	Швидкодія, простота у використанні, безпека	Показати переваги сервісу, у тому числі і перед конкурентами	Демо-ролик із використання

5.6 Висновки до розділу 5

Згідно до проведених досліджень існує можливість ринкової комерціалізації проекту. Також, варто відмітити, що існують перспективи впровадження з огляду на потенційні групи клієнтів, бар'єри входження не є високими, а проект має дві значні переваги перед конкурентами.

Для успішного виконання проекту необхідно реалізувати сервіс із автоматизації планування розумного будинку. В рамках даного дослідження були розраховані основні фінансово-економічні показники проекту, а також проведений менеджмент потенційних ризиків. Проаналізувавши отримані результати, можна зробити висновок, що подальша імплементація є доцільною.

ВИСНОВОК

В роботі розглядається одна з найактуальніших тенденцій розвитку інформаційних технологій Internet of Things (IoT). Проводиться огляд і аналіз рішень, розроблених відповідно до концепції Інтернету речей. Дана технологія набуває все більшої популярності, оскільки дозволяє об'єднати в єдину мережу прилади для моніторингу та керування елементами оточуючого простору, що в свою чергу надає безліч можливостей для автоматизації як виробничих процесів так і буденних справ кожної людини. Тому дослідження даної технології, а також створення програмного засобу для планування розумного будинку, є актуальним напрямком досліджень саме на сьогоднішній день.

Метою даної роботи є дослідження існуючих засобів для розгортання IoT, а також аналіз комерційних і некомерційних хмарних платформ Інтернету речей.

Було проведено оглядовий аналіз існуючих рішень для створення інфраструктури розумного дому. Розглянуті компанії займають лідерські позиції на ринку Інтернету речей і мають ряд переваг. Кожна з них надає власний веб-сервіс для взаємодії користувача з Інтернет речами і їх обслуговування, що є значною перевагою, але це виключає можливість підключення Інтернет речей від інших виробників. Для рішення цієї проблеми можна використати хмарну IoT платформу.

Було проведено аналіз хмарних платформ, знайдені відмінності між розглянутими сервісами та виділені сильні та слабкі сторони кожного з них.

В даному розділі були розглянуті хмарні платформи керування розумним будинком, такі як: Microsoft Azure IoT Suite, AWS IoT, ThingSpeak, Google Cloud IoT, IBM Watson IoT.

В аспекті безпеки передачі даних розглянуті платформи являються однаково надійними. В них реалізовані передові технології аутентифікації користувачів і

пристроїв, а також шифруванні даних. В аспекті візуалізації, моніторингу та аналізу даних всі п'ять платформ надають подібний набір засобів для адміністрування та налаштування датчиків.

Також всі висвітлені платформи надають широкий вибір пристроїв, що можуть бути підключеними. Тому ключовим фактором для вибору конкретної платформи є достатньо вільний доступ до ресурсів платформи, де лідирує ThingSpeak.

Було створено онлайн калькулятор для проектування розумного будинку, що дозволяє користувачеві обрати необхідний набір пристроїв та розгорнути їх в декілька нескладних кроків. В основі веб-додатку лежить платформа для мережеских застосунків Node.js, система управління базами даних PostgreSQL та ThingSpeak API. Створена система передбачає наявність у користувача одноплатового комп'ютера Raspberry Pi, що виконує роль хабу, тобто пересилає дані підключених приладів в ThingSpeak хмару. Варто зазначити, що користувачу не потрібно налаштовувати ThingSpeak сервіси для реєстрації хабу з всіма пристроями, що до нього підключені, це все виконує онлайн калькулятор.

Реалізована система відмінно зарекомендувала себе на практиці оскільки надає зручний і зрозумілий інтерфейс, а також є масштабованою. Крім того система працює без перебоїв за умов низької пропускну здатності завдяки використанню протоколу передачі даних Message Queue Telemetry Transport.

Даний програмний засіб можна також визначити, як стартовий комплект для початківців в сфері IoT, адже дозволяє користувачеві власноруч налаштувати зв'язки між пристроями мережі, крім того не обов'язково використовувати дорогі прилади таких відомих компаній як Nest, SmartThings, Ecobee, а й звичані сенсори та прилади, що можна підключити через GPIO інтерфейс Raspberry Pi. Також в ході виконання дипломної роботи була побудований робочий прототип розумного будинку з датчиками температури,

вологості та освітленості.

ПЕРЕЛІК ПОСИЛАНЬ

1. Безопасность в IoT [Електронний ресурс]. – Режим доступа: <https://habrahabr.ru/company/microsoft/blog/313762/>. Дата доступу: 05.05.2018р
2. Інтеграція і взаємодія в мережі Веб [Електронний ресурс]. — Режим доступа: <http://www.intuit.ru/studies/courses/485/341/lecture/8211>. — Дата доступа: 05.05.2018р
3. Офіційний сайт компанії Amazon [Електронний ресурс]. — Режим доступа: <https://aws.amazon.com/ru/iot/how-it-works/> — Дата доступа: 05.05.2018р
4. Офіційний сайт компанії Google [Електронний ресурс]. — Режим доступа: <https://cloud.google.com/solutions/iot/> — Дата доступа: 05.05.2018р
5. Офіційний сайт компанії Samsung [Електронний ресурс]. — Режим доступа: <http://developer.samsung.com/smart-home> — Дата доступа: 05.05.2018р
6. Офіційний сайт компанії Amazon [Електронний ресурс]. — Режим доступа: https://aws.amazon.com/ru/pricing/?nc2=h_ql_ny_livestream_blu — Дата доступа: 05.05.2018р
7. Офіційний сайт компанії ThingSpeak [Електронний ресурс].— Режим доступа: https://thingspeak.com/pages/how_to — Дата доступа: 05.05.2018р
8. Сайт MQTT [Електронний ресурс]. — Режим доступа: <http://mqtt.org/> — Дата доступа: 05.05.2018р
9. Офіційна сторінка Raspberry Pi [Електронний ресурс]. — Режим доступа: <https://www.raspberrypi.org/documentation/> — Дата доступа: 05.05.2018р
10. Офіційна сторінка Raspberry Pi на порталі GitHub [Електронний ресурс]. — Режим доступа: <https://github.com/raspberrypi> — Дата доступа: 05.05.2018р
11. Офіційна сторінка NodeJS на російській мові [Електронний ресурс]. — Режим доступа: <http://nodejs.ru/> — Дата доступа: 05.05.2018р

- 12.Офіційний сайт Python [Електронний ресурс] — Режим доступу:
<https://www.python.org> — Дата доступу: 05.05.2018р
- 13.Офіційна сторінка Express для NodeJs [Електронний ресурс]. — Режим
доступу:
<http://expressjs.com/uk/> — Дата доступу: 05.05.2018р
- 14.Офіційний сайт PostgreSQL [Електронний ресурс] — Режим доступу:
<https://www.postgresql.org/> — Дата доступу: 05.05.2018р