

**Факультет інформаційних технологій**  
**Кафедра мережевих та інтернет технологій**

**ЗАТВЕРДЖУЮ**  
завідувач кафедри  
мережевих та інтернет технологій  
\_\_\_\_\_ Ю.В. Кравченко  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ року

**КВАЛІФІКАЦІЙНА РОБОТА**  
**БАКАЛАВРА**

галузі знань 17 «Електроніка та телекомунікації»  
за спеціальністю 172 «Телекомунікації та радіотехніка»

на тему:

**ВИКОРИСТАННЯ СИСТЕМИ ДИСТАНЦІЙНОГО**  
**КОНТРОЛЮ ФУНКЦІОНУВАННЯ ПРИСТРОЇВ ІОТ**

**Виконав: студент групи МІТ-41**

\_\_\_\_\_ Горноста́й Максим  
(прізвище ім'я по-батькові)

\_\_\_\_\_ (підпис)

**Керівник: доцент кафедри мережевих та інтернет технологій**

\_\_\_\_\_ (прізвище ім'я по-батькові)

\_\_\_\_\_ (підпис)

**Київ 2021**

Міністерство освіти і науки України  
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій  
Кафедра мережевих та інтернет технологій

**ЗАТВЕРДЖУЮ**

завідувач кафедри

мережевих та інтернет технологій

Ю.В. Кравченко

« \_\_\_\_\_ » \_\_\_\_\_ 2021 року

**ЗАВДАННЯ  
НА ДИПЛОМНУ РОБОТУ**

Здобувачу вищої освіти

Горностай Максим Володимирович

(прізвище, ім'я, по батькові)

1. Тема роботи:

«Використання системи дистанційного контролю функціонування пристроїв IoT»

затверджена на засіданні кафедри МІТ «11» грудня 2020 р. протокол № 6

2. Термін здачі закінченої роботи

«30» травня 2021р

3. Вихідні дані до проекту (роботи)

Бібліотека для взаємодії з АПІ застосунку «Telegram»

Програмне забезпечення та скрипти для контролю і керування

4. Зміст пояснювальної записки (перелік питань, що їх потрібно розробити, обсяг – 35-50 стор.)

1. Дослідження підходів та інструментів для реалізації мережі IoT.

2. Проектування системи керування для мереж IoT.

5. Перелік графічного матеріалу 8-12 слайдів

Огляд існуючих систем з IoT.

Огляд засобів що використовуються при проектуванні мереж IoT

Проектування компонентів.

Реалізація проекту.

Демонстрація роботи програмного забезпечення.

Дата видачі завдання

Керівник роботи

д.т.н., доцент кафедри МІТ Труш О.В.

(підпис)

(посада, прізвище, ім'я, по батькові)

Завдання прийняв до виконання

(підпис)

(прізвище, ім'я, по батькові)

## КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ РОБОТИ

Номер	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Підготовчий	29.01.2021	
2	Розділ 1	01.03.2021	
3	Розділ 2	01.04.2021	
4	Розділ 3	01.05.2021	
5	Доповідь та слайди	27.05.2021	
6	Пояснювальна записка	30.05.2021	

Здобувач вищої освіти \_\_\_\_\_  
(підпис) (прізвище, ім'я, по батькові)

Керівник \_\_\_\_\_  
(підпис) (прізвище, ім'я, по батькові)

## РЕФЕРАТ

Дипломний проект, 52 с., 18 рис., 2 додатки, 14 джерел.

Об'єктом дослідження була система дистанційного контролю функціонування мереж з використанням IoT для автоматизації діяльності підприємств.

Метою дослідження був якісний аналіз існуючих систем дистанційного контролю за виконанням функцій в мережах IoT, аналіз основних технологій в мережах IoT, опис і порівняння основних протоколів.

У ході дипломного проекту отримані:

- Опис технологій в сфері IoT що впроваджуються та вже працюють;
- Опис основних технологій побудови мережі з використанням IoT;
- Порівняння основних протоколів IoT;
- Програмний застосунок для дистанційного контролю за функціонуванням мережі IoT;
- Рекомендації щодо впровадження програмного продукту на підприємстві та щодо побудови мереж IoT.

Основним методом проектування були мова Python, програмний застосунок Telegram.

Виконання роботи проводилося відповідно до навчального плану за освітньою програмою «Мережеві та інтернет технології».

Стан роботи: Програмний застосунок в робочому стані, на стадії альфа тестування.

Прогноз розвитку даної розробки – як вже зазначено в даній дипломній роботі, сфера «розумних» IoT мереж досить популярна на сьогодні, тому представлене дослідження та програмний застосунок несуть неопосередкований внесок в розвиток і систематизацію знань в сфері інфокомунікаційних технологій.

АВТОМАТИЗАЦІЯ ПІДПРИЄМСТВА, РОЗУМНИЙ ДІМ, СИСТЕМИ ДИСТАНЦІЙНОГО КОНТРОЛЮ МЕРЕЖ ІОТ, ПРОТОКОЛИ ІНТЕРНЕТУ РЕЧЕЙ, ДАТЧИКИ, МІКРОКОНТРОЛЕРИ, ТЕЛЕГРАМ БОТ.

## ЗМІСТ

ВСТУП.....	7
1 КОНЦЕПЦІЯ СВІТУ ІНТЕРНЕТ РЕЧЕЙ.....	10
1.1. Історія виникнення концепції інтернету речей.....	10
1.2. Приклади використання IoT систем у сучасному світі ...	13
2 АНАЛІЗ ТЕХНОЛОГІЙ І СТАНДАРТІВ ДЛЯ ПОБУДОВИ МЕРЕЖІ ІОТ .....	24
2.1. Канали передачі даних.....	24
2.2. Протоколи і стандарти для IoT пристроїв .....	29
2.3. Constrained Application Protocol .....	33
2.4. Data Distribution Service .....	34
2.5. Extensible Messaging and Presence Protocol.....	35
2.6. Message Queue Telemetry Transport.....	35
2.7. Advanced Message Queuing Protocol.....	36
2.8. Simple Text Oriented Message Protocol .....	37
2.9. Пристрої інтернету речей .....	37
2.10. Системи управління мережами інтернет речей .....	39
3 СИСТЕМА ДИСТАНЦІЙНОГО КОНТРОЛЮ ІОТ .....	41
3.1. Програма дистанційного управління.....	41
3.2. Склад програми дистанційного управління .....	41
3.3. Захист програми дистанційного управління .....	44
3.4. Використання програми дистанційного управління .....	45
ВИСНОВОК .....	46
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48
ДОДАТКИ .....	49

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

ДП – дипломний проект

БД – база даних

ІоТ – Інтернет речей (Internet of Things)

ШІ – штучний інтелект

## ВСТУП

*Актуальність теми.* Ось уже понад сто років людство переживає бурхливу промислову революцію. Останнім часом світ кардинально змінюється все частіше. Сто років тому були вигадані перші промислові машини та летючі апарати, а сьогодні людина стрімко рухається до зірок космічними апаратами, а у кожного в кишені розумний апарат, що має блискавичний доступ до будь-якої точки зв'язку у світі, а також до електронної бібліотеки лівової частки знань людства за 40 000 років еволюції.

Я виділив саме останнє століття розвитку суспільства через те що саме в цей час людина стала відходити від основної ролі створення продукту, і віддала це місце механічним, а в наш час ще і логічним, пристроям що значно спрощують життєдіяльність, і повсякденне життя людини розумної. Це стосується процесів здобування нових знань та дослідів, а також вироблення основних харчових продуктів, побуту, тощо...

Саме попереднє століття показало нам що людський інтелект може вигадати пристрій який може виконувати ту чи іншу роботу, а його друга половина дала нам розгалужені системи комунікації одне з одним, та логічні пристрої, що набагато спрощують інтелектуальну роботу, і можуть виконувати задачі не маючи жодного рухомого компонента всередині.

Перехід людства до цифрової епохи знаменувався все більшим розрахунком на автоматизовані системи, які будучі добре налагодженими виконують роботу замість людини швидше і точніше. Людині залишається лише слідкувати за роботою чи то механізмів, чи то алгоритмів, обслуговуючи їх, налагоджуючи їх, та виправляючи помилки в них, вдосконалюючи їх до ідеалу.

Саме в такому світі ми зараз можемо бути свідком достатку товарів необхідних людині, високотехнологічних розробок що допомагають людині продовжувати час свого існування в цьому світі, та купити освітніх та розважальних програм що її навчають та веселять. І це все завдяки тому що налагоджені автоматичні системи знімають з людини ярмо важкої роботи, звільняючи її

дорогоцінний час існування для роботи більш значущої та вічної. Саме через це людина так відчайдушно створює все новітніші, все досконаліші системи.

Це стало причиною виникнення нових концепцій управління, будь то моніторинг критично значущих параметрів (як от клімат чи показники приладів), чи то дистанційне виконання операцій управління, перемикання чи вимкнення систем. Все це підштовхує нас до основної теми моєї дипломної роботи, використання систем моніторингу функціонування, і покладеної в основу концепції інтернету речей, як однієї з найсучасніших доробок в області автоматизації та управління виконання роботи.

*Ступінь наукової роботи:* У світі існує безліч різноманітних систем IoT, в тому числі й систем моніторингу, й управління ними. Але більшість з них є пропрієтарними, засекреченими або просто недоступними широкому загалу через права інтелектуальної власності на такі доробки. Це підводить нас до думки що такі системи є надзвичайно цінними та корисними, а їх розробка є досить місткою і дорогою процедурою. Також слід зазначити що є і готові системи що продаються в якості продукту, що спрощує повсякденне життя людини. Так, такі системи є доступними, але їх функціонал обмежений технічними завданнями що були перед ними поставлені на початку розробки, а використання їх як основи для нової, більш універсальної системи не може бути й мови, бо така система продається вже в готовому вигляді, для чітко визначених задач, і її особливості (як от вихідний код) є корпоративною таємницею що приносить прибутки, що і спонукає нас до розробки власних доробок в даній області.

*Практичне значення одержаних результатів:* користь результатів одержаних в ході виконання дипломної роботи складно оцінити. В ДП наведено основні відомі технологічні доробки в області IoT, основні стандарти та протоколи що існують для створення систем управління розумними датчиками та пристроями, що під'єднані до мережі інтернет. Також результатом виконання роботи є готовий програмний продукт, що є доволі дешевою, але водночас і доволі доступною і функціональною альтернативою системи моніторингу систем інтернет речей. Даний програмний доробок може бути використаний як окрема система виконання

простих операцій з пристроями під'єднаними буцім то до інтернету чи обчислювальних машин (граничного маршрутизатора), так і як частина складнішої системи контролю виконання технологічного процесу чи адміністрування корпоративних чи особистих систем на основі IoT, як основа системи виводу чи вводу інформації, так і як резервна чи допоміжна система контролю/адміністрування, а також як дистанційний пульт керування системи контролю чи виконання того чи іншого процесу чи системи на основі IoT.

*Апробація результатів:* Як дослідження теми дипломної роботи були вивчені вже перевірені на практиці стандарти чи готові системи на основі IoT, що мають попит серед людей, чи створені для особистих чи корпоративних потреб. Також деякі теми що входять в дану дипломну роботу освітлювалися на науково-технічних конференціях. Приклад програмного доробку також може виконувати поставлені перед ним задачі, і може бути використаний як програмний продукт корпоративного класу.

# 1 КОНЦЕПЦІЯ СВІТУ ІНТЕРНЕТ РЕЧЕЙ

## 1.1. Історія виникнення концепції інтернету речей

Про передумови виникнення взагалі ідеї інтернету речей в наш час ми можемо тільки здогадуватись, але для того, щоб розібратись в цьому питанні необхідно зазирнути у той час, коли те що ми зараз називаємо інтернет тільки зароджувалося.

Отже, в другій половині XIX століття з'являються патенти на автоматичне управління телефонними мережами, модель яких передувала всеосяжному інтернету. Така ідея була досить кардинальною для того часу, коли всі телефонні мережі обслуговувалися спеціально навченими телефоністками, які саме і з'єднували абонентів між собою. Перехід на автоматику поступово знищив велику кількість робочих місць, а також економив як і гроші телефонних компаній, так і час абонентів що користувалися телефонним зв'язком.

В другій половині 1960-х, новоспечена компанія найновітніших розробок Америки (DARPA) починає впровадження своєї розробки що має назву ARPA-NET. Саме її можна найменувати попередницею того самого інтернету що ми можемо спостерігати сьогодні. ARPA-NET була створена як військовий проєкт, сутність її полягала в тому щоб з'єднувати між собою перші комп'ютери, для передачі інформації, для пришвидшення новітніх розробок наукових інститутів. Саме так у зародку виглядала міжмашинна взаємодія (M2M). За основу такої розробки були взяті ідеї Пола Берна про розподілену систему комунікації, а також пакетної маршрутизації. Інженери та науковці державного агентства розробили комплекс основних протоколів і концепти передачі інформації що стали основою сучасної мережі на моделі TCP/IP.

Отже, першими ідеями створення всесвітньої мережі для обміну інформацією були з'єднання між собою рахункових машин, для комунікації і їх спільної роботи, але сама концепція між машинної взаємодії була запропонована дещо пізніше. Ідею створення самого вислову приписують Теодору Параскевакосу,

греко-американському винахіднику. Він володіє достатньо великою кількістю патентів, і в основному його винаходи пов'язані зі сферою комунікації та передачі інформації. Саме йому належить винахід пристрою для ідентифікації абонента що хоче зв'язатися аналоговою чи цифровою мережею, служба що з часом перетворилася в добре нам відому службу VoIP, а також цифрові системи зв'язку сигналізації, що зчитують та передають кардіологічні сигнали, покази лічильників, та банківські системи інтелектуальної перевірки валюти.

Міжмашинна передача даних почала активно просуватися на початку XXI століття, основними компаніями з комунікації та організації передачі даних, таких як Nokia, AT&T, Verizon, IBM, і вже сучасних GOOGLE і Amazon. Основними принципами були швидша і надійніша передача інформації між великою кількістю пристроїв без необхідності людини брати участь в них. Такий підхід вирішує велику кількість питань і проблем що безумовно з'являються з плином часу і розвитком суспільства. Людина з кожним днем використовує все більше ресурсів, а якісна автоматична організація дозволяє їй не брати безпосередньої участі у великій кількості процесів, що пришвидшує прогрес людства в цілому.

Основою концепту M2M є двоточкове з'єднання з машиною, що дозволяє їй повідомляти про показники датчиків чи здійснювати віддалене керування нею. До пристроїв M2M відносяться торгові автомати (першим пристроєм мережі M2M був торговельний автомат компанії Coca-Cola в університеті Карнегі-Мелон, їм керували місцеві програмісти, що за допомогою датчиків змусили автомат повідомляти про температуру яка підтримується в автоматі та кількість товару що в ньому знаходиться), банкомати, камери відеоспостереження, датчики та сенсори сигналізації, та інші.

Розвиток інфокомунікаційних систем призвів до все більшого розповсюдження між машинної передачі даних, і от вже в 1999 році Кевіном Ештоном була представлена концепція «інтернету речей» (IoT).

Слід зазначити, що терміни M2M та IoT досить подібні, і багато хто помилково вважає що це одне і теж саме, але вони між собою дуже різняться. Концепт M2M передбачає двоточкове з'єднання між пристроями, тобто якщо це

банкомат, то він має виділену лінію до банку, куди присилає інформацію про недостатність чи надлишок готівки, дані з датчиків, інформацію про транзакції проведені через апарат. Концепт IoT наслідує основні принципи між машинної взаємодії, але значно їх розширюючи. В інтернеті речей необов'язково протягувати окремі лінії до кожного пристрою, достатньо під'єднати датчик до мережі інтернет, і зв'язок оброблювальної машини з датчиками буде налагоджено. Також необов'язково щоб на дальньому кінці була складна машина, яка поєднує в собі деяку кількість датчиків, а також апаратних і функціональних частин. Зв'язок налагоджується з окремими функціональними частинами, що в перспективі створює величезну розподілену систему, що має універсальний характер, і керування якою може проводитись віддалено і централізовано, з окремого пульта керування, чи через вебінтерфейс, що дозволяє створювати мережі «речей» більш гнучко і просто, з невеликими грошовими витратами на купівлю матеріальної бази для інфраструктури мережі. Отже, M2M може існувати без інтернету речей, а от IoT створюється на базі ліній передачі даних і між машинних мереж.

Все більший розголос тем пов'язаних з IoT досить звичайна справа. Все більший розвиток розумної електроніки та систем передачі інформації спонукав до збільшення кількості під'єднаних до всесвітньої мережі інтернет пристроїв, збільшення пропускної спроможності мережевого обладнання, та розвитку все новіших та досконаліших винаходів. Так вже у 2010-му до мережі було під'єднано майже вдвічі більше пристроїв ніж було людей на планеті, а у 2020 ця різниця виросла до приблизно 7 пристроїв на кожну людину. Така тенденція продукує нові виклики для організації та керування такою великою кількістю пристроїв та просто величезним обсягом інформації яка створюється цим самим приладдям. Створюються все нові області дослідження, такі як Big Data.

Станом на сьогодні навіть концепція IoT пережила модернізацію. Їй на заміну прийшла концепція «Internet of Everything» (IoE), фундаментом для якої слугував інтернет речей. Дана концепція об'єднує взагалі будь-які пристрої що генерують інформацію чи виконують ту чи іншу роботу, в одну глобальну інфокомунікаційну систему. Таким чином людство може легко і ефективно створювати системи

керування, на перший погляд, досить тривіальними пристроями, такими як світлофори, крани запирання/відпирання систем водопостачання, автоматизованими кавомашинами, для забезпечення їх сумісної і якісної роботи в досить об'ємних макросистемах, таких як підприємство, чи навіть ціле місто. Такі системи спрощують організацію спільного простору людини, допомагають більш економно та ефективно розпоряджатися наявними ресурсами, або просто спрощувати вирішення більш окремих і особистих проблем, на прикладі автоматизованої сигналізаційної системи оселі, або більш інтелектуальної системи «розумного будинку», що якісно вирішує низку побутових і рутинних операцій. Автоматизовані системи ІоЕ, чи ІоТ, вже сьогодні покращують умови повсякденного життя людини, зміцнюючи системи протипожежної, антивандальної чи правоохоронної безпеки, налагоджуючи організацію руху автотранспорту, чи контролюючи системи опалення оселі.

## **1.2. Приклади використання ІоТ систем у сучасному світі**

Системи на основі під'єднаних до інтернету речей, будь то датчики, сенсори або що, оточують нас всюди, і ми про це навіть не здогадуємось. На планеті вже з'являються цілі розумні міста. Впровадження інтелектуальних систем можна побачити на прикладі не тільки великих міст Європи чи Америки, таких як Барселона, Мадрид, Лондон чи Нью-Йорк, а і на прикладі міст Казахстану (Астана), Китаю (Пекін), Азербайджану (Баку), Киргизії (Бішкек), Узбекистану (Ташкент), Грузії (Батумі), Канади (Торонто), Австралії (Сідней), Бразилії (Ріо-де-Жанейро), Катару (Доха), Ізраїлю (Тель-Авів), а також міст в ОАЄ, Нідерландах, Сінгапурі, Саудівській Аравії та Японії.

Кожна з цих країн розвиває новітні технології в купі з іншими країнами, чи самостійно, для розв'язку нагальних питань своїх жителів. Впроваджуються розумні системи контролю руху автомобільного транспорту, пошуку парко місць, системи централізованого контролю систем водопостачання, слідкування за витратами споживання електроенергії, системи простежування руху громадського

транспорту, або єдині системи оплати проїзду в них, а подекуди навіть муніципальні системи безпеки та відео фіксації правопорушень.

Все це потребує великого вливання грошових та інтелектуальних ресурсів, створення складної багаторівневої інфраструктури, розробки систем контролю та моніторингу пристроїв, великі штати персоналу з обслуговування та інженерів що впроваджують та налагоджують складні та багатофункціональні мережі, створення центрів з обробки великої кількості інформації та великих даних, командні пункти з управління інфраструктурою.

Володіючи цією інформацією можна скласти впевнене враження, що на інтелектуальні системи IoT є всесвітній тренд, і кожній нації необхідні досвідчені спеціалісти для винайдення і впровадження розрізнених систем автоматизації, а також те що кожна країна виділяє досить велику кількість ресурсів для створення корисних для суспільства інфраструктурних рішень.

Розглянемо новітні доробки у сфері інтернет речей більш детально на прикладі окремих міст.

Іспанія займає перші щаблі серед країн Європи по впровадженню і вдосконаленню інтелектуальних систем. У двох найбільш густонаселених містах, Барселоні та Мадриді, вже встановлені сотні датчиків що постійно повідомляють про загальні обставини у місті. Спеціальні пристрої вимірюють, від загальних параметрів, наприклад, температури, рівня освітлення, рівня шуму і дорожньої ситуації до навіть спеціальних сенсорів сповіщення про наповненість смітєвих баків по місту. Муніципальні служби з вивозу сміття, догляду за парковими зонами переміщуються містом за автоматично створеними графіками, що дозволяє ефективно економити, як і пальне, так і людські ресурси. Дані отримані центрами консолідації даних допомагають міським службам планово та ефективно прокладати нові комунікації, дороги, зводити нові будівлі, або переорганізовувати старі простори міста під сучасні задачі й виклики. Також дані що продукуються сучасним мегаполісом грають ключову роль в організації руху громадського транспорту, і створенні нових маршрутів, що глобально розвантажують міські автотранспортні комунікації. Всі данні що надходять з датчиків є відкритими для

широкого загалу, що спонукає приватні компанії до розробки все нових та вдосконалених систем управління непередбачуваним міським життям.

В столиці Австралії муніципальні служби також розкидують сітку датчиків для моніторингу критично важливих для життя великого міста даних. Спеціально розроблені алгоритми на основі даних що продукуються транспортними артеріями міста, регулюють роботи пристроїв регуляції автомобільного руху. На основі цілодобового моніторингу завантаженості доріг алгоритми регулюють світлофори, що дозволило зменшити кількість і протяжність заторів на дорогах на 40%, час що необхідний кожному на подолання міських маршрутів зменшився на 20%. Інтелектуальною візитівкою Сіднею можна вважати системи контролю якості повітря. Кожен резидент сучасного міста одночасно з прогнозом погоди може побачити інформацію про склад повітря, наявність в ньому важких і шкідливих елементів, спеціальний алгоритм на основі цих даних навіть розраховує такий параметр як прозорість повітря. Такі дані можуть бути критично важливими для кожної людини, щоб розуміти як саме змінюється екологічна ситуація навколишнього довкілля.

Дещо схожа система впроваджується в Батумі, Грузія. Основна тенденція розгортання інтелектуальних мереж направлена на розробку досконалішої системи клімат контролю. Планується розробка алгоритмів сповіщення людей про погодні умови, та видачу рекомендацій у випадку погіршення погоди. Така система стане вкрай корисною для розвитку аграрного сектору економіки країни, а також для планування архітектурного розвитку міста.

Міська адміністрація Лондону також впроваджує розумні технології у життя своїх містян. Окрім інтелектуальних алгоритмів регуляції руху, систем орієнтування жителів у величезній, розгалуженій системі вулиць та маршрутів громадського транспорту, систем оповіщення змін в організаційному та інфраструктурному житті міста, систем відео нагляду для організації безпеки на вулицях міста, в Лондоні реалізована статистична аналітична система (САС). Основна задача САС полягає в завчасному повідомленні міських служб про потенційно схильні до займання будівлі. Аналітична система обробляє низку

демографічних даних (приблизно 60 критеріїв) кожного окремого жителя міста, таких як вік, стать, освіта, рівень доходів, сімейний стан, район проживання, тип зайнятості, а також історичні, географічні, демографічні показники, що дозволяє службам міста скласти оцінку ризику займання будівель. На основі особистих даних містян і схильних до займання будівель міста, владі вдалося знизити ризики займання будівель, зменшити кількість займань і ефективно реагувати на повідомлення про пожежу, щоб уникнути повторення історії 1666 року, коли було вщент спалена більша частина міста.

Нью-Йорк також не пасе задніх. Системи датчиків по всьому місту допомагає не тільки ефективно налагоджувати роботу комунальних служб що прибирають просто величезні обсяги сміття що продукуються мегаполісом, а навіть повідомляти про правопорушення. Система датчиків по всьому місту реєструє вібрації від пострілів вогнепальної зброї, визначають місце де був постріл і оперативно повідомляють правоохоронні органи, що одразу вирушають на місце потенціального злочину. Також працює подібна до САС система передбачення пожеж, що до 70% покращила ефективність перевірки об'єктів на предмет дотримання протипожежних заходів. Також в місті, компанією IBM було впроваджено систему контролю за здоров'ям містян. Центри обробки даних збирають показники з особистих пристроїв, на кшталт розумних годинників, фітнес браслетів чи кардіостимуляторів, що допомагає лікарям ставити діагнози на основі показників цих пристроїв.

На прикладі Ріо-де-Жанейро можна побачити як інтелектуальні системи можуть допомагати рятувати життя людей. Після руйнівних сходжень ґрунту в деяких районах міста, адміністрація міста сумісно з IBM впровадила систему раннього оповіщення людей про небезпечні погодні умови, що дозволяє завчасно, чи дуже швидко направити в райони, що знаходяться в небезпеці від непогоди, рятувальників.

Окрім муніципальних систем слід особливо виділити корпоративні виклики до інтернету речей.

Більшість сучасних підприємств з виготовлення товарів спожитку або побуту мають невеликі штати співробітників. Основною тенденцією вироблення того чи іншого товару є налагодження автоматизованої лінії її виготовлення, що переймає на себе необхідність створення великого штату майстрів, навчених виробляти цей самий товар. На великих підприємствах все частіше можна зустріти механічні руки та конвеєрні лінії разом з інженерами по ремонту і догляду за ними, замість станка за яким необхідна постійна присутність спеціаліста. Майже всі автомобілі (за винятком авто елітного класу) збираються автоматично, на чітко сформованій та вивірених конвеєрній лінії, де всі необхідні операції по збиранню та зварюванню окремих функціональних частин лежать на автоматиці, що з однаковою точністю і невпинністю можуть створювати готовий технологічний виріб. Людині не треба постійно слідкувати за процесом збирання виробу, достатньо проводити перевірку на відповідність виробу вимогам, але і те вже також виконується із застосуванням роботизованого зору, що позбавлене недоліків людського ока, і завжди гостро і неупереджено бачить недоліки, і не може випадково пропустити ту чи іншу ваду. На передових станціях обслуговування, нехай то буде склад чи поштове відділення все більше використовують електронні реєстри обліку речей та товарів, а в коморах компанії Amazon, разом із людьми між величезними стелажми їздять спеціальні роботи, що перевозять на собі невіддільні для людини вантажі. Компанія Google хоче замінити кур'єра на розумного дрона що в обхід напруженого міського трафіку, вчасно доставить товар.

На аграрних підприємствах все частіше можна зустріти інтелектуальні системи клімат контролю. Немає потреби постійно слідкувати за процесом вирощування тих чи інших культур, розумні датчики самі підкажуть коли є необхідність в поливі, а коли поливати культуру не слід. Такі системи дуже допомагатимуть виростити хороший врожай, а автоматичні системи поливу заберуть на себе необхідність постійного перебування на полі персоналу компанії, що звільняє робочі руки, а головне розум для, наприклад тоншої, селекційної роботи.

Приклади інтелектуальної між машинної взаємодії особливо можна побачити в банківській системі, адже саме в цьому секторі людського господарювання зароджувалися перші автоматизовані взаємодії. Людині більше немає потреби мати свого власного рахівника, який буде виписувати чеки та розпоряджатися вашими закордонними придбаннями, всі транзакції здійснюються і перевіряються автоматично, у два клаца, а в новітніх банках вже навіть немає фізичних відділень, де стоять черги охочих перевести готівку своїм родичам, або забрати пенсійне надходження. Достатньо витратити до 10 хвилин, щоб налаштувати застосунок що зможе надати вам повний спектр банківських послуг і територіально буде знаходитись у вашій кишені, без перерв і вихідних.

Якщо ще більше занурюватися в цю тему можна навести в приклад Китай. Ні для кого не секрет що все більше державних послуг також переходять у цифрове поле, для більш зручної та швидкої взаємодії різних держустанов між собою. В Китаї майже усі послуги що тільки необхідні громадянину доступні лише в одному застосунку. Завдяки цьому людина може придбати квитки на літак, або замовити необхідний документ, зв'язатися з рідними й знайомими, оплатити комунальні платежі або продовольчі товари в будь-якому супермаркеті країни. Якщо подивитися більш широко на цифрову інфраструктуру Китаю можна згадати про систему соціального рейтингу що поступово впроваджується по всій країні прямо зараз. Фундамент цієї системи дуже добре перекликається з темою дипломної роботи, оскільки в її основі лежить обробка великих об'ємів даних, що стікаються в центри обробки даних звідусіль, в основному з камер відеоспостереження та особистих пристроїв. Вона покликана регулювати суспільне життя держави починаючи з кожного окремо взятого громадянина, спонукати його робити суспільно прийнятні вчинки, і попереджати широкий загальний поведінку є неприйнятною для тих норм що держава просуває в широкі маси. Ця система побудована на системах розпізнавання обличчя, і вона цілодобово виявляє порушення в поведінці людей, будь-то скоєння правопорушення, чи не настільки значущі вчинки, як, наприклад, перехід дороги в недозволеному місці, чи на червоний колір світлофора. На основі зібраних даних людині приписується

соціальний рейтинг що вказує на його рівень в суспільстві та якою мірою його поведінка може вважатися взірцевою для оточення. Громадяни з хорошим рейтингом можуть мати право на заохочувальні знижки на білети в транспорті, вигідні умови для отримання кредитування, чи претендувати на апартаменти в районі міста з хорошим рівнем життя і соціальним рейтингом. Люди чия поведінка владою вважається неприйнятною, втрачають такі можливості, втрачають право користуватися авіасполученням всередині країни, а також виїжджати за кордон.

Окрім бізнесу, соціальної та адміністративної сфери є ще одна і дуже важлива ніша застосування інтелектуальних систем і систем IoT. Це приватне, особисте життя людини. Так, вона дещо перекликається з підприємницькою сферою, як сферою надання послуг, але має дещо інший характер.

Вже сьогодні людину оточують інформаційні системи, і більшість людей що проживають в цивілізованому світі мають особисті обчислювальні пристрої, як от телефон чи комп'ютер. Деякі пристрої можуть збирати критично важливі дані про здоров'я людини: пульс, частоту дихання, температуру. Для прикладу, саме зараз в розумні годинники компанії Apple додають спеціальні сенсори для реєстрації таких даних. Спеціальні програми моніторингу цих показників інформують людину про стан її здоров'я, зміни що можуть вказувати на початок захворювання, що в перспективі дозволяє діагностувати його до того як було нанесено катастрофічний вплив на здоров'я, та навіть викликати швидку для порятунку життя людини, чим сама компанія неодноразово хизувалася на презентаціях своїх продуктів. В більш масштабному використанні таких застосунків можливо запобігати навіть глобальним проблемам, як показує сучасна ситуація у світі, таким як епідемії чи навіть пандемії. Сучасні розумні годинники можуть навіть виміряти рівень кисню в крові що дозволяє попередити людину про початок захворювання на коронавірус, що зупинить розповсюдження зарази, дозволить людині вчасно звернутися за допомогою до лікаря. В масштабі планети це дозволить ефективно боротися з захворюваннями, а згодом стримувати розповсюдження хвороби, що вбереже незліченну кількість людських життів і дасть світу більше часу на винахід розв'язання таких скрутних проблем.

Але користь інтернет речей не обмежується сферою охорони здоров'я. Все частіше можна почути про «розумні будинки». На ринку вже є велика кількість представлених різними компаніями рішень для своєї оселі що дозволяє контролювати опалення та кліматичні показники, збільшувати ефективність використання електроенергії, що є неоціненним вкладом в екологію планети, і звісно ж організувати безпеку власного майна. Щодо сигналізаційних систем і організації безпеки ми вже неодноразово згадували в даній роботі, що на рахунок споживання енергії?

Так, можна довго перераховувати готові комплексні системи що представляються широкому загалу великими ІТ компаніями, але вони в основному зосереджені лише на аспекті безпеки оселі, наочніше буде розглядання окремих відомих випадків інтелектуального підходу до організації житлового простору, на прикладі «розумних будинків» відомих людей.

Відомі люди, власники величезних підприємств, в тому числі пов'язаних зі сферою ІТ, не просто мріють про інтелектуальну систему власного житла, а і втілюють свої задуми у реальність. Так, наприклад, більшість топ бізнесменів і зірок мають особисту інтелектуальну систему у себе вдома, так би мовити під ключ, розроблену з врахуванням особистих вимог і бажань. Серед таких людей Ілон Маск, Девід та Вікторія Бекхем, Марк Цукерберг, Ріанна, Білл Гейтс, Опра Вінфрі, Девід Ян, Дженіфер Еністон, Пірс Броснан та ще точно багато зірок та підприємців. Між собою їхні системи розумних осель не так вже й сильно різняться, але є й особливі, аналогів яким у світі може навіть і не знайти. Основа такої інтелектуальної системи в основному одна: ефективне розпорядження електроенергією, опаленням та системою водопостачання. В основному, обчислювальний центр будівлі регулює освітлення, приглушаючи його або вимикаючи якщо в кімнаті ніхто не перебуває, автоматизований термостат регулює температуру в оселі так, щоб використовувати якнайменше електроенергії й підтримувати власний мікроклімат в середині будівлі, виходячи зі смаків домівників. В оселі Пірса Броснана, визнаного борця за екологію, система водопостачання взагалі замкнутого циклу, тобто вся вода фільтрується

спеціальним устаткуванням всередині будівлі й використовується повторно, та й взагалі, концепція автономності власного будинку заслуговує визнання, його оселя самотужки виробляє електроенергію і не вимагає підключення до «брудних» джерел енергопостачання. А от в Ілона Маска розум будинку під'єднаний до особистого автомобілю, тому основні системи клімат контролю вмикаються тільки коли господар наближається до свого будинку, завчасно створюючи приємні умови всередині. До речі, будинок Опри Вінфрі взагалі оснащений власною метеостанцією, що і виставляє термостат всередині оселі, а також доглядає її особисту оранжерею, мається навіть розумний асфальт що прибирає її подвір'я від снігу без лопати, просто нагріваючись і розтоплюючи його.

Для власного комфорту в розумному будинку має бути гнучка і зручна система управління, і для ІТ магнатів це вже не така і непосильна задача, особливо в купі із сучасними технологіями. Так, наприклад, у власника компанії Facebook, Марка Цукерберга, оселею опікується електронний дворецький, на ім'я Джарвіс (це ім'я запозичене із фільму про залізну людину, там штучний ІІІ з іменем Джарвіс опікувався оселею відомого героя із коміксів). Цукерберг підійшов до створення домашнього затишку ґрунтовно, готові рішення у сфері інтелектуального управління господарством його не вразили, тому єдиним виходом із ситуації було зайнятися його створенням самотійно. Його електронний помічник може розрізняти мешканців по голосу і запам'ятовувати їх вподобання, смаки в музиці та інші забаганки. Дещо подібне у власному маєтку має і Білл Гейтс, засновник компанії Microsoft. Його оселя так і рясніє безліччю хай-тек систем, що допомагають доглядати за кліматом, садом і безпекою величезної ділянки на березі озера Вашингтон (одна тільки площа будинку понад 7000 кв. м). Організація адміністрування доволі проста, на в'їзді на територію маєтку гостям пропонується причепити на одяг шпильку-датчик, що допоможе орієнтуватися на такій величезній території, і якщо ви загубитесь, допоможе охороні вас знайти. Також таким чином розумне ядро будинку одразу обмежить зони доступу для вас та підбере музику до смаку що буде супроводжувати вас прогулянкою по території. Особливою гордістю маєтку для її господаря слугує дуже гнучка система

мікроклімат контролю. Показники температури та вологості власні для кожної кімнати, і також підлаштовуються під смаки мешканця. Так, наприклад, в особистій бібліотеці Біла, де зосереджена велика кількість друкованих видань різноманітної художньої та освітньої літератури підтримується особливий рівень вологості та температура, щоб запобігти псуванню такого дорогоцінного скарбу від плісняви та інших чинників, а всілякого роду шкідники та комахи знищуються лазерами прямо у польоті як в якому-небудь науково-фантастичному фільмі. Окрім цього на більшості стін можна зображати що завгодно за допомогою смарт екранів, а також, за допомогою низки датчиків, система може піклуватися за рослинами на прибудинковій території й особистим кленом Гейтса.

Якщо людина одного разу захоче скинути більшу частину своїх турбот на штучний інтелект, що в наш час не здається таким вже неймовірним, то більшість пристроїв в обов'язковому порядку мають бути під'єднані до мережі, щоб мати можливість отримувати команди керування. Саме тому власник компанії АBBYY, Девід Ян, створює власну, не просто «розумну» оселю. Його мета полягає в тому щоб створити штучний інтелект, який буде на основі власного досвіду «вирішувати» робити йому щось чи ні. Він створюється на основі технології «unsupervised machine learning», що в перекладі означає «машинне навчання без нагляду», що і буде ставити вибір перед інтелектом перед виконанням завдань. Штучний інтелект Девіда отримав ім'я Морфеус (як ім'я героя фільму «Матриця», що за сумісництвом був однією з підсистем створеного людиною цифрового світу, Матриці). Окрім основних задач він запрограмований вивчати філософські твори та писати власні есе на різних мовах, що дасть йому в купі з «unsupervised machine learning» непередбачуваний характер і більше свободи у порівнянні зі звичайними голосовими помічниками. Сама ідея втілення даного проєкту радше схожа на експеримент, чи вдасться штучному інтелекту опанувати свободу думки та розвинути справжній «інтелект», щоб надалі стати вільним.

Коротко резюмуючи даний розділ, можна дійти висновку, що сфера IoT несе в собі велику кількість користі для більшості, якщо не всіх, сфер людської життєдіяльності, та є необхідною та важливою для впровадження, для подальшого

розвитку людського суспільства в цілому, для збільшення ефективності в розпорядженні ресурсами доступними людині й роботі що виконується нею, так і для перспективи втілення інноваційних технологій, які вже в наш час передбачаються передовими науковцями та аналітиками, в повсякденне життя людського суспільства.

## 2 АНАЛІЗ ТЕХНОЛОГІЙ І СТАНДАРТІВ ДЛЯ ПОБУДОВИ МЕРЕЖІ ІОТ

### 2.1. Канали передачі даних

Для того, щоб створити надійну та ефективну мережу для використання інтернет речей, яка відповідає всім вимогам і якісно виконує поставлені перед собою задачі перш за все слід визначитися з компонентами та функціональними елементами з яких буде складатися мережа. Основою на якій будується мережа засобів ІоТ є канали зв'язку і передачі даних.

На перший погляд, їх не так вже і багато, може бути дротове і бездротове з'єднання, але якщо почати досліджувати це питання, то виявиться, що фізичні (провідні) канали передачі даних мають різні властивості, різну пропускну здатність, і різні характеристики які необхідно враховувати при безпосередньому прокладанні кабелів, так і бездротові канали передачі даних можуть бути зав'язані на різних технологіях, що в основі своїй хоч і створені на основі передачі радіосигналів (хоча не усі), але також мають різні характеристики та дальність взаємодії, що можуть бути відповідними далеко не в усіх проектах.

Дротові канали передачі даних є більш надійними й захищеними в плані безпеки. По перше сигнал в таких каналах стабільний і якщо цілісність кабелю не порушена, то вони виконують усі задачі що перед ними поставлені. Також через дротові канали передачі даних можна передавати електроенергію для живлення пристроїв що потребують велику кількість електроенергії. У виділеному каналі неможливо перехопити пакети передачі даних, а втручання в цілісність каналу є трудомісткою і непосильною задачею, особливо якщо зловмиснику невідоме розташування цього самого каналу, до того ж таке грубе втручання у цілісність лінії можливо помітити, і в короткий час знайти та знешкодити загрозу безпеці системи. Також слід зазначити незалежність даного методу від кліматичних та більшості атмосферних особливостей місцевості.

Щодо недоліків даного методу слід виділити коштовність монтажу у порівнянні з бездротовим з'єднанням, що зростає пропорційно довжині

прокладання ліній передачі. Прокладання фізичних каналів передачі зв'язку може бути недоступна для важкодоступних місць, або через особливості рельєфу. Основним недоліком такого методу є неможливість під'єднання рухомих об'єктів.

Якщо розбирати особливості різних доступних фізичних середовищ передачі інформації для застосування їх для «розумних» датчиків та різних виконавчих приладів, можна обмежитись дешевшими проводами, з меншою пропускнуою здатністю, позаяк дані засоби, по одинці, не потребують передачі великих обсягів інформації, а також досить часто можна обійтись вже наявною інфраструктурою та каналами зв'язку що вже прокладені (така особливість доступна завдяки універсальності каналів передачі мультимедійної інформації), що значно зменшує суму кошторису проєкту.

На ринку доступні декілька типів кабелів спеціалізованих для передачі інформації. В основному для подібних мереж та мереж інтернету використовують оптоволоконні кабелі, коаксіальні кабелі та кабелі «вита пара». Всі вони мають різну дальність передачі даних, та мають різне застосування.

Оптоволоконні кабелі (Рис. 2.1.1.) є найшвидшими для передачі інформації, і в основному, через велику ціну устаткування що використовується для передачі, використовуються як магістральні канали передачі даних. Саме за допомогою такого середовища передачі даних об'єднуються мережі різних континентів, шляхом прокладання товстелезних кабелів з величезною пропускнуою здатністю дном океанів що їх розділяють. Передача інформації в даному середовищі здійснюються закодованими світловими сигналами лазерів чи світлодіодів, що обмежує швидкість передачі даних лише швидкістю світла та пропускнуою спроможністю устаткування що кодує пакети з даними у світлові сигнали. Але такі кабелі також мають свої недоліки. Через конструкцію кабелю його не можна сильно згинати, що робить неможливим монтаж у важкодоступні місця де є потреба викривляти провід, також такі кабелі є досить крихкими й дорогими.

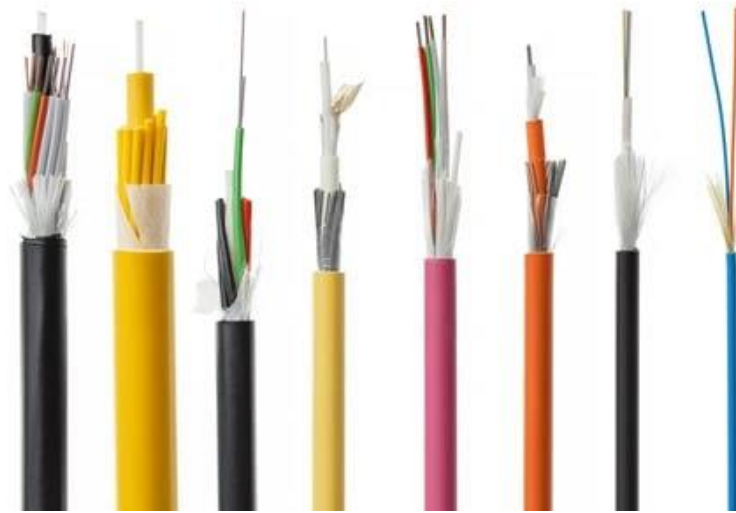


Рисунок 2.1.1 -- Приклади різних оптоволоконних кабелів

Коаксіальні кабелі є досить надійною технологією передачі інформації. Конструктивно це звичайний провід із міді або іншого провідного металу (Рис. 2.1.2.), з декількома шарами ізоляційного, а іноді ще й з захисного матеріалу. Кодування сигналу відбувається як і в інших подібних кабелях, різницею електричного сигналу. Такі кабелі використовуються для прокладання ліній зв'язку, і в залежності від типу кабелю можуть мати різні характеристики дальності і швидкості передачі даних. В основному такі кабелі монтують у стіни, прокладають по комунікаціях між будинками або навіть між містами.



Рисунок 2.1.2 – Будова коаксіального кабелю

Кабель типу «вита пара» мають не таку велику дальність передачі інформації як його побратими, але на відміну від них має повнодуплексну конструкцію що дозволяє йому передавати інформацію одночасно в обидва боки. В середині це вісім попарно звитих між собою кабелів, що можуть додатково вкриватися екрануючим шаром, для захисту від перешкод. Така конструкція своєю чергою також зменшує кількість завад у каналі, компенсуючи це витком з кабелем в якому струм тече в протилежний бік. Такі кабелі в основному використовуються в середині будівлі, для під'єднання ПК або мережевого обладнання до загальної мережі. В залежності від типу (Рис. 2.1.3.) кабелю може мати різну швидкість передачі даних, а також дальність передачі.

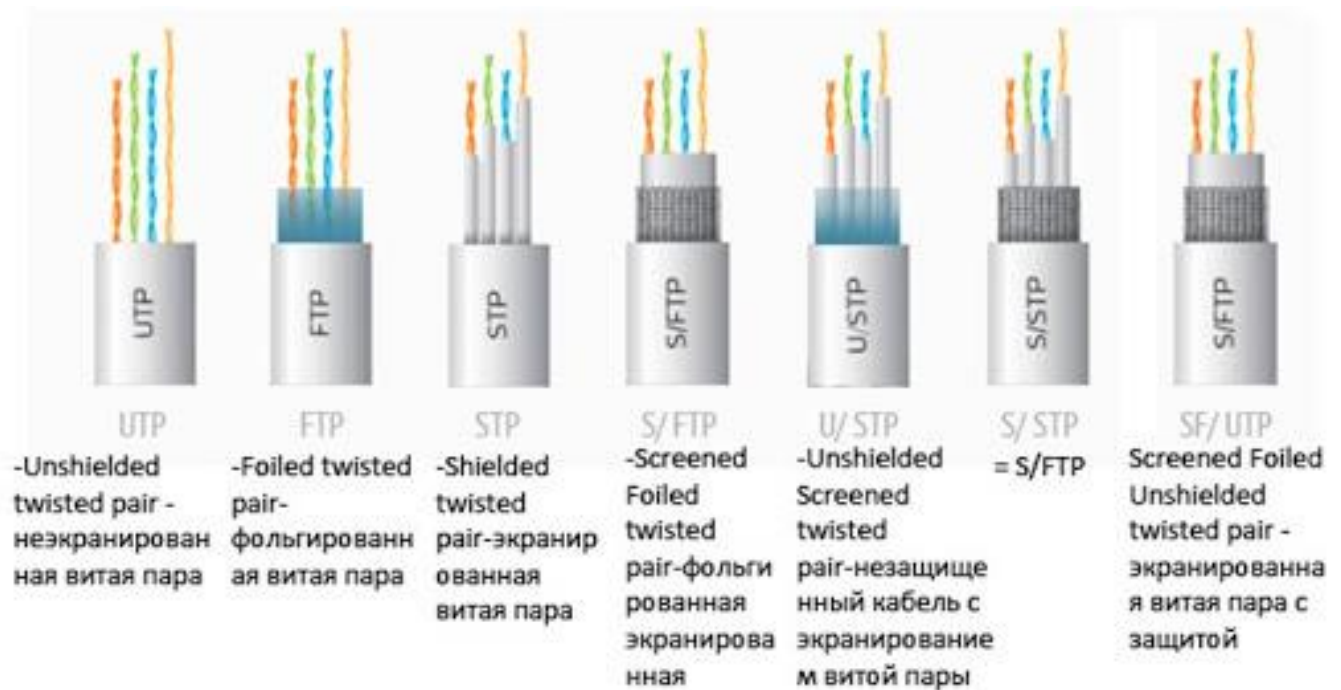


Рисунок 2.1.3 – Типи кабелів «вита пара»

Щодо бездротової передачі даних то тут також є кілька альтернатив. В основному інтернет речі можуть бути під'єднані до мережі через WiFi, але виходячи з поставлених завдань датчик можна під'єднати за допомогою бездротового сотового зв'язку (LTE), чи напряму до радіоканалу передачі даних (WiMax). Якщо є необхідність, мережу або датчик також можна під'єднати до супутникового зв'язку, але швидкість в таких каналах, поки що, досить низька, але

в цій сфері активно ведуться розробки, і навіть зараз будуються мережі супутникового інтернету (приклад Starlink компанії Ілона Маска).

До недоліків такої середі передачі даних можна віднести більшу вразливість для зловмисників (дані передаються у відкритому і, в основному, в нешифрованому вигляді, що дозволяє «підслухати» про що іде передача даних). Для передачі даних через LTE необхідно щоб пристрій постійно був в зоні доступу вежі мобільного зв'язку, що може бути недоступно в малонаселених кутках планети, а також може бути досить дорого, в залежності від умов оператора зв'язку що обслуговує вежі. Пряме підключення WiMax також дуже хороша альтернатива бездротового зв'язку, але для неї необхідна пряма видимість між передавачем і отримувачем, а також така технологія може бути досить дорогою для невеликого бізнесу. Всі вище перераховані системи бездротового зв'язку мають відносно невелику пропускну здатність у порівнянні з дротовим з'єднанням, але все ж достатню для передачі інформації від розумних датчиків.

Окремо слід виділити з'єднання пристроїв через мережу WiFi. Даний метод є найрозповсюдженішим серед інших технологій бездротової передачі даних, і на сьогодні, майже у кожного вдома є роутер і налагоджена мережа WiFi, для виходу у всесвітню павутину. Така середа передачі даних є не дуже дорогою, і має радіус передачі 30-50 метрів навколо, що дозволяє покрити повністю територію житла, або невеличкого офісу, а сама технологія досить надійно захищена і дозволяє легко масштабувати мережу, додаючи нові точки доступу.

Обидва типи з'єднання пристроїв між собою мають як свої переваги, так і недоліки, але справжні мережі з використанням датчиків і виконавчих пристроїв скоріш за все будуть з'єднані і дротовими каналами передачі даних, і бездротовими, що створює необхідність в універсальному методі передачі даних від датчиків і команд на виконавчі пристрої.

## 2.2. Протоколи і стандарти для IoT пристроїв

З каналами передачі інформації ми розібралися, але як саме нам передавати команди через них? На це питання мається відповідь в інститутах з розробки та впровадження нових інформаційних технологій та стандартів, таких як ISO, RFC, IEEE та інших. Їх основною задачею є розробка нових стандартів таким чином, щоб підтримувати здорову конкуренцію серед компаній, які бажають зайнятися розробкою нових інформаційних технологій, шляхом створення вільного та зрозумілого середовища, що надалі стає базою інфраструктури мультибрендових застосунків. Саме у вже створених та впроваджених стандартах слід шукати глибинне розуміння процесів що будуть відбуватися в нашій мережі, а це своєю чергою дасть нам змогу створити програму керування цими процесами.

Перш за все слід згадати на чому саме базується так добре нам всім відомий інтернет, в контексті глобальної розрізної «мережі мереж». Слід згадати логічну структуру інтернету під назвою OSI (Рис. 2.2.1), запропоновану міжнародною організацією по стандартизації ISO. Вона має сім різних рівнів взаємодії між мережного обладнання, починаючи з фізичного рівня, передачі інформації у каналі, закінчуючи прикладним рівнем, що передбачає отримання і сприймання цієї інформації користувачем. З фізичним рівнем ми розібралися, це безпосередньо фізичне середовище передачі інформації, як то кабель чи радіохвиля, де сигнал кодується коливанням потенціалів електричного струму. Канальний рівень, що йде слідом за фізичним, більшою мірою нас не цікавить, на ньому відбувається передача пакетів даних між мережевим обладнанням, тому слід звернути увагу на наступний, транспортний рівень. Саме тут від логічної моделі побудови інтернет взаємодії ми можемо перейти до безпосередньої моделі мультимедійної мережі під назвою TCP/IP (Рис 2.2.2), яка складається з двох найпоширеніших протоколів третього логічного рівня моделі OSI, транспортного. Ці два протоколи створюють адресацію в середині мережі (протокол IP, internet protocol, в спрощеному представленні являє собою адресу хостів що беруть участь у передачі інформації, на прикладі поштової адреси будинку, для того, щоб пакети даних не губилися під

час передачі, а мережеве обладнання чітко знало куди їх слід відправляти), та транспортування (протокол TCP, transmission control protocol, регламентує порядок кодування інформації в самому пакеті даних (в основному у двійковий код) для його подальшого коректного читання мережевим обладнанням, а також контролю цілісності даних, що передаються декількома різними пакетами, і можуть дійти до отримувача різними маршрутами, більш спрощена версія цього протоколу має назву UDP, user datagram protocol, також регламентує кодування інформації в пакеті даних, але не перевіряє їх на порядок повного надходження отримувачу, що робить його менш вимогливим до ресурсів мережі, і більш привабливим для передачі інформації для якої втрата 1-2 пакетів не є суттєвою, а швидкість передачі навпаки (потоків аудіо та відео, службова інформація мережевого обладнання чи сенсорів)). Саме на їх основі були створені сучасні уніфіковані стандарти, що використовуються в мережах інтернет речей.

### Модель OSI

Дані	7 прикладний application	Доступ до мережевих служб
	6 представлень presentation	Представлення і кодування даних
	5 сеансовий session	Управління сеансом зв'язку
Сегменти	4 транспортний transport	Прямий зв'язок між кінцевими пунктами і надійність
Пакети	3 мережевий network	Визначення маршруту і логічна адресація
Кадри	2 каналний data link	Фізична адресація
Біти	1 фізичний physical	Робота з середовищем передачі, сигналами і двійковими даними

Рисунок 2.2.1 -- Модель OSI

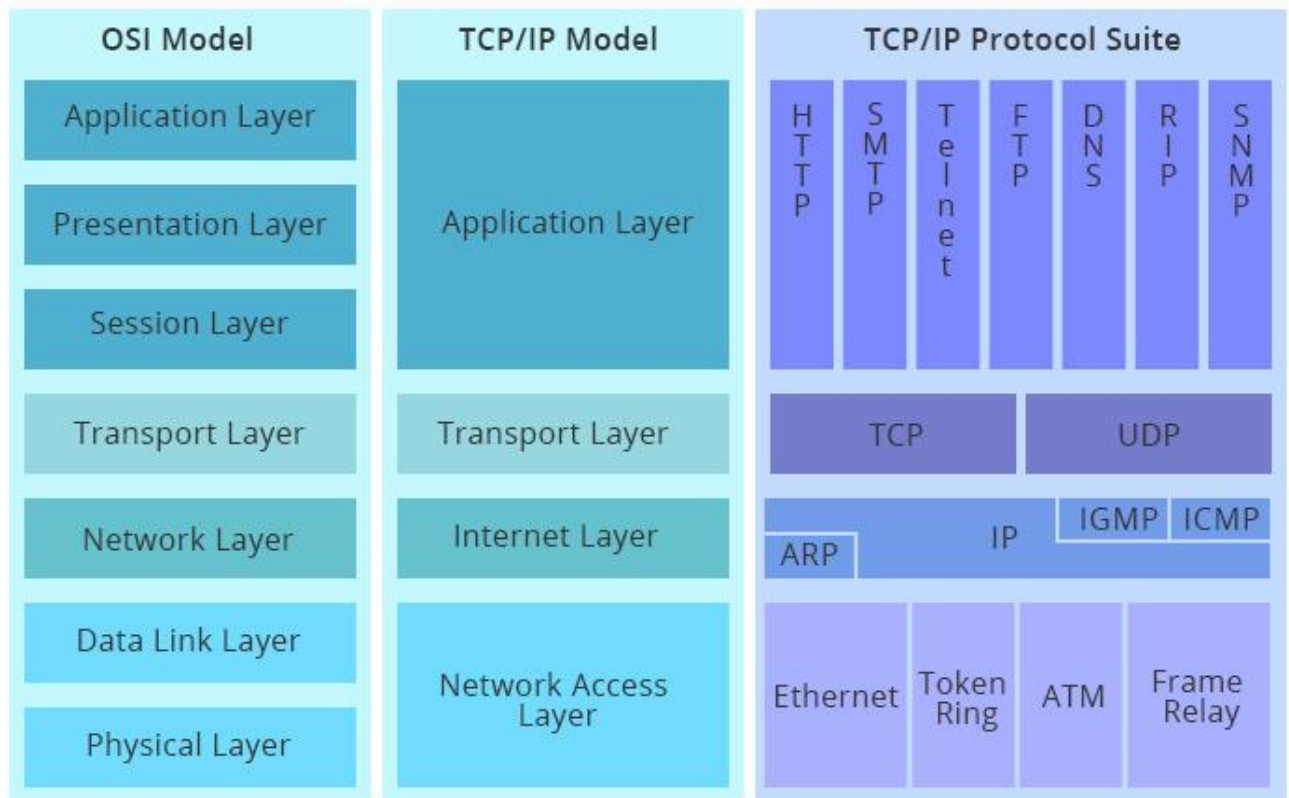


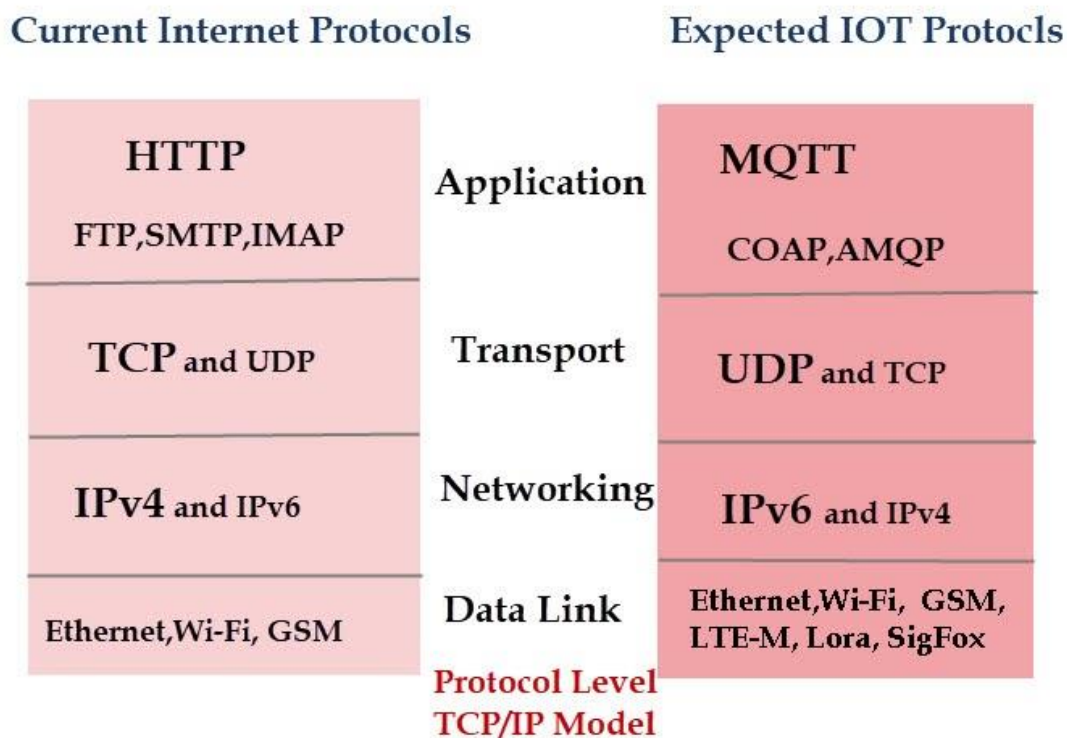
Рисунок 2.2.2. -- Порівняння моделей OSI та TCP/IP, а також протоколів між мережевої передачі даних

Перед мережами для використання виключно, чи переважно інтернет речей, датчиків та виконавчих пристроїв стоять свої задачі та вимоги. Розумні пристрої в основному невибагливі, тому для них достатньо каналів передачі даних з невеликою пропускнуою здатністю, невеликим споживанням електроенергії, тому впровадити навіть складні інфраструктури з інтернет речами у звичайну комунікаційну мережу яку використовує людина досить просто. Але якщо мережа будується «з нуля», і для інтернет-речей, ефективніше та економніше буде проектувати її з невеликим запасом потужностей. Та попри це, для таких мереж звичайні інтернет-протоколи не підійдуть. Великі компанії, з достатком що дозволяє вести клопітку розробку нових технологій, в основному розроблюють власні, приватні протоколи, для взаємодії пристроїв між собою, але для того, щоб пришвидшити процес впровадження новітніх розробок, інститутами були розроблені загальнодоступні стандарти, щоб спонукати до конкуренції невеликі контори, що також хочуть створювати та впроваджувати нові «розумні мережі».

До таких стандартів відносяться:

- CoAp (Constrained Application Protocol);
- DDS (Data Distribution Service);
- XMPP (Extensible Messaging and Presence Protocol);
- MQTT (Message Queue Telemetry Transport);
- AMQP (Advanced Message Queuing Protocol);
- STOMP (Simple (or Streaming) Text Oriented Message Protocol)

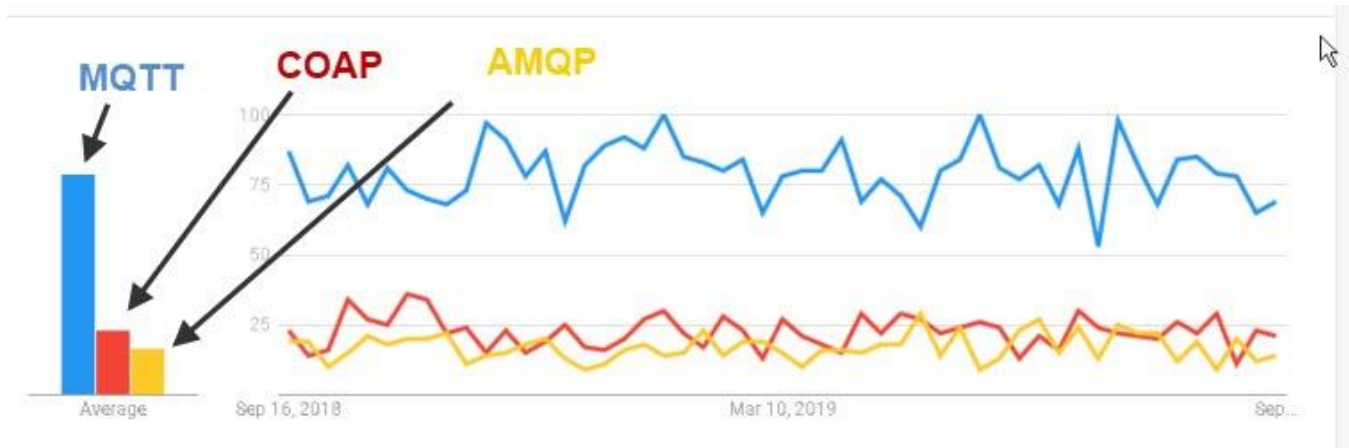
Кожен з них був розроблений як «простий» протокол, спеціально для мереж з використанням датчиків та сенсорів. Деяким з них можна поставити у відповідність протоколи що використовуються у звичайних мережах, так, наприклад аналогічним до CoAP (і деяким іншим) є всім відомий HTTP протокол (Рис. 2.2.3).



## IOT and Internet Protocols

Рисунок 2.2.3 -- Відповідність звичайних мережевих протоколів спеціалізованим протоколам інтернету речей.

Кожен з них має свої особливості передачі й обміну даними між пристроями, тому для деяких, спеціальних задач, необхідно підібрати відповідні протоколи, але в основному глобальні задачі що мають вирішуватися цими протоколами однакові, тому найбільш ходовим та використовуваним можна виділити протокол MQTT, та його більш модернізовану версію MQTT-SN (MQTT for Sensor Networks) що використовує швидший транспортний протокол UDP, для швидшого відгуку пристроїв і використання в бездротових мережах. Друге та третє місце по використанню в мережах інтернет речей ділять між собою CoAP і AMQP (Рис. 2.2.4).



### Google Trends MQTT,COAP and AMQP

Рисунок 2.2.4 -- Порівняння популярності протоколів IoT на ресурсі Google Trends

### 2.3. Constrained Application Protocol

Протокол CoAP спеціально розроблений для обмежених пристроїв, і специфікований RFC 7252. Він дозволяє обмінюватися даними пристроям в одній локальній мережі з загальною мережею інтернет. Для транспортування він використовує транспортний протокол UDP, тому по стандарту повний пакет даних CoAP має вміщатися в одну дейтаграму. Пакет даних цього протоколу без корисного навантаження, керівних бітів і токена вміщується всього в 4 байти (Рис. 2.3.1), що дозволяє використовувати його в мережах з великими втратами та

невеликою пропускною здатністю. Також одним з переваг даного протоколу є наявність багатоадресної розсилки.

Bit:	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
	Ver	T	TKL		Code				Message ID																							
Token (if any)																																
Options (if any)																																
Payload Marker								Payload (if any)																								

Рисунок 2.3.1 -- Структура даних пакету CoAp

## 2.4. Data Distribution Service

У 2004 році американська компанія Real-Time Innovations і французька Thales Group розробили новий стандарт між машинної взаємодії що отримав назву DDS. Цей стандарт був затверджений і запатентований компанією Object Management Group (OMG), і почав використовуватися в мережах обміну даних в реальному часі. В його основі закладений принцип видавець-підписник, кожний пристрій що використовує цей стандарт створює власну тему (наприклад температура чи рівень освітлення) і постить згенеровані дані по своєму топіку. Інші пристрої що мають потребу у цих даних самостійно дивляться у пакет DDS і отримують інформацію по своєму топіку (Рис. 2.4.1). Дана служба пропонує ефективну взаємодію між пристроями різних виробників, не чутливу до затримок всередині мережі з простою адресацією по IP, а також, за потреби, перетворення локальних даних, вбудовування в транспортний рівень і збереження метаданих.

<i>&lt;class name&gt;</i>		
attributes		
	<i>&lt;attribute name&gt;</i>	<i>&lt;attribute type&gt;</i>
...		...
operations		
	<i>&lt;operation name&gt;</i>	<i>&lt;return type&gt;</i>
	<i>&lt;parameter&gt;</i>	<i>&lt;parameter type&gt;</i>
	...	...
...		...

Рисунок 2.4.1 -- Структура даних стандарту DDS

## 2.5. Extensible Messaging and Presence Protocol

Даний протокол, оснований на XML, раніше мав назву «Джаббер». Він був розроблений в 1999 році Джереми Міллером, і на початку позиціонувався як єдиний протокол обміну повідомленнями. Він схожий з деякими комерційними системами GTalk, MSN, AIM, Yahoo! Messenger, ICQ, і був розроблений як відкритий протокол що дозволяв спілкування між користувачами різних платформ обміну повідомленнями. Даний протокол може передавати як текстові, так і голосові повідомлення, або навіть файли. Сама ідея обміну повідомленнями між різними месенджерами так і не отримала достатньої підтримки, але все ж таки даний протокол використовується для побудов мереж з використанням IoT пристроїв, для налагодження обміну даних між пристроями. Ще одним із застосувань цього протоколу є адресування пристроїв всередині мережі.

## 2.6. Message Queue Telemetry Transport

Протокол MQTT один із найпопулярніших протоколів для мереж інтернет речей. Він працює на протоколах TCP/IP, що дозволяє дуже легко налагоджувати його роботу в будь-якій мережі. Він працює на основі принципу видавець-підписник, через сервер (брокер) (Рис. 2.6.1), що дозволяє пов'язувати ним пристрої від різних виробників. Даний протокол з самого початку розроблявся як

«легкий» протокол, для роботи в мережах з похибками в каналі та втрати пакетів. Був розроблений Ален Ніппер із компанії Arcom, та Енді Станфорд-Кларком з IBM в 1999 році, і поширювався з роялті-фри ліцензією, а згодом був стандартизований консорціумом OASIS у 2014 році. На його основі був розроблений MQTT-SN, ще один, більш сучасний, протокол що використовує транспортні дейтаграми UDP, тому має швидший відклик, використовується для передачі в бездротових та Bluetooth мережах, і в основному розроблений для різного роду сенсорів миттєвого реагування.

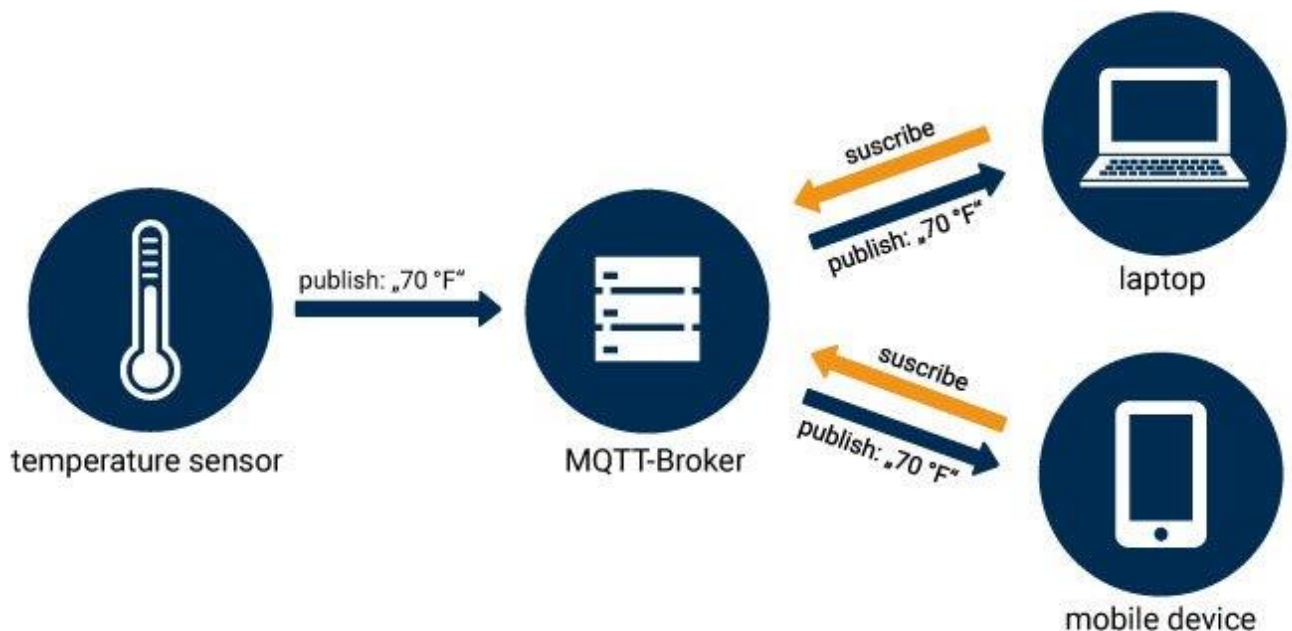


Рисунок 2.6.1 -- Принцип роботи протоколу MQTT

## 2.7. Advanced Message Queuing Protocol

AMQP також протокол заснований на концепції видавець-підписник, винайдений Джоном О Харюю в банку JP Morgan Chase & Co. Він створений для обміну даними (повідомленнями) між різними підсистемами через сервер (брокер), що сам по собі здійснює маршрутизацію трафіку, налаштовує черги отримання даних та гарантує що повідомлення було доставлене потрібному підписнику (Рис. 2.7.1).

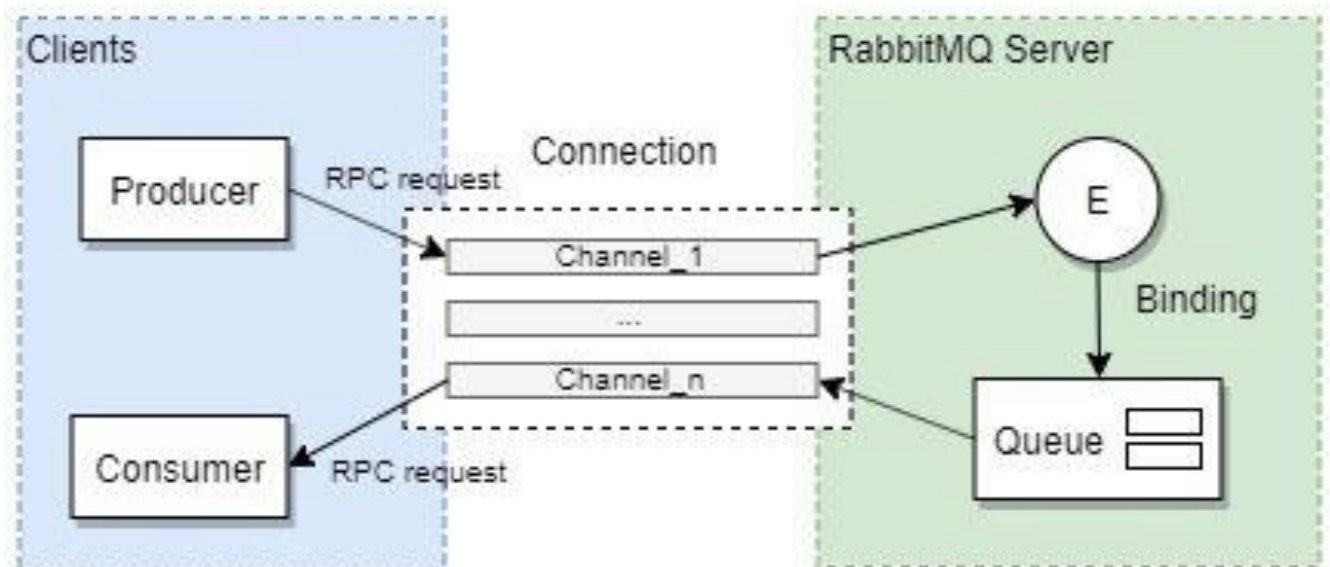


Рисунок 2.7.1 -- Принцип роботи протоколу AMQP

## 2.8. Simple Text Oriented Message Protocol

STOMP, також відомий як TTMP, це ще один протокол передачі будь-яких повідомлень. Він виконаний по аналогії до HTTP, але набагато простішим, що дозволяє його використання в мережах з обмеженими ресурсами. Він також використовує сервер на якій йдуть повідомлення від пристроїв, і надсилає їх підписникам, що мають потребу в цих даних.

## 2.9. Пристрої інтернету речей

Якщо у звичайній мережі учасниками є, або хости (персональні комп'ютери) користувачів, або сервери, то в мережах IoT всіма учасниками будуть пристрої під'єднані до мережі, і більша частина взаємодії (якщо не вся) буде проходити саме між ними й виділеним оброблювальним сервером. Розберемося які саме пристрої можуть зустрічатися в подібних системах.

Якщо не брати в розрахунок оброблювальний сервер, що може бути як і повноцінним комп'ютером, так і спеціальним мікроконтролером, або навіть мережевим пристроєм, то інші пристрої можна умовно розділити на дві категорії.

Першу категорію можна назвати «сенсори» або «датчики». Вони виконують роль органів чуття нашої мережі, як от очі або шкіра. Це найпростіші пристрої що мають на собі один або декілька сенсорів (іноді декілька різних) що постійно вимірюють той чи інший параметр навколишнього середовища, як от температура, освітлення, рівень вологості, рівень задимлення, тиск, або навіть якісь більш специфічні показники, як от напруга, поверхнєве натягнення, склад розчину. Для роботи таких датчиків не потрібно багато електроенергії, навіть якщо вони працюють цілодобово, і під час роботи вони не створюють великі об'єми даних самі по собі, але от в купі вони створюють велику кількість потокових даних, що оновлюються кожену секунду, для чого і є необхідним окремий обчислювальний пристрій, що буде в змозі обробляти увесь потік різнорідних даних, і вчасно реагувати на зміни показників.

До таких пристроїв слід віднести, датчики руху, температури, освітлення, пристрої для вимірювання показників, датчики відкриття дверей та розбиття вікон, датчики диму, камери відеоспостереження (Рис. 2.9.1).



Рисунок 2.9.1. Приклади різноманітних датчиків

До другої категорії можна віднести різноманітні виконавчі пристрої, такі як різноманітні реле, механізми замикання дверей, перемикачі, пневматичні помпи, поршні, автоматичні клапани та іншого роду пристрої що мають механічне або логічне управління (Рис. 2.9.2). Саме вони мають в автоматичному, або в напівавтоматичному режимі регулювати процеси в приміщенні чи на підприємстві де вони використовуються, тому важливо налагодити стабільний та неперервний зв'язок виконавчих пристроїв з сенсорами чи пристроями керування.



Рисунок 2.9.2 -- Автоматичні пристрої керування

## 2.10. Системи управління мережами інтернет речей

Будь-які системи потребують належного управління, особливо мережі з використанням датчиків та виконавчих пристроїв. В будь-якому випадку мережу IoT можна побудувати повністю автономною, де не потрібне буде втручання людини у процеси що виконуються всередині мережі, але для коректної роботи такої мережі все одно необхідна система моніторингу і контролю процесів що відбуваються, для простежування помилок, непередбачуваних ситуацій, швидкого і своєчасного реагування на випадки всередині мережі. Звісно, якщо мережа будується всередині великого підприємства (а особливо якщо в цьому підприємстві

реалізовані серйозні політики безпеки та конфіденціальності) то вони скоріш за все будуть мати систему управління власного виробництва, незалежну від навколишніх ресурсів, але для малих підприємців, або тих хто просто хоче розробити власну розумну мережу вихід також є.

Кожен шанобливий ІТ гігант вже має свою власну хмарну платформу, яка має функціонал та ресурси для людей і підприємців, що не можуть собі дозволити дороге мережеве устаткування. Натомість за доволі демократичними цінами, людина може налагодити свою систему у хмарі, що буде доступна у кожному куточку земного шару. Але великі цифрові компанії також оцінюють і новітні тенденції розвитку, щоб вчасно вкласти гроші у потенціально прибуткові сфери, тому й інтернет речей не оминув їх уваги. Такі великі компанії як Amazon, Google, Microsoft, Oracle та багато інших вже мають свої хмарні системи управління інтернет речами.

До таких систем відносяться:

- Amazon Web Services;
- Google Cloud IoT;
- Microsoft Azure IoT Suite;
- SAP;
- Salesforce IoT;
- Oracle Internet of Things;
- Cisco IoT Cloud Connect;
- Bosh IoT Suite;
- IBM Watson Internet of Things;
- ThingWorx IoT Platform;

Всі вони пропонують виділений хмарний простір для вашої мережі, обчислювальні ресурси, легку та універсальну систему під'єднання пристроїв до мережі, моніторинг їх стану, формування статистики роботи устаткування, системи управління, а деякі навіть підтримку технологій машинного навчання та нейромереж, для навчання на даних що створюються пристроями, обробки та роботи з цими даними.

### **3 СИСТЕМА ДИСТАНЦІЙНОГО КОНТРОЛЮ ІОТ**

#### **3.1. Програма дистанційного управління**

Для цього дипломного проєкту мною була написана власна програма моніторингу та функціонування розумної мережі. Ця програма є дешевою та не потребує багато мережевих ресурсів. Вона виконує функції моніторингу процесів, також вона може виконувати будь-які команди, що будуть прописані в програмі. Вона створена на мові Python, та використовує безоплатне програмне забезпечення Telegram (месенджер) для доставляння та отримання повідомлень.

Перевагами даної програми можна назвати універсальність (усі засоби що використані можуть бути запуснені на будь-якому сучасному смартфоні або комп'ютері), доступність (все що необхідно доступ в загальний інтернет), простоту у використанні й налаштуванні, невелику ціну та відсутність необхідності у великих ресурсах та специфічному устаткуванню.

До недоліків слід віднести використання стороннього програмного забезпечення, відносно низьку захищеність (інформація проходить хоч і через загальну мережу, але захищена засобами застосунку Telegram), необхідність виділеного оброблювального пристрою в мережі (з підтримкою мови програмування Python), необхідність у постійному доступі до мережі інтернет.

#### **3.2. Склад програми дистанційного управління**

Програма складається з двох скриптів Python.

Перший скрипт (Додаток А), під назвою «BotLib», має в собі клас та методи для Telegram бота. Месенджер Telegram дозволяє створювати всередині свого застосунку сутність, що нічим не відрізняється від звичайного користувача цієї соціальної мережі. За допомогою чату @BotFather, кожен користувач може створити власного бота, якого в майбутньому можна запрограмувати на виконання тих чи інших завдань, пересилку повідомлень, або навіть вести спілкування з

іншими учасниками соцмережі. Спілкування з ним проходить через API, який видається при створенні бота за допомогою HTTP запитів. API Telegram використовує формат JSON, для цього в класі бота прописані методи для спрощення відправлення запитів на сервер месенджеру. В даному скрипті містяться всі основні методи для відправлення текстових повідомлень, фотографій, опитувань, та отримання повідомлення написаного боту, для їх подальшої обробки (Рис. 3.2.1).

```
def forward(self, mess):
    chat_id = mess['message']['chat']['id']
    message_id = mess['message']['message_id']
    method = 'forwardMessage'
    params = {'chat_id': self.chat_id, 'from_chat_id': chat_id, 'message_id': message_id}
    resp = requests.get(self.api_url + method, params)
    return resp

def send_photo(self, chat_id, photo, text):
    files = {"photo": photo}
    params = {'chat_id': chat_id, 'caption': text}
    method = 'sendPhoto'
    resp = requests.post(self.api_url + method, data=params, files=files)
    return resp

def send_sticker(self, chat_id, sticker):
    params = {'chat_id': chat_id, 'sticker': sticker}
    method = 'sendSticker'
    resp = requests.post(self.api_url + method, data=params)
    return resp

def send_poll(self, chat_id, question, answers):
    json_answers = json.dumps(answers)
    params = {'chat_id': chat_id, 'options': json_answers, 'question': question,
              'is_anonymous': False, 'allows_multiple_answers': True}
    method = 'sendPoll'
    resp = requests.post(self.api_url + method, data=params)
    return resp
```

Рисунок 3.2.1 – Методи прописані в класі бота

Другий скрипт (Додаток Б), містить в собі код обробки команд та планування подій. Кожна необхідна команда прописується в цьому скрипті з інструкціями до виконання. Для демонстрації роботи програми в ній були прописані команди «Say hi», «Status» та «Ok» (Рис. 3.2.2). Для планування подій була використана бібліотека Schedule, але також є можливість прописати такий функціонал

самостійно або за допомогою додаткових бібліотек для роботи з часом. Для демонстрації такої можливості була прописана функція що в необхідний час постить повідомлення «Hello World!!!» (Рис 3.2.3). При отриманні невідомої команди бот відповідає «~Unknown Command~».

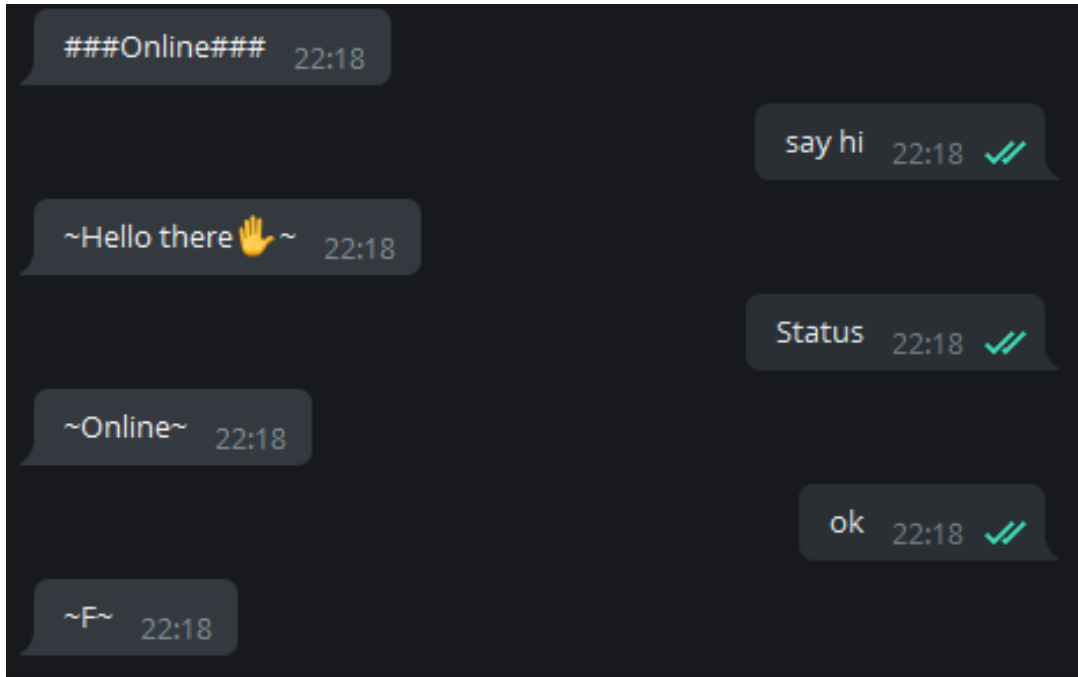


Рисунок 3.2.2 – Відповіді на команди від користувача

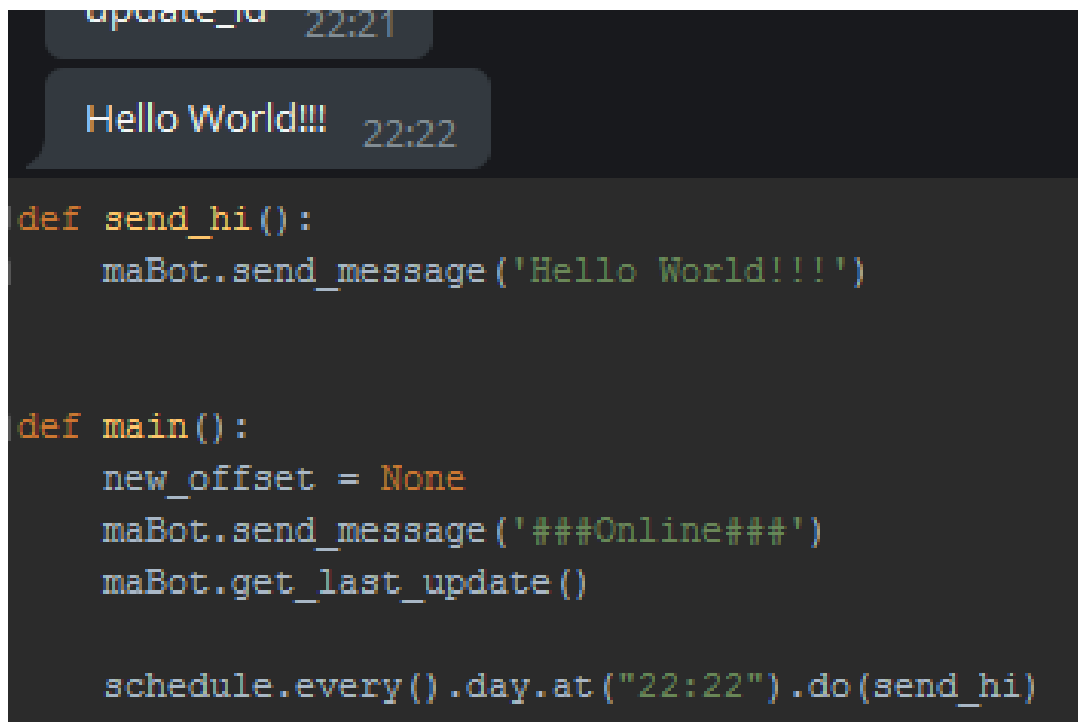


Рисунок 3.2.3 – Модуль планування виконання задачі

### 3.3. Захист програми дистанційного управління

Даний програмний застосунок для передачі інформації через відкриту мережу використовує засоби захисту месенджеру «Telegram». Ця програма обміну повідомлень займає третє місце в рейтингу найбільш захищених месенджерів за версією аналітичного відділу Artezio (Артезио) за 2018 рік, але з тих пір захист програми був значно покращений.

Месенджер «Telegram» з серпня 2019 року працює на власному протоколі MTProto. Через проблеми доступності до даного застосунку в деяких країнах (зокрема Іран і Китай блокували трафік що передавався даним протоколом, відокремлюючи його від іншого по довжині пакету) командою розробників месенджеру було додано додатковий шар «маскування» пакетів протоколу MTProto. Технологія маскування полягає в тому що протоколи загортаються в пакети HTTPS (TLS + HTTP/2.0), з додаванням випадкового байту. Повідомлення що зберігаються та передаються месенджером шифруються за допомогою base16 (hex).

Зі сторони коду, для коректної роботи цього бота необхідно прописати отриманий при створенні токен (ідентифікатор бота), та, для більшої надійності, в ньому також прописується власний ідентифікатор Telegram (Рис. 3.3.1). Якщо користувач в бота всього один, цього достатньо для того, щоб повідомлення що йдуть від бота не могли бути підслухані.

```
def __init__(self):
    self.token = '#####'
    self.api_url = "https://api.telegram.org/bot#####/"
    self.chat_id = '40#####'
    self.last_update_id = 0
```

Рисунок 3.3.1 -- Функція init класу бота

Так як в середині месенджеру бота можливо знайти по імені, хоча і дуже складно, наприклад методом підбору, при отриманні повідомлення від

неавторизованого користувача бот відповідає «~Undefined user~» і пересилає отримане повідомлення власнику бота (Рис. 3.3.2). Всі повідомлення що посилаються боту і надходять від нього перевіряються на автентичність і надходять тільки авторизованим користувачам, прописаним в коді самого бота.



Рисунок 3.3.2 – Захист від неавторизованих користувачів

#### **3.4. Використання програми дистанційного управління**

Для дистанційного управління і моніторингу мережі IoT достатньо застосувати дане програмне забезпечення на оброблювальному пристрої мережі, ПК чи мікроконтролері, і написати низку інструкцій яку і коли інформацію необхідно сповіщати власнику, які дії виконуються при тих чи інших показниках, а також які дії слід виконати при отриманні виконавчої команди. Також функціонал програми дозволяє налаштовувати постійні події що виконуються у визначений час. В програму закладена можливість її подальшої модернізації та розширення функціоналу.

## ВИСНОВОК

Проаналізувавши весь викладений в даній роботі матеріал можна дійти висновку що розумні мережі з використанням технологій IoT оточують нас маже всюди. Ця сфера є досить перспективною і популярною на ринку праці, бо кожна поважна компанія, при можливості намагається автоматизувати усі наявні процеси на підприємстві. Світ вже наповнений всілякими системами що покращують і підвищують ефективність процесів взаємодії жителів міста, людей та пристроїв, і така тенденція буде досить потужною найближчим часом, так, як можливість покращити стан речей надихає кожного до впровадження новітніх технологій у повсякденне життя.

Виконуючи дану роботу я провів дослідження різних інформаційних систем із використанням пристроїв IoT, дослідив технології що були впроваджені в даній сфері, а також ті що тільки розроблюються, або впроваджуються в наш час.

Перший розділ даної роботи містить комплексні описи вже наявних мереж з використанням IoT. В ньому містяться інформація про те які саме технології необхідні людині сьогодення і як вони впроваджуються в різних кутках земного шару.

В другому розділі я провів якісний опис стандартів, технологій, протоколів та пристроїв що існують і використовуються в мережах IoT, також було описано їх переваги та недоліки, проведено аналіз специфіки використання тих чи інших пристроїв в загальних та спеціальних умовах.

Результатом даної дипломної роботи була розробка, реалізація та впровадження програмного застосунку для дистанційного контролю та управління мережею з використанням розумних датчиків та виконавчих пристроїв, що була описана в третьому розділі даної дипломної роботи, її подальший опис і роз'яснення щодо переваг та недоліків використання саме цього програмного застосунку, та методології роботи цього програмного забезпечення і відповідності його до заявлених вимог.

Провівши аналіз технологій інтернету речей, та розробивши власний застосунок для керування і моніторингу розумних мереж я дізнався які саме технології поширені в наш час в цій сфері, як саме підходять до розробки аналогічних систем великі підприємства та інші зацікавлені в цьому люди, надихнувся вже створеними системами автоматичного управління чи моніторингу, про які можна дізнатися у вільному доступі, проаналізував запропоновані готові продукти з елементами IoT.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Історія між машинної взаємодії: <https://www.affirmednetworks.com/m2m-rise-of-the-machines/>
- 2 Принципи між машинної взаємодії: <https://internetofthingsagenda.techtarget.com/definition/machine-to-machine-M2M>
- 3 Документація Telegram Bot API: <https://core.telegram.org/bots/api>
- 4 Н. Борисова, Автоматична телефонна комутація: легенди, версії, факти: Випуск № 8, 2017 рік, <https://www.lastmile.su/journal/article/6448>
- 5 Історія виникнення інтернету речей: <https://www.itransition.com/blog/iot-history>
- 6 Огляд розумних міст світу: [https://www.tadviser.ru/index.php/\\_\(Smart\\_cities\)](https://www.tadviser.ru/index.php/_(Smart_cities))
- 7 Історія виникнення інтернету: <https://www.youtube.com/watch?v=wTMFQJ5ckT8>
- 8 Огляд ризиків систем «Розумний дім»: <https://tech-house.su/kakie-riski-mogut-voznikat-u-vladelcev-umnogo-doma/>
- 9 Документація протоколів TCP/IP: <https://www.ibm.com/docs/ru/aix/7.2?topic=management-transmission-control-protocolinternet-protocol>
- 10 Протоколи передачі даних IoT: <https://iot.ru/wiki/protokoly-peredachi-dannykh-iot>
- 11 Ознайомлення з протоколом MQTT-SN: <http://www.steves-internet-guide.com/mqtt-sn/>
- 12 Огляд IoT платформ: <https://www.edsson.com/ru/blog/article?id=iot-platforms>
- 13 Інформаційний бюлетень Microsoft Azure: <https://azure.microsoft.com/ru-ru/overview/internet-of-things-iot/iot-technology-protocols/>
- 14 Історія Пола Берна, винахідника мереж з комутацією пакетів: <https://habr.com/ru/post/404167/>

## ДОДАТКИ

## Додаток А

## #BotLib

```

import requests
import json

class BotHandler:

    def __init__(self):
        self.token = '#####'
        self.api_url =
"https://api.telegram.org/bot#####/"
        self.chat_id = '#####'
        self.last_update_id = 0

    def get_updates(self, offset=None, timeout=30):
        method = 'getUpdates'
        if self.last_update_id > 0:
            params = {'timeout': timeout, 'offset': self.last_update_id}
            resp = requests.get(self.api_url + method, params)
            result_json = resp.json()['result']
        else:
            params = {'timeout': timeout, 'offset': offset}
            resp = requests.get(self.api_url + method, params)
            result_json = resp.json()['result']

        return result_json

    def send_message(self, text):
        params = {'chat_id': self.chat_id, 'text': text}
        method = 'sendMessage'
        resp = requests.post(self.api_url + method, params)
        return resp

    def send_common_message(self, chat_id, text):
        params = {'chat_id': chat_id, 'text': text}
        method = 'sendMessage'
        resp = requests.post(self.api_url + method, params)
        return resp

    def send_keyboard(self, chat_id, text, keyboard):
        reply_markup = {"keyboard": keyboard,
                        "one_time_keyboard": True, 'resize_keyboard': True}
        params = {'chat_id': chat_id, 'text': text, 'reply_markup':
json.dumps(reply_markup)}
        method = 'sendMessage'
        resp = requests.post(self.api_url + method, params)
        return resp

    def send_post(self, channel_id, text):
        params = {'chat_id': channel_id, 'text': text, "parse_mode": "Markdown"}
        method = 'sendMessage'
        resp = requests.post(self.api_url + method, params)
        return resp

    def get_last_update(self):

```

```

get_result = self.get_updates()
new_updates = []
last_update = get_result[-1]
if last_update['update_id'] > self.last_update_id:
    for update in get_result:
        if update['update_id'] > self.last_update_id:
            new_updates.append(update)
self.last_update_id = last_update['update_id']
return new_updates

def forward(self, mess):
    chat_id = mess['message']['chat']['id']
    message_id = mess['message']['message_id']
    method = 'forwardMessage'
    params = {'chat_id': self.chat_id, 'from_chat_id': chat_id, 'message_id':
message_id}
    resp = requests.get(self.api_url + method, params)
    return resp

def send_photo(self, chat_id, photo, text):
    files = {"photo": photo}
    params = {'chat_id': chat_id, 'caption': text}
    method = 'sendPhoto'
    resp = requests.post(self.api_url + method, data=params, files=files)
    return resp

def send_sticker(self, chat_id, sticker):
    params = {'chat_id': chat_id, 'sticker': sticker}
    method = 'sendSticker'
    resp = requests.post(self.api_url + method, data=params)
    return resp

def send_poll(self, chat_id, question, answers):
    json_answers = json.dumps(answers)
    params = {'chat_id': chat_id, 'options': json_answers, 'question':
question,
            'is_anonymous': False, 'allows_multiple_answers': True}
    method = 'sendPoll'
    resp = requests.post(self.api_url + method, data=params)
    return resp

```

## #Handler

```

import time
import schedule
import BotLib

maBot = BotLib.BotHandler()
me = #####

def send_hi():
    maBot.send_message('Hello World!!!')

def main():
    new_offset = None
    maBot.send_message('###Online###')
    maBot.get_last_update()

    schedule.every().day.at("22:22").do(send_hi)

    while True:
        try:
            maBot.get_updates(new_offset)
            last_updates = maBot.get_last_update()
        except ConnectionError:
            continue

        #####
        schedule.run_pending()
        #####

        for last_update in last_updates:
            user_name = last_update['message']['chat']['username']
            user_id = last_update['message']['from']['id']

            try:
                message_text = last_update['message'].get('text')
            except KeyError:
                message_text = None

            if message_text is not None:
                if user_id != me:
                    maBot.send_message(
                        '~Undefined user~' + '\n' +
last_update['message']['chat']['first_name'] + ' => '
                        + str(user_id))
                    maBot.send_message(message_text)
                elif message_text.lower() == 'status':
                    maBot.send_message('~Online~')
                elif message_text.lower() == 'say hi':
                    maBot.send_message('~Hello there👋~')
                elif message_text.lower() == 'ok' or message_text.lower() == 'ok!':
                    maBot.send_message('~F~')
                elif user_id == me:
                    maBot.send_message('~Unknown command~')
                else:
                    maBot.send_message('~Undefined command~')

```

```
        elif message_text is None:
            maBot.forward(last_update)
            maBot.send_message('~Undefined user~' + '\n' +
last_update['message']['chat']['first_name'])
            maBot.send_message(user_name)

            time.sleep(1)

if __name__ == '__main__':
    try:
        main()
    except KeyboardInterrupt:
        exit()
```