МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА СИСТЕМ ІНФОРМАЦІЙНОГО ТА КІБЕРНЕТИЧНОГО ЗАХИСТУ

«На правах рукопису»
УДК 681.3.06

«До захисту допущено»
Завідуючий кафедрою СІКЗ
_____к.т.н. Г.В. Шуклін
«____» _____2023 р.

БАКАЛАВРСЬКА АТЕСТАЦІЙНА РОБОТА

зі спеціальності 125 “Кібербезпека”

на тему: **ЗАХИСТ ІНФОРМАЦІЇ В БЕЗПРОВОДОВІЙ МЕРЕЖІ IoT**

Студент групи СЗД-41 Разваляєв Володимир Анатолійович _____
(підпис)

Науковий керівник: д.т.н**.** Крючкова Лариса Петрівна _____
(підпис)

Нормоконтроль Зозуля Сергій Анатолійович _____
(підпис)

КИЇВ – 2023

«ЗАТВЕРДЖУЮ»
Завідувач кафедри СІКЗ
_____ к.т.н. Г.В. Шуклін
(підпис)
«_____»_____ 2023р.

ЗАВДАННЯ

на атестаційну роботу бакалавра

студенту: Разваляєву Володимиру Анатолійовичу

1.Тема роботи: Захист інформації в безпроводовій мережі IoT від « » 2023р. №

2.Термін здачі студентом оформленої роботи «____» _____ 2023р.

3. Об'єкт дослідження: методи бездротової передачі даних, які використовуються в мережі IoT та їх безпека.

4. Предметом дослідження: протоколи бездротової передачі даних, захист даних у протоколах бездротової передачі даних.

5. Мета роботи: дослідження методів захисту даних у бездротових мережах IoT.

6. Перелік питань, які мають бути розроблені:

Для досягнення вказаної мети виконуються такі основні задачі:

- Дослідження архітектури інтернету речей.
- Дослідження протоколів бездротової передачі даних та їх безпеки.
- Практичне дослідження побудови інтернету речей.

7. Перелік публікацій

8. Перелік ілюстрованого матеріалу

Презентація матеріалу на слайдах.

9. Дата видачі завдання «____» _____20____р.

Науковий керівник _____ Крючкова Л.П.

Завдання прийняв до виконання _____ Разваляєв В.А.

КАЛЕНДАРНИЙ ПЛАН

Дата видачі завдання «____» _____2023р.

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	до 21.02.23р.	
2	Обґрунтування актуальності теми роботи	до 28.02.23р.	
3	Написання першого розділу роботи	до 16.03.23р.	
4	Написання другого розділу роботи	до 12.04.23р.	
5	Написання третього розділу роботи	до 10.05.23р.	
6	Написання висновків по роботі	до 12.05.23р.	
8	Підготовка демонстраційних матеріалів	до 18.05.23р.	
9	Підготовка доповіді	до 26.05.23р.	
10	Захист в ДЕК		

Студент: СЗД-41 Разваляєв В.А. _____
(підпис)

Науковий керівник: д.т.н. Крючкова Лариса Петрівна _____
(підпис)

Нормоконтроль: Зозуля С.А. _____
(підпис)

ЗМІСТ

РЕФЕРАТ

Дипломна робота містить 70 сторінок, 22 рисунки.

Бездротова локальна мережа Internet of Things (IoT) є однією з найважливіших та швидкозростаючих галузей сучасної технології. Її значення полягає у забезпеченні безперервного зв'язку та обміну даними між фізичними пристроями, які можуть включати датчики, актуатори, розумні пристрої та системи контролю, для забезпечення автоматизації, моніторингу та управління різними аспектами нашого повсякденного життя.

Об’єктом дослідження: методи бездротової передачі даних, які використовуються в мережі IoT та їх безпека.

Предметом дослідження є протоколи бездротової передачі даних, захист даних у протоколах бездротової передачі даних.

Мета роботи дослідження методів захисту даних у бездротових мережах IoT.

Для досягнення вказаної мети виконуються такі основні задачі:

- Дослідження архітектури інтернету речей.
- Дослідження протоколів бездротової передачі даних та їх безпеки.
- Практичне дослідження побудови інтернету речей.

Новизна отриманих результатів полягає в наступному:

1. Розглянуті основні технології на яких базуються бездротові локальні мережі IoT.

2. Розроблені віртуальні мережі IoT.

Галузь використання — бездротові мережі.

ІНФОРМАЦІЯ, ІНФОРМАЦІЙНА СИСТЕМА, ЗАХИСТ ІНФОРМАЦІЇ, ІНТЕРНЕТ РЕЧЕЙ, БЕЗДРОТОВА ПЕРЕДАЧА ДАНИХ, ЛОКАЛЬНА МЕРЕЖА.

ABSTRACT

Thesis contains 70 pages, 22 figures.

A wireless local network of Internet of Things (IoT) is one of the most important and rapidly growing fields in modern technology. Its significance lies in providing continuous communication and data exchange among physical devices, which can include sensors, actuators, smart devices, and control systems, to enable automation, monitoring, and management of various aspects of our everyday lives.

Object of research: wireless data transfer methods used in the IoT network and their security.

The subject wireless data transmission protocols, data protection in wireless data transmission protocols.

The purpose of the work is to research data protection methods in IoT wireless networks.

To achieve this goal, the following main tasks are performed:

- Study of the architecture of the Internet of Things.
- Study of wireless data transmission protocols and their security.
- Practical research on building the Internet of Things.

The novelty of the results is as follows:

1. The main technologies underlying wireless local IoT networks have been considered.

2. Virtual IoT networks have been developed.

Field of use - information security.

INFORMATION, INFORMATION SYSTEM, INTERNET OF THINGS, INFORMATION SECURITY, WIRELESS DATA TRANSMISSION, LOCAL NETWORK.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AES (Advanced Encryption Standard) – розширений стандарт шифрування.

AMP (Alternative MAC/PHY) – альтернативний MAC/PHY.

AMQP (Advanced Message Queuing Protocol) – протокол передачі розширених повідомлень черги.

AODV (Ad hoc On-Demand Distance Vector) – Спеціальний вектор відстані на вимогу.

AP (Access Point) – точка доступу.

BR/EDR (Bluetooth Basic Rate / Enhanced Data Rate) – базова швидкість Bluetooth/розширена швидкість передачі даних.

CSMA/CA (Carrier-Sense Multiple Access with Collision Avoidance) – Множинний доступ із визначенням несучої з уникненням зіткнень.

CoAP (Constrained Application Protocol) – протокол обмеженої програмної аплікації

DNS (Domain Name System) – система доменних імен.

DoS (Denial of Service) – відмова сервісу.

FDDI (Fiber Distributed Data Interface) – розподілений волоконний інтерфейс даних.

FHSS (Frequency-Hopping Spread Spectrum) – псевдовипадкова перебудова робочої частоти.

HTTP (HyperText Transfer Protocol) – протокол передачі гіпертексту.

IEEE (Institute of Electrical and Electronics Engineers) – Інститут інженерів електротехніки та електроніки.

IIoT (Industrial Internet of Things) – промисловий інтернет речей.

IP (Internet Protocol) – інтернет протокол.

IPSec (Internet Protocol Security) – безпека Інтернет-протоколу.

ISM (Industrial, Scientific and Medical) – промислові, наукові та медичні.

IoT (Internet of Things) – інтернет речей.

LE/BLE (Low Energy / Bluetooth Low Energy) – низьке енергоспоживання / Bluetooth з низьким енергоспоживанням.

LPN (Low Power Node) – вузол з ниьким енергоспоживанням.

MAC (Media Access Control) – нагляд за доступом до середовища.

MLE (Mesh Link Establishment) – встановлення mesh з'єднання.

MQTT (Message Queuing Telemetry Transport) – транспортування повідомлень із чергами для телеметрії.

OEM (Original Equipment Manufacturer) – виробник оригінального обладнання.

PDU (Protocol Data Unit) – блок даних протоколу.

PHY (Physical layer) – фізичний рівень.

PIN (Personal Identification Number) – персональний ідентифікаційний номер.

QoS (Quality of Service) – якість обслуговування.

REED (Router-Eligible End Devices) – Придатні для маршрутизатора кінцеві пристрої.

TCP (Transmission Control Protocol) – протокол управління передачею.

UDP (User Datagram Protocol) – протокол користувальницьких датаграм.

UUID (Universally Unique Identifier) – всесвітньо унікальний ідентифікатор

WAN (Wide Area Network) – глобальна мережа.

WEP (Wired Equivalent Privacy) – конфіденційність, еквівалентна проводовій мережі.

WLAN (Wireless Local Area Network) – бездротова локальна мережа.

WPA (Wi-Fi Protected Access) – Захищений доступ Wi-Fi

ZC (Zigbee Controller) – контролер Zigbee.

ZED (Zigbee End Device) – кінцеве пристрій Zigbee.

ZR (Zigbee Router) – маршрутизатор Zigbee.

Модель OSI (Open Systems Interconnection) – модель взаємодії відкритих систем.

ВСТУП

З ростом бездротових технологій Інтернету речей (IoT) наш світ стає все більш підключеним і залученим до цифрового простору. Величезна кількість пристроїв, що з'єднуються в мережу, надає нам неймовірні можливості для комунікації, автоматизації та збору даних. Однак, разом з цими перевагами зростають і загрози, пов'язані з безпекою і захистом інформації.

В бездротових мережах IoT з'являються нові проблеми безпеки, оскільки вони мають свої особливості порівняно з традиційними комп'ютерними мережами. Аутентифікація та захист від несанкціонованого доступу стають складними завданнями у зв'язку з величезною кількістю пристроїв, що з'єднуються, різноманітністю їх функцій і обмеженими ресурсами.

Однією з найважливіших аспектів безпеки в бездротових мережах IoT є захист інформації. У таких мережах передаються і зберігаються важливі дані, які можуть містити конфіденційну інформацію про користувачів, підприємства та навколишнє середовище. Зловмисники можуть намагатися перехопити ці дані, використовувати їх для своїх цілей або навіть викликати функціональні збої в системі.

РОЗДІЛ 1 АРХІТЕКТУРА МЕРЕЖЕЙ IOT

1.1 Опис інтернету речей

Концепція Інтернету речей (Internet of Things, IoT) полягає у створенні мережі передачі даних між фізичними об'єктами, так званими "речами", які мають вбудовані засоби та технології для взаємодії між собою та з зовнішнім середовищем. Ця концепція передбачає можливість організації таких мереж, що здатні перебудувати економічні та суспільні процеси, а також уникнути необхідності участі людини в деяких діях та операціях.

Термін «інтернет речей» завдячує своєю появою Кевіну Ештону, який у 1997 році, працюючи на компанію Protocol and Gamble, для управління системою поставок застосував технологію радіочастотної ідентифікації. Завдяки цій роботі в 1999 році його запросили до Массачусетського технологічного інституту, де він з групою однодумців організував дослідницький консорціум Auto-ID Center. З того часу інтернет речей здійснив перехід від простих радіочастотних міток до величезної індустрії [1].

Пристрої Інтернету речей відрізняються від багатьох наявних IT-пристроїв тим, що вони переважно пов'язані з фізичною дією або подією. Це проявляється у наявності в мережі IoT таких пристроїв, як датчики та актуатори. Вони можуть використовуватись для отримання даних з навколишнього середовища або впливу на нього.

Однією з найбільш значущих тенденцій Інтернету речей в останні роки є вибуховий ріст пристроїв, які підключені та контролюються через інтернет. Широкий спектр застосувань технології Інтернету речей означає, що особливості можуть сильно відрізнятися від одного пристрою до іншого, але є основні характеристики, спільні для більшості.

Інтернет речей створює можливості для більш прямої інтеграції фізичного світу в комп'ютерні системи, що призводить до підвищення ефективності, економічних вигод та зменшення навантаження на людину [2].

1.2 Архітектура IoT

Архітектура Інтернету речей охоплює багато технологій. Складність та багатогранність IoT пов'язані з тим, що ця технологія є набагато більш комплексною, ніж традиційні технології: її відрізняє не тільки великий масштаб, але й поєднання різних, не пов'язаних між собою типів архітектури. Кількість можливих проектних рішень величезна. Безвідповідальний вибір протоколу передачі даних може призвести до проблем з обміном даними та помітно низької якості сигналу. Також потрібно враховувати перешкоди в локальних та глобальних мережах, потрібно подумати і про стійкість системи до збоїв.

Архітектура Інтернету речей починається з простих датчиків, розташованих у найвіддаленіших куточках світу, і перетворює аналоговий фізичний вплив на цифрові сигнали. Потім дані подорожують складним шляхом через дротові та бездротові сигнали, різні протоколи, природні перешкоди та накладання електромагнітних полів, і в результаті потрапляють в Інтернет. Звідти пакети даних передаються по різних каналах до хмари або до центру обробки даних, після чого надсилаються користувачеві.

Архітектура IoT пристроїв складається з чотирьох рівнів: рівень датчиків і актуаторів, мережевий рівень, рівень обробки даних та прикладний рівень (рис. 1.1.). Більш детальне описання кожного рівню архітектури наведено нижче [3].

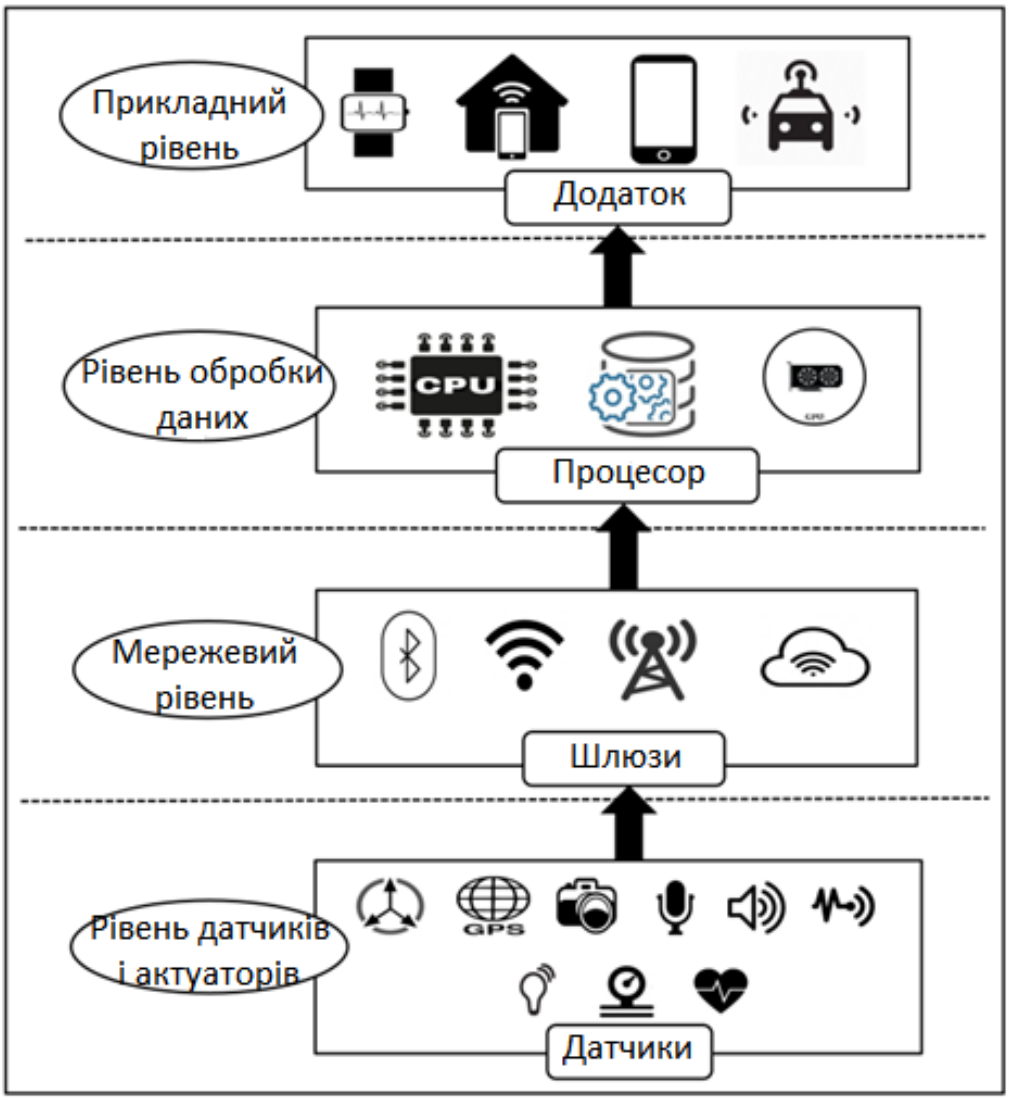

Рис. 1.1 – Рівні IoT

На рівні датчиків та актуаторів відбувається взаємодія з навколишнім середовищем. Датчики збирають інформацію, а актуатори виконують дії, задані керуючим сигналом.

Датчики – це пристрої, призначені для зчитування інформації з навколишнього середовища. Вони можуть зчитувати швидкість вітру, вологість, електропровідність та інше. Існує велика кількість різноманітних датчиків, які використовуються повсюдно, серед них: акселерометр, ультразвуковий сенсор, датчик вологості, датчик температури, гіроскоп, інфрачервоний дальномір. На сьогоднішній день можливості електронних сенсорів не тільки досягли рівня людського сприйняття, але і в багатьох випадках перевершили його. Наприклад,

людина без додаткового обладнання не здатна визначити рівень атмосферного тиску, рівень радіації, наявність у повітрі небезпечних домішок.

Актуатори, які також називають виконавчими елементами, є пристроями, які перетворюють один вид енергії в інший, що призводить до виконання певної дії, заданої управляючим сигналом. Найчастіше актуатори перетворюють електричну енергію в механічну. До актуаторів можна віднести такі пристрої, як: вібромотор, кроковий двигун, пневматичний насос, будь-який електромеханічний пристрій. У теорії автоматичного керування під виконавчим пристроєм розуміють пристрій, який передає вплив від керуючого пристрою на об'єкт керування. Іноді розглядається як складова частина об'єкта керування.

Мережевий рівень використовується як комунікаційний канал для передачі даних від датчиків і до актуаторів. У мережах Інтернету речей найчастіше використовуються бездротові технології для з'єднання пристроїв, хоча може застосовуватися й дротове з'єднання. У мережевому рівні IoT важливо забезпечити якість передачі даних, зменшити затримки і втрати даних, та забезпечити захист і конфіденційність даних. Для досягнення цих цілей, використовуються різноманітні протоколи передачі даних, такі як Wi-Fi, Bluetooth, Zigbee та інші. Крім того, використовуються спеціальні системи керування мережевим трафіком, такі як MQTT, CoAP, AMQP та інші.

Рівень обробки даних відповідає за збір та обробку даних, що надходять від датчиків та інших підключених до мережі IoT пристроїв. Одним з головних завдань рівня обробки даних є визначення алгоритмів та методів обробки даних, які дозволяють здійснювати різні операції зі зібраними даними, включаючи фільтрацію, агрегацію та аналіз. На цьому рівні також можуть бути реалізовані різноманітні технології для оптимізації роботи мережі IoT, наприклад, кешування даних, компресія даних, розподілена обробка даних тощо.

До складу рівня обробки даних можуть входити різні компоненти, такі як мікроконтролери, процесори та інші електронні компоненти. В залежності від конкретного застосування, рівень обробки даних може бути реалізований як на пристроях IoT, так і на хмарних платформах.

Основні виклики для рівня обробки даних IoT полягають у забезпеченні високої продуктивності та ефективності обробки даних, а також в забезпеченні безпеки та конфіденційності даних, що обробляються на цьому рівні. Для цього можуть використовуватися різноманітні методи та технології, включаючи шифрування даних, використання блокчейну та інші заходи безпеки.

Прикладний рівень IoT пристроїв є рівнем архітектури Інтернету речей, орієнтованим на користувачів. Він дозволяє користувачам отримувати результати обробки даних, які були зібрані на рівні обробки даних. Цей рівень є основою для розробки різноманітних додатків, які забезпечують більш зручний та простий спосіб взаємодії користувачів з IoT пристроями.

Додатки IoT пристроїв можуть виконувати різні завдання для користувачів, включаючи моніторинг та управління різними пристроями, віддалену діагностику, навігацію та інші функції. Наприклад, за допомогою додатків розумного будинку користувач може контролювати температуру, освітлення, безпеку та інші параметри в будинку. А додатки для розумного транспорту можуть допомогти водієві планувати маршрут, контролювати витрати палива та підтримувати безпеку на дорозі.

Рівень додатків є необхідним для того, щоб IoT пристрої могли бути доступними та зрозумілими для звичайних користувачів. Він дозволяє взаємодіяти з IoT пристроями за допомогою звичайного смартфона або комп'ютера, що дуже зручно та просто для більшості людей. Без рівня додатків Інтернет речей не міг би стати настільки популярним та використовуваним у повсякденному житті.

1.3 Вимоги до IoT

1.3.1 Основні характеристики

IoT має такі основні характеристики [4]:

- *Можливість встановлення з'єднань*: що стосується IoT, будь-яку річ можна підключити до глобальної інформаційно-комунікаційної інфраструктури

- *Послуги, пов'язані з речами*: IoT може надавати послуги, пов'язані з речами, з урахуванням характерних обмежень, таких як захист приватності та семантична відповідність між фізичними речами та відповідними їм віртуальними речами. Для надання послуг, пов'язаних з речами, з урахуванням характерних обмежень, зміняться технології фізичного та інформаційного світу.

- *Гетерогенність*: в мережі інтернету речей пристрої є гетерогенними та базуються на різних апаратних платформах та мережах. Вони можуть взаємодіяти з іншими пристроями або платформами послуг через різні мережі.

- *Динамічні зміни*: динамічні зміни характерні для стану пристроїв, наприклад, режими сну та прокидання, підключене та / або відключене стану, а також контексту пристроїв, включаючи місцезнаходження та швидкість. Крім того, кількість пристроїв може динамічно змінюватись.

- *Великий масштаб*: кількість пристроїв, якими потрібно керувати, та які обмінюються даними один з одним, як мінімум на порядок перевершить кількість пристроїв, підключених до Інтернету на даний момент. Значно збільшиться частка інформаційного обміну, ініційованого пристроями, порівняно з часткою інформаційного обміну, ініційованого людьми. Значення управління створюваними даними та їх інтерпретації в прикладних цілях підвищиться. Це стосується семантики даних, а також їх ефективної обробки.

1.3.2 Вимоги високого рівня

Нижче наведені вимоги високого рівня, які є актуальними для IoT:

- *З'єднання на основі ідентифікації*. Треба забезпечити, щоб у IoT з'єднання між різними речами та IoT встановлювалося на основі ідентифікатора кожної речі. Крім того, це вимагає забезпечення однорідного підходу до обробки, гетерогенних ідентифікаторів різних речей.

- *Функціональна сумісність*: потрібно забезпечити функціональну сумісність між гетерогенними та розподіленими системами для надання та споживання найрізноманітніших видів інформації та послуг.

- *Організація автономних мереж*: потрібно, щоб функції IoT, пов'язані з управлінням організацією мереж, підтримували організацію автономних мереж (включаючи методи та/або механізми автоматичного управління, автоматичного конфігурування, самовідновлення, автоматичної оптимізації та автоматичного захисту) для адаптації до різних прикладних областей, різних серед передачі даних та великої кількості пристроїв найрізноманітніших типів.

- *Надання автономних послуг*: потрібно, щоб послуги надавалися за допомогою автоматичного збору, передачі та обробки даних речей на основі правил, встановлених операторами або налаштовуємими абонентами. Автономні послуги можуть залежати від методів автоматизованої спільної обробки та інтелектуального аналізу даних.

- *Можливості, що базуються на визначенні місцезнаходження*: у IoT потрібно забезпечити можливості, що базуються на визначенні місцезнаходження. Зв'язок та послуги, що стосуються якої-небудь речі, будуть залежати від інформації про місцезнаходження речей та/або користувачів. Необхідно, щоб інформація про місцезнаходження вимірювалась та відстежувалась автоматично. Зв'язок та послуги, що базуються на визначенні місцезнаходження, можуть бути обмежені законами та нормативними актами та повинні відповідати вимогам безпеки.

- *Безпека*: в Інтернеті речей кожна річ має з'єднання, що призводить до серйозних загроз безпеці, таких як загрози конфіденційності, автентичності та цілісності як даних, так і послуг. Одним з найважливіших прикладів вимог безпеки в Інтернеті речей є необхідність поєднання різних принципів та методів забезпечення безпеки, що стосуються багатьох пристроїв та мереж користувачів.

- *Захист приватності*: в IoT необхідно забезпечувати захист приватності. Багато речей мають власників та користувачів. Дані вимірювань речей можуть містити особисту інформацію про їх власників або користувачів. У IoT необхідно

забезпечувати захист приватності під час передачі, зберігання, інтелектуального аналізу та обробки даних. Захист приватності не повинен перешкоджати аутентифікації джерела даних.

- *Високоякісні та високозахищені послуги, пов'язані з організмом людини*: в IoT вимагається підтримувати високоякісні та високозахищені послуги, пов'язані з організмом людини. У різних країнах існують різні законодавчі та нормативні акти, що стосуються цих послуг. ЗАУВАЖЕННЯ. - Під послугами, пов'язаними з організмом людини, розуміються послуги, які надаються за допомогою збору, передачі та обробки даних, пов'язаних зі статичними властивостями та динамічними характеристиками організму людини, при цьому здійснюється або не здійснюється втручання людини.
- *Автоматична конфігурація*: необхідно забезпечити можливість автоматичної конфігурації в IoT, що дозволить оперативно створювати, формувати або отримувати конфігурації, основані на семантиці, з метою незатримного інтегрування та взаємодії підключених речей з додатками, а також задовольнення вимог додатків.
- *Управління*: в IoT потрібно забезпечувати управління, щоб забезпечити нормальне функціонування мережі. Зазвичай, додатки IoT працюють в автоматичному режимі, без участі людей, проте весь процес їх роботи повинен підлягати управлінню відповідними сторонами.

1.4 Використання інтернету речей

1.4.1 Промисловий інтернет речей

Сучасний промисловий світ все більше переходить до промислового інтернету речей (IIoT) з можливістю дистанційного контролю ресурсів підприємства та їх автоматизованого управління. За допомогою систем інтернету речей можна отримувати інформацію про доступність обладнання, його технічний

стан, навантаження, технологічні порушення, графік технічного обслуговування тощо. Промисловий інтернет речей дозволяє оперативно, у режимі реального часу отримувати інформацію про всі обладнання на підприємстві, за секунди розрахувати його коефіцієнт корисного використання, а застосовуючи передбачувальну аналітику та нейронні мережі - передбачити графік планово-попереджувальних ремонтів та навантаження [5].

Застосування Інтернету речей в промисловості створює нові можливості для розвитку виробництва та вирішує ряд важливих завдань: підвищення продуктивності обладнання; зниження матеріальних та енергетичних витрат; підвищення якості, оптимізація та поліпшення умов праці співробітників компанії; зростання рентабельності виробництва та конкурентоспроможності на ринку.

Характеристиками цього сегменту є необхідність надавати готові рішення системі в режимі реального часу або майже в режимі реального часу. Це означає, що для Інтернету речей головним параметром в усьому, що стосується виробничого цеху, буде час відгуку. Крім того, важливою роллю будуть відігравати тривалість простою та безпека. Це передбачає потребу в запасі потужностей та, ймовірно, наявності приватних хмарних мереж та сховищ даних. Промисловий Інтернет речей - це один з найбільш швидко розвиваючихся сегментів на ринку.

Приклади застосування промислового інтернету речей:

- проведення профілактичного обслуговування нового та використовуваного раніше промислового обладнання;
- системи безпеки, такі як вимірювання температури, вимірювання тиску та контроль за витоком газу;
- експертна система для виробничого цеху.

1.4.2 Споживач

Споживчі пристрої були однією з перших категорій предметів, що підключаються до Інтернету. Споживчий інтернет речей розпочався із підключеної

до інтернету кавоварки в одному університеті у 1990-х роках. Він розквітнув з поширенням технології Bluetooth на початку 2000-х років. Тепер мільйони будинків оснащені розумними термостатами, лампочками, віртуальними голосовими помічниками та телевізійними приставками. Споживчий ринок зазвичай першим переймає нові технології. Всі ці пристрої поставляються в акуратній упаковці, і, в основному, всі вони діють за принципом «встав та увімкни» [6].

Одна із складностей споживчого сегменту полягає у біфуркації стандартів. Наприклад, ми бачимо, що в основі деяких протоколів бездротової персональної мережі лежать стандарти Bluetooth, Zigbee та Z-wave (які не є інтероперабельними).

Цей напрямок також має дуже багато спільного з медичним сегментом, куди відносяться спеціалізовані портативні пристрої та домашні системи спостереження за здоров'ям.

Приклади застосування споживчого інтернету речей:

- розумні пристрої для дому: система поливу, гаражні двері, замки, ліхтарі, термостати та система охорони;
- портативні пристрої: трекери здоров'я та руху, розумний одяг/аксесуари;
- тварини: системи відстеження місцезнаходження домашніх тварин, розумні двері для вихованців.

1.4.3 Маркетинг, торгівля, фінанси

Ця категорія відноситься до будь-якої галузі, де здійснюється роздрібна торгівля. Це може бути магазин або торговий намет. Крім того, ця категорія також тісно пов'язана з фінансовими організаціями та сферою маркетингу. Сюди входять традиційні банківські та страхові послуги, а також дозвільний та готельний бізнес. Інтернет речей у сфері роздрібної торгівлі вже впливає на цю сферу, його завдання – знизити витрати реалізації та підвищити якість обслуговування. Для реалізації цього завдання існує безліч інструментів IoT [7].

У цьому сегменті цінність виявляється у негайних фінансових операціях. Якщо інтернет речей не приносить цього результату, необхідно переглянути доцільність вкладень у рішення IoT. Це привносить додаткові складності як і необхідності знаходити нові способи зниження витрат чи підвищення доходів. Якщо покупці зможуть ефективніше вирішувати свої завдання, продавці товарів та послуг зможуть обслуговувати швидше та обходитися меншим штатом співробітників.

Приклади застосування інтернету речей для роздрібної торгівлі:

- Цільова реклама, наприклад пошук фактичних або потенційних покупців у безпосередній близькості та надання їм інформації про товар/послуги;
- Оповіщення, наприклад, маркетинговий аналіз на основі таких даних, як розпізнавання наближення клієнта, схема руху та інтервали часу;
- Облік матеріальних активів, зокрема інвентаризація, управління збитками та оптимізація системи постачання;
- Цифрові вивіски у торгових точках, готелях та по місту;
- Системи оповіщення у розважальних закладах, на конференціях, концертах, парках розваг та музеях.

1.4.5 Транспортування та логістика

Транспортна сфера та логістика стануть важливою, якщо не основною, сферою застосування інтернету речей. До прикладів застосування інтернету речей у цій сфері відноситься відстеження вантажу, що доставляється, переміщується або транспортується, який би транспорт не використовувався: фура, поїзд, літак або корабель. Сюди відноситься підключення до інтернету транспортних засобів, завдяки чому вони можуть пропонувати водієві допомогу або здійснювати профілактичний ремонт і обслуговування замість водія. На даний момент будь-який середньостатистичний транспортний засіб, куплений новим, оснащений приблизно 100 датчиками. Ця цифра подвоїться, коли такі функції, як зв'язок між

машинами, зв'язок між машиною та дорожньою інфраструктурою та автоматичне керування, стануть обов'язковою умовою безпеки та комфорту. Усе це поширюється як на приватні транспортні засоби, так і на залізничний транспорт і морські вантажоперевезення, де немає можливості простою. Ще один варіант застосування – технічна допомога на дорогах, можливість відслідковувати ситуацію із службовими автомобілями. Деякі випадки застосування можуть бути дуже простими, але при цьому дуже дорогими, наприклад, відстеження розташування вільних автомобілів аварійно-ремонтної служби. Необхідні також системи автоматичної побудови маршруту для службових автомобілів та технічного персоналу, залежно від ситуації на дорозі.

Ця мобільна категорія відрізняється тим, що тут важливу роль відіграє геолокація. Основна частина геолокаційних даних надходить через GPS-навігацію. Інтернет речей спирається на такі дані, як ресурси та час, а в цьому випадку й просторові координати.

Приклади застосування інтернету речей у сфері транспортування та логістики:

- відстеження переміщень та місцезнаходження автомобілів з парку;
- ідентифікація та відстеження залізничних вагонів;

Висновок до першого розділу

Інтернет речей є одним з найбільш перспективних напрямків у розвитку інформаційних технологій. Такі мережі мають ряд переваг, завдяки яким вони стають популярними в багатьох сферах життя. Сьогодні він вже активно застосовується в багатьох галузях економіки, починаючи від виробництва та логістики і закінчуючи медициною та побутовою сферою.

РОЗДІЛ 2 ПРОТОКОЛИ ПЕРЕДАЧІ ДАНИХ

2.1 Адресація у мережі

У кожного пристрою в мережі для його ідентифікації має бути будь-яка адреса. У мережах, що повсюдно використовуються - TCP/IP використовується 3 види адрес [1]:

- фізична чи апаратна;
- IP-адреса;
- символьна, або доменна адреса.

Фізична адреса – це унікальний номер, присвоєний мережному пристрою виробником відповідно до виділеного діапазону. Якщо підмережею є локальна мережа Ethernet, Token Ring або FDDI, то фізична адреса – це адреса MAC (Media Access Control). MAC адреса є унікальною, її розмір – 6 байт і зазвичай вона записується у шістнадцятковому вигляді, наприклад 00-aa-00-64-c8-09.

MAC-адреса використовується пристроями канального рівня моделі OSI – комутаторами для адресації в локальних мережах.

IP адреса є основним типом адреси, який використовується на мережевому рівні для передачі та прийняття даних у вигляді IP пакетів. IP адреса на відміну від MAC адреси не закладається у пристрої виробником і може бути переналаштована. Пристрій може одночасно входити до кількох сегментів мереж. У цьому випадку воно має кілька IP адрес, за кількістю підключених мережевих портів. Номер вузла (мережевого з'єднання) у протоколі IP призначається незалежно від його локальної адреси. Отже, IP-адреса характеризує не сам пристрій, підключений до мережі, а одне мережне з'єднання. Існують два типи IP адрес: IPv4 і IPv6 - IP версії 4 і 6 відповідно.

Адреса IPv4 складається з 32 біт (4 байт), зазвичай його записують у вигляді чотирьох чисел від 0 до 255 включно, розділених точками, наприклад: 25.63.103.12. За допомогою протоколу IPv4 можна створити 2^{32} адреси, що дорівнює числу 4 294

967 296. Враховуючи що кількість пристроїв, підключених до мережі швидко збільшується кожного року, така кількість унікальних адрес виявилася недостатньою, після чого був розроблений протокол IPv6 (стандарт протоколу IPv6 – документ RFC 2460 – був прийнятий у 1998 р.).

Адреса IPv6 складається з 128 бітів (16 байт), в 4 рази більше ніж в IPv4, що дає йому можливість створення 2^{128} адрес, якщо виразити числом, то буде 340 282 366 920 938 463 463 374 607 431 768 211 456 достатньо для сьогоднішніх реалій і на найближче майбутнє.

Символьні (доменні) адреси використовуються для комфортної роботи з адресами. Людям зручніше використовувати текстові адреси, ніж цифрові. Спеціальна служба DNS (Domain Name System) встановлює відповідність між доменними іменами та IP адресами [8].

IP адреса поділяється на дві логічні частини, які можуть складатися з різної кількості біт у різних мережах. Перша частина визначає номер підмережі, друга частина – номер вузла цієї підмережі. Коли пакет потрапив у мережу призначення номер хоста вказує на конкретний вузол в рамках цієї підмережі. Якщо відома кількість розрядів, що відводяться для представлення номера вузла (або номера підмережі), можна визначити загальну кількість вузлів (або підмережі).

Якщо число розрядів, що визначають номер вузла в підмережі, дорівнює N, то загальна кількість можливих вузлів дорівнює $(2^N – 2)$. Дві адреси (найнижча і найвища) зарезервовані для особливих функцій мережі.

Щоб однозначно визначити, яка частина IP-адреси відповідає за номер підмережі, а яка за номер хоста, застосовуються маски підмережі. У двійковому виді маски підмережі біти, що дорівнює 1, позначають номер підмережі, а біти, рівні 0, – номер хоста. Наприклад, маска 255.255.255.0 означає, що в адресі 24 біта виділяється для позначення номера підмережі та 8 для номера хоста.

При розробці IPv6 був спрощений протокол для того, щоб маршрутизатори могли обробляти пакети IPv6 швидше, і забезпечена можливість захисту даних за допомогою шифрування. Масок в IPv6 немає, але основна зміна – це довгі адреси відправника та одержувача. Як вже писалося раніше, адреси складаються з 128 біт,

вони поділяються на вісім 16-бітних блоків, що записуються в шістнадцяткових числах, розділених двокрапками. Якщо 2 або більше блоків складаються з нулів, то можна скоротити адресу, наприклад, 2001:0db8:0000:0000:0000:0000:ae21:ad12 перетворюється на 2001:db8::ae21:ad12.

Існує проблема з використанням протоколу IPv6 в системах інтернету речей, оскільки найчастіше адреса займає більше пам'яті, ніж дані, які потрібно передати. Ця проблема вирішується за допомогою мережної технології 6LoWPAN (IPv6 over Low Power Wireless Personal Area Nerworks), як показано на рис. 2.1.

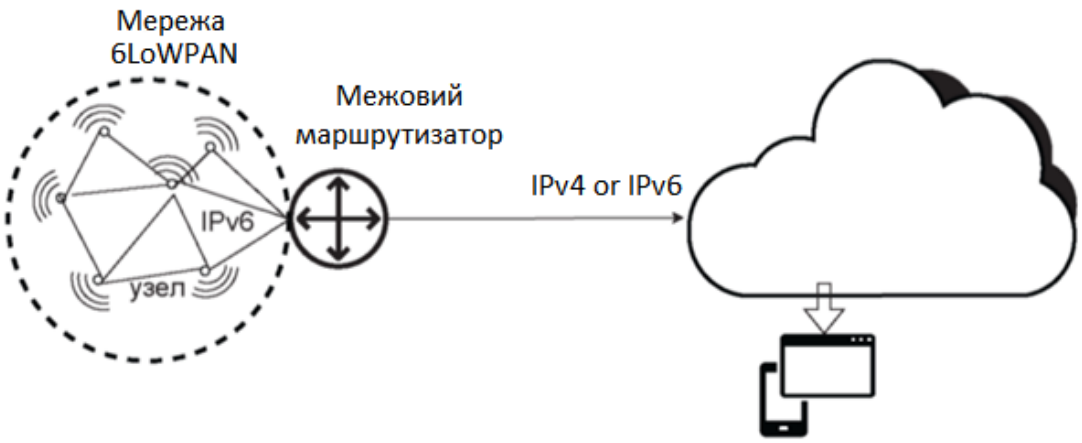

Рис 2.1 – Технологія 6LoWPAN

Умовно виділяють «маленький» інтернет, в якому знаходяться невеликі IoT пристрої, і «великий» (глобальну мережу), а посередині – шлюз, який перетворює довгу IPv6 адресу на коротку 16-бітну унікальну адресу «маленької мережі».

Підключення пристроїв до «великого» інтернету реалізується шляхом ефективної передачі пакетів IPv6 у невеликих кадрах канального рівня, визначених у бездротовому стандарті IEEE 802.15.4

2.2 Бездротові мережі не на основі IP

Мережі з використанням IP і без описуються в різних підрозділах, оскільки IP системи потребують додаткової деталізації, що не потрібні в IP-мережах.

Системи зв'язку, відмінні від IP, оптимізовані для економії витрат і енергоспоживання, тоді як рішення на базі IP зазвичай мають менше обмежень (наприклад, Wi-Fi).

2.2.1 Bluetooth

Bluetooth – це технологія бездротового зв'язку з низьким енергоспоживанням, що використовується повсюдно в інформаційних технологіях від датчиків стільникових телефонів до клавіатур та фітнес браслетів. Бездротовий зв'язок Bluetooth складається з двох бездротових систем: Basic Rate (BR) та Low Energy (LE або BLE).

З'єднання Bluetooth відбувається таким чином: ініціатор з'єднання знаходить поблизу готовий до підключення пристрій, обирає канал і запитує з'єднання. Якщо готовий до підключення пристрій згоджується на з’єднання, то ці пристрої з’єднуються. Ініціатор тепер називається майстром або ведучим, а інший пристрій називається веденим. Це з'єднання називається пікомережею. Всі події з'єднання відбуваються на тому самому початковому каналі між ведучим і веденим. Після обміну даними та завершення з'єднання для пари може бути обраний новий канал з використанням стрибкоподібної перебудови частоти.

Пікомережі формуються у двох різних режимах залежно від режиму: BR/EDR чи режиму BLE. В BR/EDR пікомережа використовує трибітну адресацію і може посилатися тільки на сім підлеглих пристроїв в одній пікомережі. Кілька пікомереж можуть утворювати об'єднання, зване scatternet. Але тут має бути другий майстер для підключення до вторинної мережі та управління нею. Ведений/головний вузол бере на себе відповідальність за об'єднання двох пікомереж разом. У режимі BR/EDR мережа використовує один і той же графік стрибкоподібної перебудови частоти, і всі вузли будуть гарантовано перебувати на одному каналі на даний момент часу. У режимі BLE ця система використовує 24-бітну адресацію, тому кількість можливих керованих пристроїв, пов'язаних з майстром, виражається в мільйонах [9]. Кожне відношення «ведучий-відомий»

саме по собі є пікомережею та може використовувати унікальний канал. У пікомережі вузли можуть бути ведучими, підлеглими, резервними чи зарезервованими.

На рисунку 2.1 показані топології пікомережей Bluetooth:

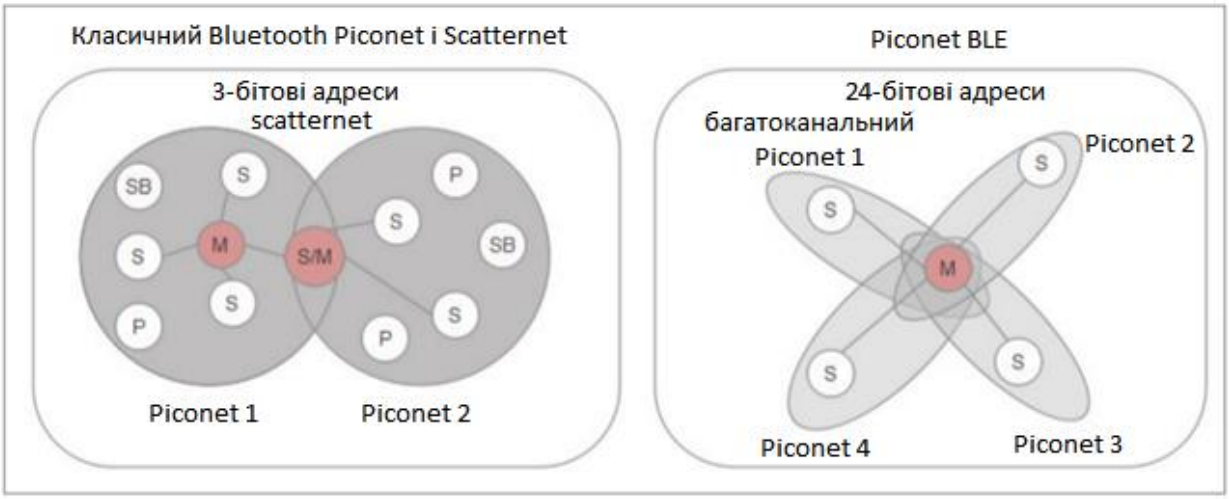

Рис. 2.2 – Пікомережі Bluetooth

Bluetooth має три основні компоненти: апаратний контролер, програмне забезпечення для хоста та профілі додатків. Пристрої Bluetooth поставляються в одно- та дворежимних версіях, що означає, що вони або підтримують лише стек BLE, або одночасно підтримують класичний режим та BLE. Bluetooth дозволяє підключати один або кілька контролерів до одного хоста.

Існують три режими роботи Bluetooth:

- Режим низького споживання енергії (LE) – використовується смуга ISM 2.4 ГГц і FHSS для захисту від перешкод. LE працює із частотою 1 Мсим/с зі швидкістю передачі 1 Мбіт/с. Bluetooth 5 підтримує кілька налаштовуємих швидкостей передачі даних 125 Кбіт/с, 500 Кбіт/с, 1 Мбіт/с та 2 Мбіт/с;
- Режим базової швидкості/покращеної швидкості передачі даних (BR/EDR) – використовується не таке радіо, як BLE та AMP, але воно працює в діапазоні ISM 2.4 ГГц. Базова радіостанція розрахована на 1 Мсім/с та підтримує швидкість передачі 1 Мбіт/с. EDR підтримує швидкість передачі даних 2 або 3 Мбіт/с. Використовується FHSS для захисту від перешкод;

- Альтернативний MAC/PHY (AMP) – це додаткова функція, яка використовує 802.11 для високошвидкісної передачі до 24 Мбіт/с. Цей режим вимагає, щоб провідний та ведений пристрій підтримували AMP. Це вторинний фізичний контролер, але він вимагає, щоб система мала контролер BR/EDR для встановлення початкового з'єднання та узгодження.

Класичний режим Bluetooth (BR/EDR) орієнтований на з'єднання. Якщо пристрій підключено, зв'язок підтримується, навіть якщо дані не надсилаються. Перш ніж з'явиться з'єднання Bluetooth, пристрій повинен бути виявлений, щоб він відповідав на сканування фізичного каналу і згодом відповідав адресою свого пристрою та іншими параметрами.

Процес підключення виконується у 3 етапи:

1) запит - на цьому етапі два пристрої Bluetooth ніяк не асоційовані або не пов'язані; вони нічого не знають один про одного. Пристрої повинні знайти один одного через запит. Якщо інший пристрій слухає, він повинен відповісти своєю адресою;

2) пейджинг - це утворення з'єднання між двома пристроями. Кожен пристрій знає адресу один одного у цей момент;

3) підключено – існує чотири підрежими стану підключення. Це нормальний стан, коли два пристрої активно спілкуються:

а) активний режим – це нормальний режим роботи для передачі та прийому даних Bluetooth або очікування наступного слота передачі;

б) режим аналізу – режим енергозбереження. Пристрій по суті спить, але прослуховуватиме передачі під час певних слотів, які можуть бути змінені програмно;

в) режим очікування – це тимчасовий режим низького енергоспоживання, ініційований провідним чи веденим пристроєм. Пристрій не слухатиме передачі, як у режимі аналізу, і ведений тимчасово ігнорує пакети. У цьому режимі переключення на підключений стан відбувається дуже швидко;

г) режим паркування – цей режим застарів у Bluetooth 5

Mesh-мережа Bluetooth заснована на BLE і розташовується на фізичному та канальному рівнях BLE. Вона використовує концепцію заливок. Вузли в mesh-мережі Bluetooth включають наступне:

- *Вузли* – пристрої Bluetooth, які були попередньо підготовлені та є членами mesh-мережі;
- *Непідготовлені пристрої* – пристрої, які можуть приєднатися до mesh-мережі, які ще не є її частиною та не були підготовлені;
- *Елементи* – вузол з декількома складовими частина. Кожна частина може контролюватись та адресуватися незалежно. Прикладом може бути вузол Bluetooth з датчиками температури та вологості;
- *Шлюз mesh-мережі* – вузол, який може переводити повідомлення між mesh-мережею та технологією не-Bluetooth;

Після підготовки вузол може підтримувати необов'язковий набір функцій, які включають:

- *Реле* – вузол, який підтримує реле, називається вузлом ретрансляції і може повторно передавати отримані повідомлення;
- *Проксі* – дозволяє пристроям Bluetooth LE, які не підтримують mesh-мережі Bluetooth спочатку, взаємодіяти з вузлами в mesh-мережі. Це виконується за допомогою проксі-вузла. Проксі-сервер надає інтерфейс застарілим пристроям Bluetooth, і визначається проксі-протокол, що базується на каналі, орієнтованому на з'єднання. Успадкований пристрій зчитує і записує протокол проксі-сервера, а проксі-вузол перетворює повідомлення на справжні PDU mesh-мережі;
- *Мінімальна потужність* – деякі вузли в mesh-мережі повинні мати надзвичайно низькі рівні енергоспоживання. Вони можуть обслуговувати інформацію про екологічні датчики (наприклад, температуру) один раз на годину і налаштовуватися за допомогою вузла або хмари, що управляє, один раз на рік. Вузол входить у роль, яка називається вузлом низької потужності (LPN), який з'єднує його з дружнім вузлом. LPN переходить у стан глибокого сну та періодично

опитує пов'язаного друга для будь-яких повідомлень, які могли виявитися під час сну;

- *Друг* – дружній вузол, пов'язаний з LPN, не обов'язково обмежується потужністю. Обов'язок друга полягає в тому, щоб зберігати та буферизувати повідомлення, призначені для LPN, доки LPN не прокидається і не опитає його для повідомлень. Багато повідомлень може бути збережено, і друг передасть їх у порядку, використовуючи прапор додаткових даних.

Безпека в Bluetooth BR/EDR при сполученні досягається таким чином: для сполучення генерується секретний симетричний ключ. У режимі BR/EDR це називається ключем з'єднання, тоді як у режимі BLE він називається довгостроковим ключем. Старі пристрої Bluetooth використовували режим сполучення персонального ідентифікаційного номера (PIN), щоб ініціювати ключі з'єднання. Нові пристрої використовують безпечне просте сполучення.

Безпечне просте сполучення забезпечує процес сполучення за допомогою низки різних моделей асоціацій для різних випадків дослідження. Воно також використовує криптографію з відкритим ключем для захисту від підслуховування та атак типу "людина-посередині".

Аутентифікація в режимі BR/EDR – це виклик-відповідь; наприклад, введення PIN-коду на клавіатурі. Якщо аутентифікація не вдалася, пристрій буде очікувати деякий час, перш ніж вирішити нову спробу. Інтервал зростає експонентно з кожною невдалою спробою. Це просто розчаровує людину, яка намагається вручну підібрати код ключа.

Шифрування в режимі BR/EDR може бути встановлене таким чином, щоб воно було відключено для всього трафіку, так що зашифрований трафік даних, але широкомовний зв'язок буде необробленим або щоб весь зв'язок був зашифрований. Шифрування використовує криптографію AES-CCM.

2.2.2 Zigbee

Zigbee – це протокол, призначений для комерційних та житлових мереж IoT, які обмежені вартістю, потужністю та простором. У мережі Zigbee є три основні компоненти [10]:

- *Контролер Zigbee* (ZC) – високопродуктивний пристрій мережі Zigbee, який використовується для формування та запуску мережевих функцій. Кожна мережа Zigbee матиме один ZC, який виконує роль координатора мережі. Після формування мережі ZC може поводитися як ZR (маршрутизатор Zigbee). Він може призначати логічні мережеві адреси та дозволяти вузлам приєднуватись або залишати mesh-мережу;
- *Маршрутизатор Zigbee* (ZR) – це компонент є необов'язковим, виконує мережеву стрибкоподібну перебудову та координацію маршрутизації. ZR бере участь у маршрутизації повідомлень з кількома переходами і може призначати логічні мережеві адреси та дозволяти вузлам приєднуватися або залишати mesh-мережу;
- *Кінцевий пристрій Zigbee* (ZED) – це звичайно кінцевий пристрій, наприклад, вимикач світла або термостат. Віе має достатньо функціональних можливостей спілкування з координатором. Віе немає логіки маршрутизації; тому будь-які повідомлення, що надходять на ZED, які не націлені на цей кінцевий пристрій, просто передаються. Воно також не може виконувати асоціації.

Zigbee підтримує три основні топології:

- *Зоряна мережа* – один ZC з одним чи кількома ZED. Тільки розширює зв'язок між двома вузлами і тому обмежено відстанню від ZC. Також потрібен надійний зв'язок із єдиною точкою відмови у ZC;
- *Кластерне дерево* – мережа з кількома переходами, яка використовує маяк та розширює охоплення мережі та діапазон по мережі зірок. Вузли ZC та ZR можуть мати дочірні елементи, але ZED залишаються справжніми кінцевими точками. Вузли-нащадки взаємодіють лише зі своїм батьком (наприклад, з невеликою

мережею зірок). Батьки можуть спілкуватися вниз по дереву зі своїми дітьми або до свого батька. Досі існує проблема у єдиній точці відмови у центрі;

- *Mesh-мережа* – динамічне формування шляху та зміна форми. Маршрутизація може відбуватися з будь-якого вихідного пристрою будь-який цільовий пристрій. Використовує алгоритми маршрутизації дерева та таблиці. Радіостанції ZC та ZR повинні постійно бути запитаними, щоб виконувати вимоги до маршрутизації, забираючи для цього час автономної роботи. Крім того, обчислення затримки в mesh-мережі може бути важким, а то й недетерменованим. Деякі правила ослаблені; однак маршрутизатори у певному радіусі один від одного можуть безпосередньо зв'язуватися один з одним. Основною перевагою є те, що мережа може зрости за межі видимості та мати кілька додаткових шляхів. На рисунку 2.3 можна побачити приклади цих топологій, перша картинка – топологія «звезда», друга – кластерне дерево і третя – це mesh-мережа.

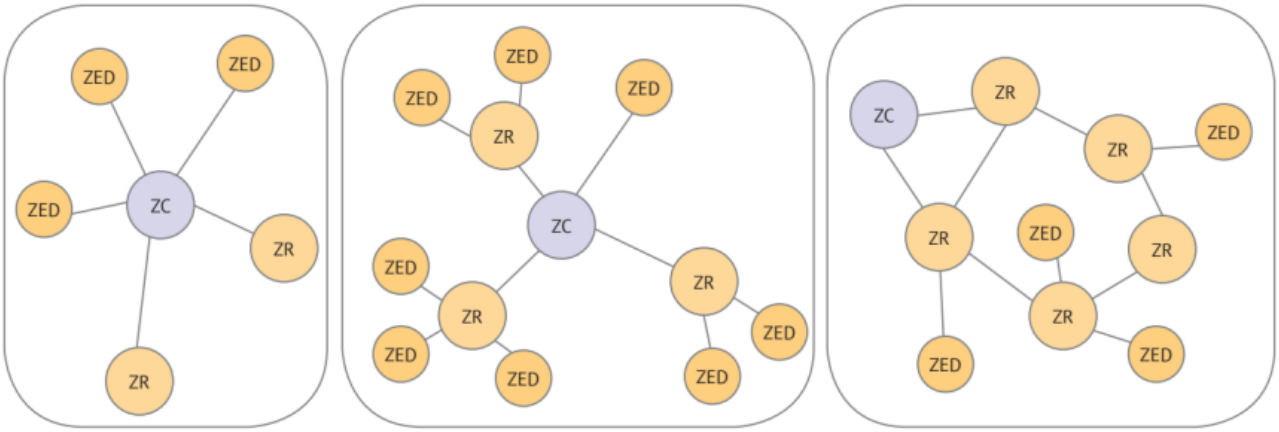

Рис. 2.3 – Топології Zigbee

Zigbee, як і Bluetooth, працює в основному в діапазоні 2.4 ГГц ISM. На відміну від Bluetooth він також працює на частоті 868 МГц у Європі та 915 МГц у США та Австралії. Через нижчу частоту він має кращу пропускну здатність проникати крізь стіни та перешкоди в порівнянні з традиційними сигналами 2.4 ГГц.

Zigbee використовує дві унікальні адреси для кожного вузла:

- *Довга адреса (64 біт)* – призначається виробником пристрою та незмінна. Унікально відрізняє пристрій Zigbee від інших пристроїв Zigbee. Верхні 24 біти

відносяться до організаційного унікального ідентифікатора, а нижні 40 біт керуються OEM. Адреси поставляються блоками та можуть бути замовлені у IEEE;

- *Коротка адреса (16 біт)* – ідентифікатор мережі, є необов'язковим.

Zigbee має можливість маршрутизувати пакети кількома способами:

- Широкомовлення – передача пакета до всіх інших вузлів у середовищі;
- Маршрутизація в mesh-мережі (таблична маршрутизація) – якщо існує таблиця маршрутизації для адресата, маршрут слідуватиме всім правилам таблиці відповідним чином.
- Маршрутизація дерева – це одноадресна передача повідомлень з одного вузла на інший. Маршрутизація дерева є необов'язковою і може бути заборонена у всій мережі. Вона забезпечує кращу ефективність пам'яті, аніж маршрутизація в mesh-мережі. Однак маршрутизація дерева не має такої надлишковості з'єднань, як mesh-мережі. Zigbee підтримує деревоподібну маршрутизацію до 10 вузлів;
- Маршрутизація від джерела – використовується головним чином за наявності концентратора даних.

Кінцеві пристрої Zigbee (ZED) не беруть участь у маршрутизації. Кінцеві пристрої взаємодіють із батьком, який також є маршрутизатором. Коли координатор Zigbee (ZC) дозволяє новому пристрою приєднатися до мережі, він переходить до процесу, відомого як асоціація.

Щоб формально приєднатися до мережі, запит маяка передається пристроєм для запиту наступних маяків від пристроїв у мережі, яким дозволено приєднувати нові вузли.

Zigbee будує правила безпеки відповідно до IEEE 802.15.4. Zigbee надає три механізми безпеки: списки керування доступом, 128-бітове шифрування AES та таймери свіжості повідомлень.

Модель безпеки Zigbee розподілена між кількома рівнями:

- Рівень додатків забезпечує створення ключів та транспортні послуги;

- Мережевий рівень управляє маршрутизацією, а вихідні кадри використовуватимуть ключ лінка, визначений маршрутизацією, якщо він доступний; в іншому випадку використовується мережевий ключ;
- Безпека рівня MAC управляється через API та контролюється верхніми рівнями.

Існує кілька ключів, якими керує мережа Zigbee:

- Майстер-ключ – може бути заздалегідь встановлений виробником або введений вручну користувачем. Він є основою безпеки пристрою Zigbee. Майстер-ключі завжди встановлюються першими та передаються з центру довіри;
- Мережевий ключ – цей ключ забезпечить захист мережевому рівні від зовнішніх зловмисників;
- Ключ з'єднання забезпечує надійне зв'язування між двома пристроями. Якщо у двох пристроях є вибір між використанням встановлених ключів зв'язку або мережевих ключів, вони завжди за промовчанням пов'язуватимуть ключі, щоб забезпечити додатковий захист.

Керування ключами має вирішальне значення для безпеки. Розподіл ключів контролюється шляхом створення центру довіри (один вузол виступає в ролі розподільника ключів до всіх інших вузлів у середовищі). Передбачається, що контролер є центром довіри. Можна реалізувати мережу Zigbee із виділеним центром довіри поза ZC. Центр довіри виконує такі послуги:

- Управління довірою – ідентифікує пристрої, що підключаються до мережі;
- Управління мережею – підтримує та розподіляє ключі;
- Керування конфігурацією: забезпечує захист від пристрою до пристрою.

Крім того, центр довіри може бути розміщений у режимі резидента (він не встановлюватиме ключі з мережевими пристроями) або може бути в комерційному режимі (встановлює ключі з кожним пристроєм у мережі).

Zigbee використовує 128-бітові ключі як частину своєї специфікації в межах канального та мережевого рівнів. Канальний рівень забезпечує три режими шифрування: AES-CTR, AES-CBC-128 та AES-CCM-128, які визначені у стандарті

IEEE 802.15.4). Однак мережевий рівень підтримує тільки AES-CCM-128, але трохи покращений, щоб забезпечити захист тільки шифрування та цілісності.

Цілісність повідомлень гарантує, що повідомлення не були змінені під час надсилання. Цей тип інструменту безпеки використовується для атак типу "людина посередині". Код цілісності повідомлення та допоміжний заголовок у структурі пакета Zigbee надають поля для перевірок для кожного надісланого повідомлення програми.

Аутентифікація забезпечується за допомогою загального мережного ключа та окремих ключів між парами пристроїв.

Таймери свіжості повідомлень використовуються для пошуку повідомлень із тайм-аутом. Ці повідомлення відхиляються та видаляються з мережі як інструмент для керування атаками повтору. Це стосується вхідних та вихідних повідомлень. Щоразу, коли створюється новий ключ, таймери свіжості скидаються.

2.3 Бездротові мережі на базі IP

2.3.1 Роль протоколу IP в інтернеті речей

З точки зору екосистеми, незалежно від протоколу, який використовується на рівні датчика, дані, отримані від датчика, в кінцевому результаті будуть передаватись у загальнодоступне, приватне або гібридне хмарне сховище для аналізу, контролю або моніторингу. Поза локальної мережі, світ базується на TCP/IP.

IP є стандартною формою глобального зв'язку з різних причин:

• Повсюдність - стеки IP використовуються майже кожною операційною системою та середовищем. Протоколи IP-зв'язку можуть працювати в різних локальних мережах, мобільних, провідних, оптичних волокнах та супутникових системах. IP визначає точний формат для всіх повідомлень та правил, що використовуються для зв'язку, підтвердження та управління зв'язком.

• Довговічність - TCP був створений у 1974 році, а стандарт IPv4, який все ще використовується сьогодні, був розроблений у 1978 році. Він витримав випробування часом протягом 40 років. Довговічність має першочергове значення для багатьох промислових та польових рішень IoT, які повинні підтримувати пристрої та системи протягом десятиліть. Різноманітні пропрієтарні протоколи були розроблені різними виробниками за ці 40 років, такі як AppleTalk, SNA, DECnet та Novell IPX, але жоден з них не завоював ринок так само, як IP.

• Масштабованість - IP-мережі продемонстрували значне масштабування для мільярдів користувачів, а IPv6 може забезпечити унікальну адресу для кожного атома землі і більше.

• Надійність – IP за своєю сутністю є надійним протоколом передачі даних. Він виконує це через систему доставки пакетів на основі мережі без встановлення з'єднання. Послуга вважається ненадійною за концепцією, тобто доставка не гарантується. IP є протоколом без встановлення з'єднання, оскільки кожен пакет обробляється незалежно від інших. IP також називається доставкою з найкращими зусиллями, тому що будуть зроблені всі спроби передати пакет різними маршрутами.

• Керованість: Існують різні інструменти для управління IP-мережами та пристроями в мережі IP. Існують інструменти моделювання, мережеві сніфери, діагностичні інструменти та різні пристрої, які допомагають створювати, масштабувати та підтримувати мережі.

2.3.2 6LoWPAN

З метою забезпечення адресації IP для найменших і найбільш обмежених ресурсів пристроїв у 2005 році була сформована концепція 6LoWPAN. 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) – це абревітура, що означає використання IPv6 з малопотужними локальними мережами. Мета полягає в тому, щоб використовувати IP-мережі через системи зв'язку з низьким енергоспоживанням для пристроїв, які обмежені в потужності та просторі та не

потребують високошвидкісних мережевих сервісів. Протокол може використовувати з іншими протоколами, такими як Bluetooth, Zigbee та ін. Основною перевагою 6LoWPAN є те, що найпростіший датчик може мати IP адресу і діяти як учасник мережі маршрутизації. Також варто відзначити широкий діапазон IPv6 адрес, що забезпечує 2^{128} або $3.5 \cdot 10^{38}$ унікальних адрес. Завдяки цій властивості IPv6 добре підходить для розвитку інтернету речей.

Мережі 6LoWPAN є mesh-мережами, розташованими на периферії великих мереж. Топології гнучкі, що дозволяє створювати спеціальні та незв'язані мережі без прив'язки до Інтернету або інших систем, або ж вони можуть бути підключені до Інтернету з використанням межових маршрутизаторів. Мережі 6LoWPAN можуть бути поєднані з кількома межовими маршрутизаторами – це називається багатоточковим підключенням. Крім того, спеціальні мережі можуть формуватись без необхідності підключення до інтернету межового маршрутизатора.

Межовий маршрутизатор необхідний для 6LoWPAN, оскільки він виконує 4 функції:

- підтримує зв'язок із пристроями 6LoWPAN та передає дані в інтернет;
- виконує стиснення заголовків IPv6, зменшуючи 40-байтовий заголовок IPv6 та 8-байтні UDP-заголовки для підвищення ефективності мережі датчиків. Типовий заголовок IPv6 може стискатися до 2-20 байт залежно від використання;
- ініціює мережу 6LoWPAN;
- займається обміном даних між пристроями в мережі 6LoWPAN.

Межові маршрутизатори утворюють mesh-мережі 6LoWPAN на більших традиційних мережевих периметрах. Вони також можуть здійснювати обмін між IPv6 та IPv4, якщо це необхідно. Дейтаграми обробляються як у IP-мережі, що має деякі переваги порівняно з пропрієтарними протоколами. Всі вузли в мережі 6LoWPAN мають той самий префікс IPv6, який встановлює межовий маршрутизатор. Вузли реєструватимуться на межових маршрутизаторах як частина фази виявлення мережі.

Під час фази виявлення мережі визначаються способи взаємодії хостів та маршрутизаторів у локальній мережі 6LoWPAN. Багатоточковість дозволяє кільком межовим маршрутизаторам 6LoWPAN керувати мережею; наприклад, коли для стійкості до збоїв потрібна наявність декількох носіїв (4G і Wi-Fi).

Існує три типи вузлів у mesh-мережі 6LoWPAN:

- *Вузли маршрутизатора* – ці вузли спрямовують дані з одного вузла мережі 6LoWPAN на інший. Маршрутизатори також можуть передавати інформацію у WAN та в інтернет;
- *Вузли хоста* - хости в mesh-мережі не можуть маршрутизувати дані в мережі і є просто кінцевими точками, що споживають або створюють дані. Хостам дозволено перебувати у стані сну та іноді прокидатися для отримання даних самим або отримання даних, кешованих їх батьківськими маршрутизаторами;
- *Межові маршрутизатори*: як зазначено, це шлюзи та контролери мережі, зазвичай, на краю WAN. Mesh-мережа 6LoWPAN керуватиметься межовими маршрутизаторами.

Вузли можуть вільно переміщатися та реорганізовуватись у мережі. У цьому випадку вузол може переміщатися та асоціюватися з іншим межовим маршрутизатором у сценарії з кількома керуючими елементами або навіть переміщатися між різними мережами 6LoWPAN. Ці зміни в топології можуть бути спричинені різними причинами, такими як зміна рівня сигналу або фізичне переміщення вузлів. При зміні топології адреса IPv6 пов'язаних вузлів також змінюється.

Стиснення заголовка є однією з основних дій у мережах 6LoWPAN. Як правило, стиснення заголовків засноване на статусі, що означає, що в мережі зі статичними лінками та стабільними з'єднаннями воно працює досить добре. У mesh-мережі, такій як 6LoWPAN, це не спрацює. Пакети переходять між вузлами та вимагають стиснення/декомпресії на кожному стрибку. Крім того, маршрути є динамічними і можуть змінюватися, і передачі можуть бути відсутніми протягом тривалого часу. Тому 6LoWPAN використовує стиск без збереження стану та

спільно використовуваного контексту. Тип стиснення може залежати від того, чи виконуються певні специфікації, а також залежно від того, де знаходяться джерело та місце призначення пакета.

Існують три випадки стиснення заголовка для 6LoWPAN залежно від того, де знаходиться маршрут: у самій мережі 6LoWPAN, поза мережею, але з відомою адресою або поза мережею з невідомою адресою. У першому випадку від початкового 40-байтного заголовка залишається 2 байти, у другому 12, а третьому 20.

Оскільки в бездротовій локальній мережі легко підслуховувати та слухати повідомлення, 6LoWPAN забезпечує безпеку на кількох рівнях. На рівні не IP протоколів 6LoWPAN використовує шифрування даних AES-128. Крім того, не-IP протоколи надають лічильник з режимом CBC-MAC (CCM) для забезпечення шифрування та перевірки цілісності. Більшість наборів мікросхем, які забезпечують мережевий блок не IP протоколів, так само включають механізм апаратного шифрування для підвищення продуктивності.

На мережному рівні 6LoWPAN має можливість використовувати стандартну безпеку IPsec. Це включає в себе:

- Обробник автентифікації: для захисту цілісності та автентифікації;
- Інкапсулювання корисного навантаження безпеки: додається шифрування для забезпечення конфіденційності в пакетах.

Інкапсулювання корисного навантаження безпеки є найбільш поширеним форматом захищеного пакету мережного рівня, крім того, воно визначає повторне використання AES/CCM, який використовується в апаратних засобах канального рівня для шифрування мережного рівня. Це робить безпеку мережного рівня придатною для обмежених вузлів.

2.3.3 IEEE 802.11

802.11 являє собою набір протоколів з багатою історією та різними варіантами використання. Wi-Fi – це визначення WLAN на основі стандартів IEEE 802.11, але підтримуваних та керованих некомерційним альянсом Wi-Fi.

Початкова мета 802.11 полягала у наданні протоколу канального рівня для бездротової мережі. У 2013 р. відбулася еволюція з базового стандарту 802.11 до 802.11ac. З того часу робоча група зосередилася інших областях. Специфічні варіанти 802.11 були розглянуті для використання та сегментів, таких як з'єднання IoT з низькою потужністю / низькою пропускною здатністю (802.11ah), зв'язок між автомобілями (802.11p), повторне використання телевізійного аналогового RF-простору (802.11af), екстремальна ширина для аудіо/відео (802.11af) і, звичайно, розвиток стандарту 802.11ac (802.11ax). Нові варіанти призначені для різних областей радіочастотного спектру або зменшення латентності та підвищення безпеки при виникненні аварійних ситуацій з автомобілями.

802.11 – це сімейство бездротового радіозв'язку на основі різних методів модуляції у смугах ISM 2.4 ГГц та 5 ГГц неліцензійного спектру. Wi-Fi сприйнятливий до тих же шумів та інтерференції, що і Bluetooth та Zigbee, і використовує різні методи для забезпечення надійності та відмовостійкості.

Системи 802.11 підтримують три основні топології (рис. 2.4):

- Інфраструктура – в цій формі станція відноситься до пристрою кінцевої точки 802.11 (наприклад, смартфону), який здійснює зв'язок із центральною точкою доступу. Точка доступу може бути шлюзом до інших мереж; Це також відоме як служба базового налаштування інфраструктури. Ця топологія є топологією зірки;
- За ситуацією – вузли 802.11 можуть формувати так званий незалежний базовий набір, де кожна станція зв'язується та керує інтерфейсом з іншими станціями. Ця конфігурація не використовує точку доступу або топології зірки. Це однорангова топологія;

- Система розподілу: ця система об'єднує дві або більше незалежних мереж інфраструктури через міжмережеві з'єднання точок доступу.

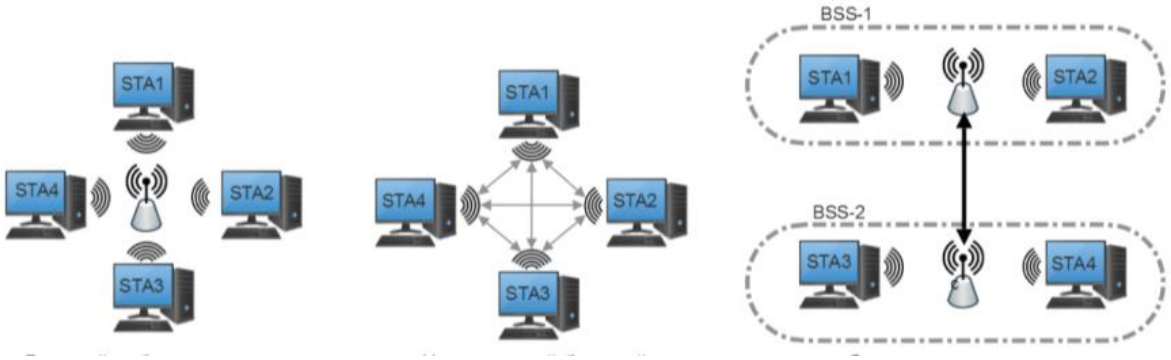

Рис. 2.4 – Основні топології 802.11

802.11ah та 802.11s підтримують топологію mesh-мережі.

Станцією вважається пристрій, оснащений контролером бездротового інтерфейсу. Станція завжди слухатиме активний зв'язок у певному каналі. Першою фазою підключення до Wi-Fi є фаза сканування. Існує два типи використовуваних механізмів сканування:

- Пасивне сканування – ця форма сканування використовує маяки та пробні запити. Після вибору каналу пристрій, що виконує сканування, отримуватиме сигнали маяків та проб від найближчих станцій. Точка доступу може передавати маяковий радіосигнал і якщо станція приймає передачу, вона може продовжити приєднання до мережі;

- Активне сканування – у цьому режимі станція спробує знайти точку доступу шляхом створення пробних запитів. Цей режим сканування використовує більшу потужність, але дозволяє швидше приєднуватися до мережі. точка доступу може відповісти на пробний запит із відповіддю на запит проби, який аналогічний сигналу маякового радіосигналу.

Маяки завжди транслюються найнижчими базовими швидкостями, щоб гарантувати, що кожна станція в діапазоні має можливість приймати маяк, навіть якщо вона не може підключитися до цієї конкретної мережі. Після обробки маяка наступною фазою підключення Wi-Fi є фаза синхронізації. Ця фаза потрібна, щоб клієнти були налаштовані на точку доступу. Пакет маяка містить інформацію, необхідну станції:

- SSID (Service Set Identifier) – це ідентифікатор набору послуг. 1-32 символьне мережеве ім'я.
- BSSID (Basic Service Set Identification) – це базовий ідентифікатор набору послуг. Унікальні 48-бітові такі угоди MAC-адреси. Формується комбінацією 24-бітного унікального ідентифікатора організації та 24-розрядного ідентифікатора виробника радіочіпа;
- Ширина каналу – 20 МГц, 40МГц тощо;
- Країна – список каналів, що підтримуються (для конкретної країни);
- Інтервал маяка – час TBTT;
- TIM/DTIM (Delivery Traffic Indication Message): час пробудження та інтервали для отримання широкомовних повідомлень – дозволяє здійснювати розширене керування живленням.
- Служби безпеки: WEP, WPA, WPA2.

Якщо станція виявляє точку доступу або іншу станцію для встановлення з'єднання, вона потім переходить до фази автентифікації. Якщо процес автентифікації та забезпечення безпеки успішно завершується, наступною фазою є асоціація. Пристрій надішле кадр запиту асоціації на точку доступу. Потім точка доступу відповість кадром відповіді асоціації, яка дозволить станції приєднатися до мережі або бути виключеною. Якщо станцію увімкнено, точка доступу видає ідентифікатор асоціації клієнту та додає його до списку підключених клієнтів.

На цьому етапі дані можуть обмінюватися з точкою доступу та навпаки. По всіх кадрах даних буде одержано підтвердження.

802.11ah – це варіант бездротових протоколів, призначених для IoT, Проект намагається оптимізувати пристрої з обмеженим терміном служби, які потребують тривалої дії батареї та можуть оптимізувати діапазон та пропускну спроможність. 802.11ah також називається HaLow [12].

Метою групи IEEE 802.11ah було створення протоколу з розширеним діапазоном для зв'язку у сільських районах та розвантаження мобільного трафіку. Вторинною метою було використання протоколу для бездротового зв'язку з

низькою пропускною здатністю в діапазоні субгігагерц. Специфікація була опублікована в 2016 році і вона найбільше відрізняється від інших стандартів 802.11, зокрема наступним:

- працює у діапазоні 900 МГц. Це дозволяє забезпечити гарне поширення та проникнення через матеріали та незалежність від атмосферних умов;
- ширина каналу змінюється і може бути встановлена на 2, 4, 8 або 15 МГц;
- доступні різні методи модуляції
- забезпечує підключення до тисяч пристроїв із єдиною точкою доступу;
- включає можливість ретрансляції зменшення потужності на станції і дозволяє використовувати грубу форму mesh-мережі з використанням методу одноразового переходу;
- дозволяє здійснювати розширене керування споживанням на кожному вузлі.

Пропускна здатність 802.ah становить від 150 Кбіт/с до 347 Мбіт/с, залежно від модуляції ширини каналів та використання потоків. Ширина каналів варіюватиметься залежно від регіону, в якому розгортається 802.11ah. Все в архітектурі стандарту 802.11ah спрямоване на оптимізацію загального діапазону та ефективності. Це стосується навіть довжини заголовків MAC.

Підключення кількох тисяч пристроїв до однієї точки доступу також виконується за допомогою унікального ідентифікатора асоціації в 13 біт. Це дозволяє групувати станції на основі критеріїв (освітлення у передпокої, вимикач освітлення тощо). Це дозволяє точці доступу підключатися до більш ніж 8191 станцій (звичайні стандарти 802.11 підтримують лише 2007 станцій). Тим не менш, багато вузлів здатні викликати величезну кількість зіткнень у каналі. Незважаючи на те, що кількість підключених станцій збільшилася, мета полягала у скороченні обсягу даних у шляху передачі цих станцій. Цільова група IEEE виконала це, вилучивши кілька полів, які не були особливо важливими для випадків використання IoT, такий як поля QoS та інш.

Ще одне покращення для керування живленням та ефективністю каналу пояснюється видаленням фреймів підтвердження. Вони не є явними для двонаправлених даних. Тобто обидва пристрої надсилають та отримують дані один від одного. Зазвичай фрейми підтвердження використовуються після успішного прийому пакета. Тут же прийом наступного пакета передбачає, що попередні дані були успішно отримані, і підтвердження не є обов'язковим.

З точки зору топології в мережі 802.11ah є три типи станцій:

- root access point – початок. Як правило, служить шлюзом для інших мереж;
- станція – типова станція 802.11 чи кінцевий клієнт;
- relaynode – спеціальний вузол, який поєднує AP-інтерфейс зі станцією;

На додаток до основних типів вузлів існують три стани енергозбереження, в яких може бути станція:

- карта індикації трафіку – прослуховує точку доступу передачі даних. Вузли періодично отримуватимуть інформацію про дані, буферизовані для них зі своєї точки доступу. Надіслане повідомлення називається інформаційним елементом карти;
- Станції, що не належать до карти індикації трафіку, - ведення переговорів з точкою доступу безпосередньо під час зв'язку для отримання часу передачі у вікнах з обмеженим доступом;
- Позапланові станції – не прослуховує жодних маяків та використовує опитування для доступу до каналів.

Енергоспоживання має вирішальне значення для датчиків IoT та периферійних пристроїв на основі батарей. Протоколи 802.11 відомі тим, що висувають високі вимоги до потужності. Для вирішення цієї проблеми 802.11ah використовує значення Max Idle Period, яке є частиною стандартних специфікацій 802.11. У загальній мережі 802.11 максимальний період очікування становить приблизно 16 годин залежно від часу 16-бітового дозволу. У 802.11ah перші два біти 16-розрядного таймера є коефіцієнтом масштабування, що дозволяє підвищити тривалість сну до п'яти років.

Ще одна функція енергозбереження називається цільовим часом пробудження і призначена для станцій, які рідко передають або отримують дані, що є дуже поширеним у IoT. Станція та пов'язана з нею точка доступу вестимуть переговори, щоб прийти до узгодженого цільового часу пробудження, і станція увійде в стан очікування доти, доки цей таймер не буде сигналізований.

У бездротових мережах Wi-Fi використовуються кілька видів аутентифікації:

- WEP – конфіденційність у провідному еквіваленті. Цей режим надсилає ключ у звичайному тексті клієнта. Потім ключ шифрується та відправляється назад клієнту. WEP використовує ключі різного розміру, але зазвичай вони мають 128 біт або 256 біт. WEP використовує спільний ключ, що означає, що той самий ключ доступний для всіх клієнтів. Його можна легко скомпрометувати, просто прослуховуючи та переглядаючи всі кадри автентифікації, що повертаються клієнтам, що входять до мережі, для визначення ключа, що використовується для всіх. Через слабкість генерації ключа перші кілька байтів псевдовипадкового рядка можуть виявити частину ключа. Перехоплюючи від 5 до 10 мільйонів пакетів, зловмисник може достатньо впевнено отримати достатньо інформації для розкриття ключа;
- WPA – захищений доступ Wi-Fi був розроблений як стандарт безпеки 802.11i для заміни WEP і є програмним / вбудованим програмним забезпеченням, що не вимагає нового обладнання. Одна істотна відмінність полягає в тому, що WPA використовує протокол цілісності тимчасового ключа, який виконує змішування та повторний ключ для кожного пакета. Це означає, що кожен пакет використовуватиме інший ключ для шифрування, на відміну від WEP. WPA починається зі створення ключа сеансу на основі MAC-адреси, тимчасового ключа сеансу та вектора ініціалізації. Це досить завантажує процесор, але виконується лише один раз за сеанс. Наступним кроком буде вилучення молодших 16 біт прийнятого пакета з результатом біт, згенерованим на першій фазі. Це 104-розрядний ключ для кожного пакета. Тепер дані можуть бути зашифровані;

- WPA-PSK – це попередній загальний ключ WPA або WPA-Personal. Цей режим існує там, де немає інфраструктури автентифікації 802.11. Тут використовується ключова фраза як загальний ключ. Кожна станція може мати власний попередньо відкритий ключ, пов'язаний з її MAC-адресою. Це схоже на WEP, і недоліки вже виявлені, якщо в ключі, що попередньо розділяється, використовується слабка кодова фраза;
- WPA2 – це замінює оригінальну розробку WPA. WPA2 використовує AES для шифрування, що набагато сильніше, ніж протокол цілісності тимчасового ключа WPA. Це шифрування також називається режимом CTR із CBC-MAC Protocol або CCMP.

2.4 Висновок до другого розділу

Використання стандартного зв'язку на основі IP значно спрощує дизайн та дозволяє швидко та легко масштабувати роботу. Масштабування має вирішальне значення для розгортань IoT, які можуть охоплювати тисячі чи мільйони вузлів. 6LoWPAN і Thread демонструють стаднрат, які можуть застосовуватися до традиційно не IP-протоколів, таким як Zigbee. Обидва протоколи дозволяють адресувати IPv6 та mesh-мережу в масивні мережі інтернету речей. 802.11 є важливим та надзвичайно успішним протоколом, який становить основу WLAN, але також може діяти у пристроях IoT та датчиках з використанням 802.11ah. Серед наведених протоколів немає однозначно "найкращого" протоколу з точки зору безпеки, оскільки кожен з них має свої переваги та обмеження, вибір оптимального протоколу залежить від конкретного використання та вимог безпеки для конкретного IoT-проекту. Важливо пам'ятати, що безпека залежить не тільки від протоколу, але й від правильного налаштування, використання сильних ключів шифрування, аутентифікації та інших механізмів безпеки в конкретній реалізації мережі.

Можна зробити висновок, що обидва види технологій бездротової передачі даних підходять для створення мереж інтернету речей, проте, залежно від ситуації, один вид може бути більш підходящим. Якщо потрібна простота, гнучкість та швидкість, то краще вибирати протоколи, засновані на IP, а якщо потрібна енергоефективність та можливість створення великих мереж із великої кількості вузлів, то краще вибирати протоколи без IP.

РОЗДІЛ 3 СТВОРЕННЯ МЕРЕЖІ ІНТЕРНЕТУ РЕЧЕЙ

3.1 Основи Packet Tracer

У цьому розділі буде проведено практичну роботу зі створення бездротових мереж інтернету речей. Для цього буде використано програмне забезпечення під назвою "Packet Tracer". Packet Tracer – це симулятор мережних пристроїв, розроблений компанією Cisco. Він є потужним засобом для моделювання, налаштування та налагодження комп'ютерних мереж. Packet Tracer дозволяє створювати віртуальні мережеві середовища, в яких можна розгортати та налаштовувати різні мережні пристрої, такі як маршрутизатори, комутатори, сервери та клієнтські комп'ютери. У цю програму також вбудовано можливості побудови мереж інтернету речей.

Для початку можна побудувати саму базову систему, яка складатиметься з наступних компонентів:

- датчик. Датчик отримуватиме дані з навколишнього середовища та передаватиме їх на мікроконтролер;
- мікроконтролер. Мікроконтролер отримуватиме дані з датчика, обробляти їх належним чином і передавати керуючий сигнал актуатору, якщо потрібно;
- акутатор. Актуатор або виконавчий пристрій отримуватиме керуючий сигнал від мікроконтролера і виконуватиме відповідну дію;

Спочатку слід відкрити Packet Tracer та вивчити інтерфейс (рис. 3.1)

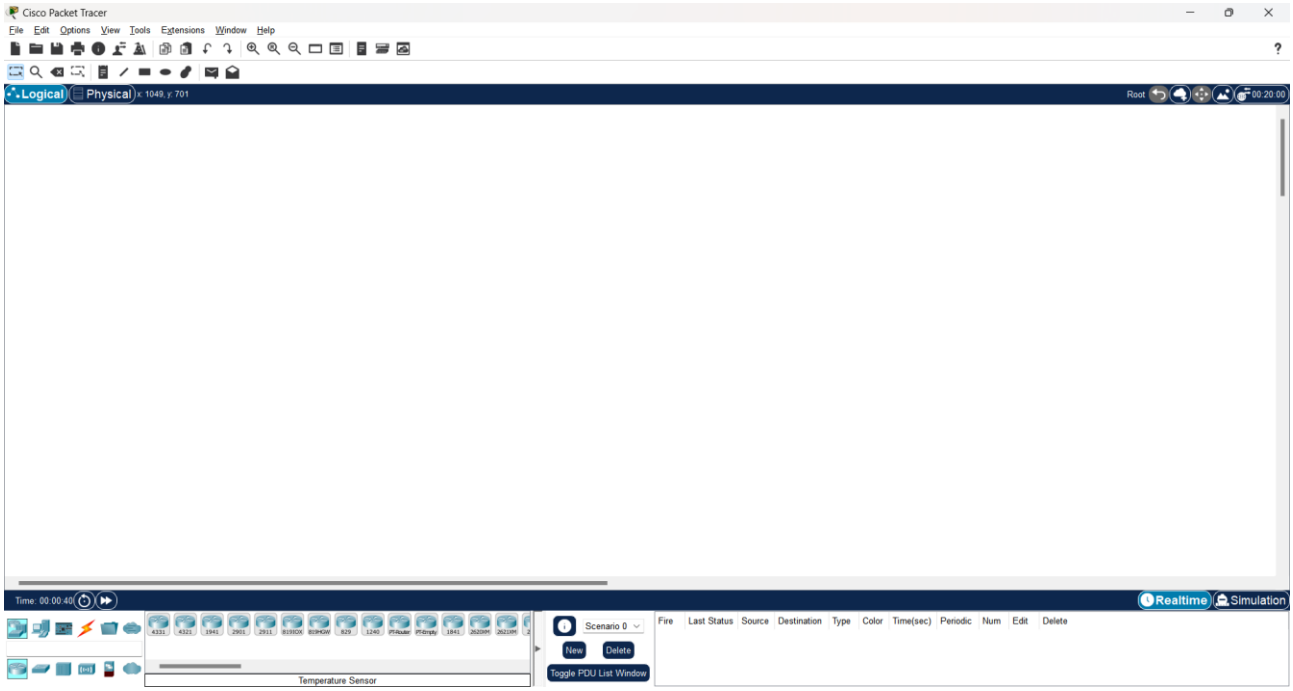

Рис. 3.1 – Інтерфейс Packet Tracer

Зліва знизу можна побачити категорії компонентів. Тут потрібно вибрати категорію "Components" та підкатегорію "Sensors". Як датчик буде використовуватися датчик температури. Далі потрібно знайти у списку потрібний нам елемент та перетягнути його на робочу область. Після цього треба вибрати підкатегорію Boards і взяти звідти MCU Board. Після цього з «Actuators» береться кондиціонер. Для зручності можна перейменувати компоненти. Виходить така схема (рис 3.2).

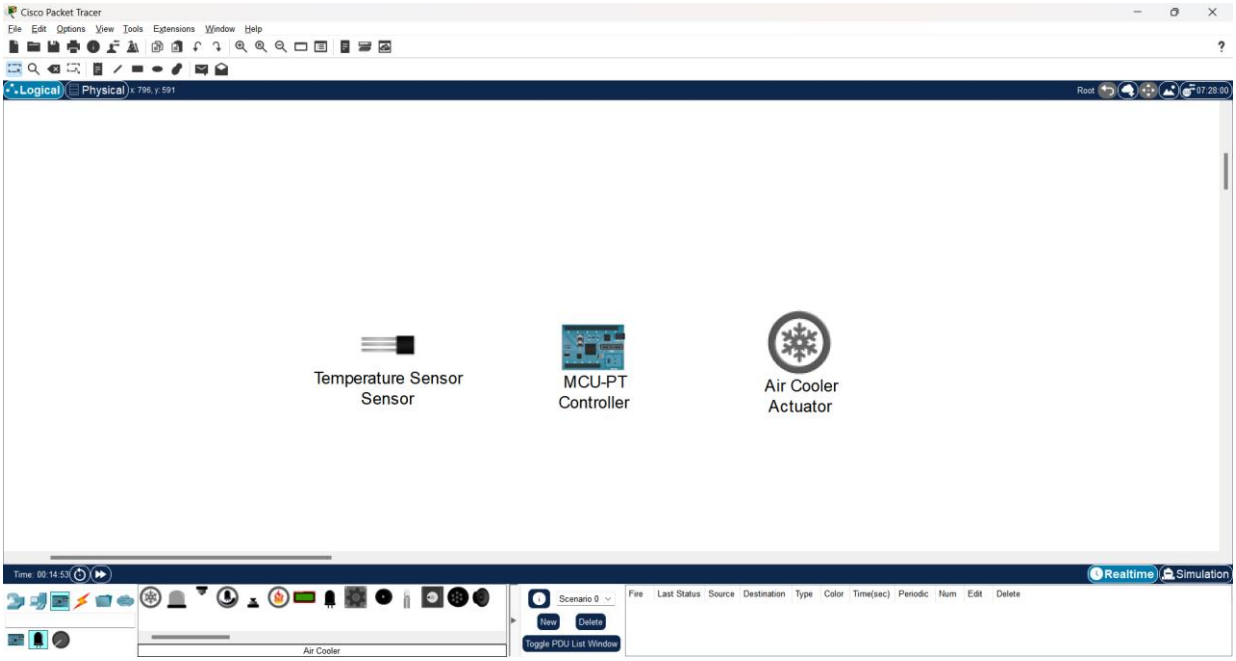

Рис. 3.2 – Компоненти мережі інтернету речей на робочій області

На робочій області є всі необхідні елементи. Тепер треба їх з'єднати та налаштувати так, щоб вони працювали належним чином. Можна поставити таке правило, що якщо температура вище 25 градусів цельсія, то треба включити кондиціонер. Для простоти спочатку буде розглянуто провідну систему. Для цього треба відкрити категорію "Connections" і натиснути на "IoT Custom Cable", далі натиснути на датчик і вибрати порт D0 після чого навести на мікроконтролер, натиснути на нього і вибрати порт D0. Далі потрібно зробити те саме, тільки з актуатором і замість порту D0 в мікроконтролері, підключити його в D1. Виходить така схема (рис 3.3).

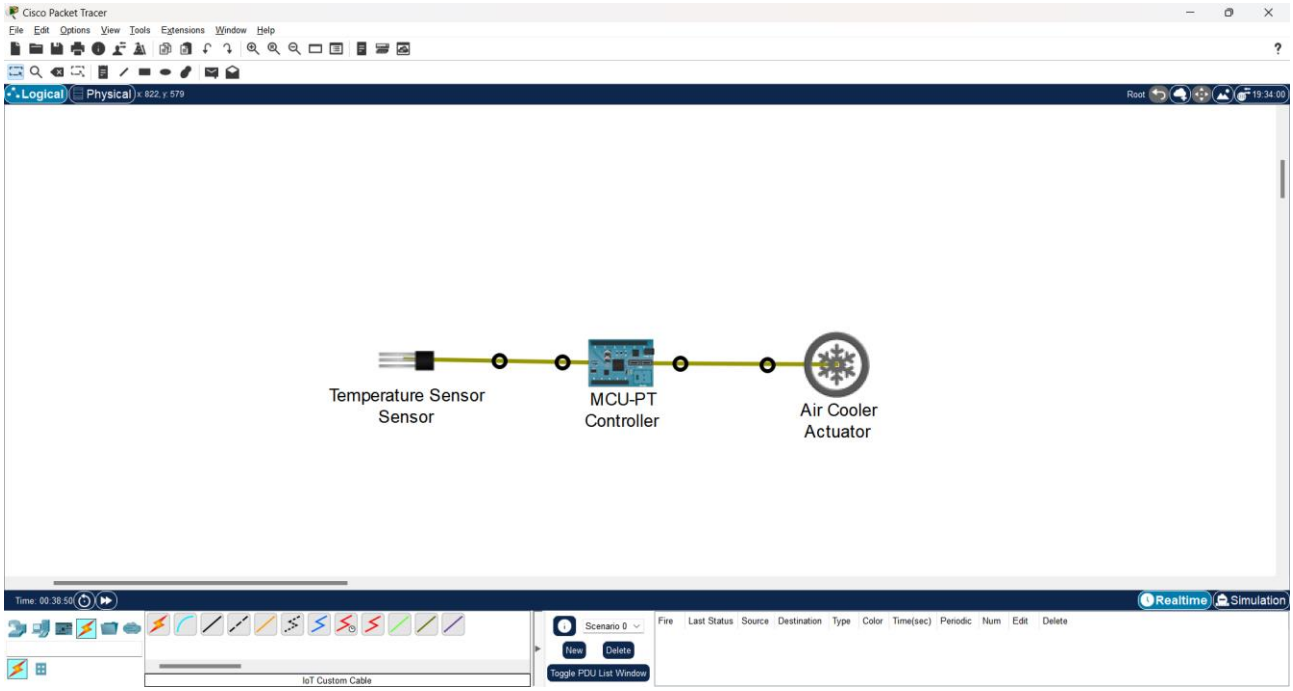

Рис. 3.3 – Базова мережа IoT

Компоненти з'єднані, але треба запрограмувати мікроконтролер, щоб вони працювали відповідним чином. Програмування мікроконтролерів є важливим елементом побудови низькорівневих мереж Інтернету речей. У сучасному світі існують готові рішення IoT, які треба лише підключити до маршрутизатора і керувати ними з персонального комп'ютера, використовуючи зручний і зрозумілий графічний інтерфейс. Однак мікроконтролери та їх програмування все ще користуються попитом за рахунок їхньої гнучкості, компактності та невеликої ціни. Для програмування мікроконтролера MCU-PT у Packet Tracer треба натиснути на мікроконтролер, перейти у вкладку "Programming", відкрити папку "Blink (JavaScript)" і в ній відкрити файл "main.js". Код усіх програм, зроблених у цій роботі буде наданий у додатку А. Цей мікроконтролер використовує синтаксис Arduino. За допомогою цього синтаксису на мові JavaScript можна написати такий невеликий скрипт (рис 3.4) [13].

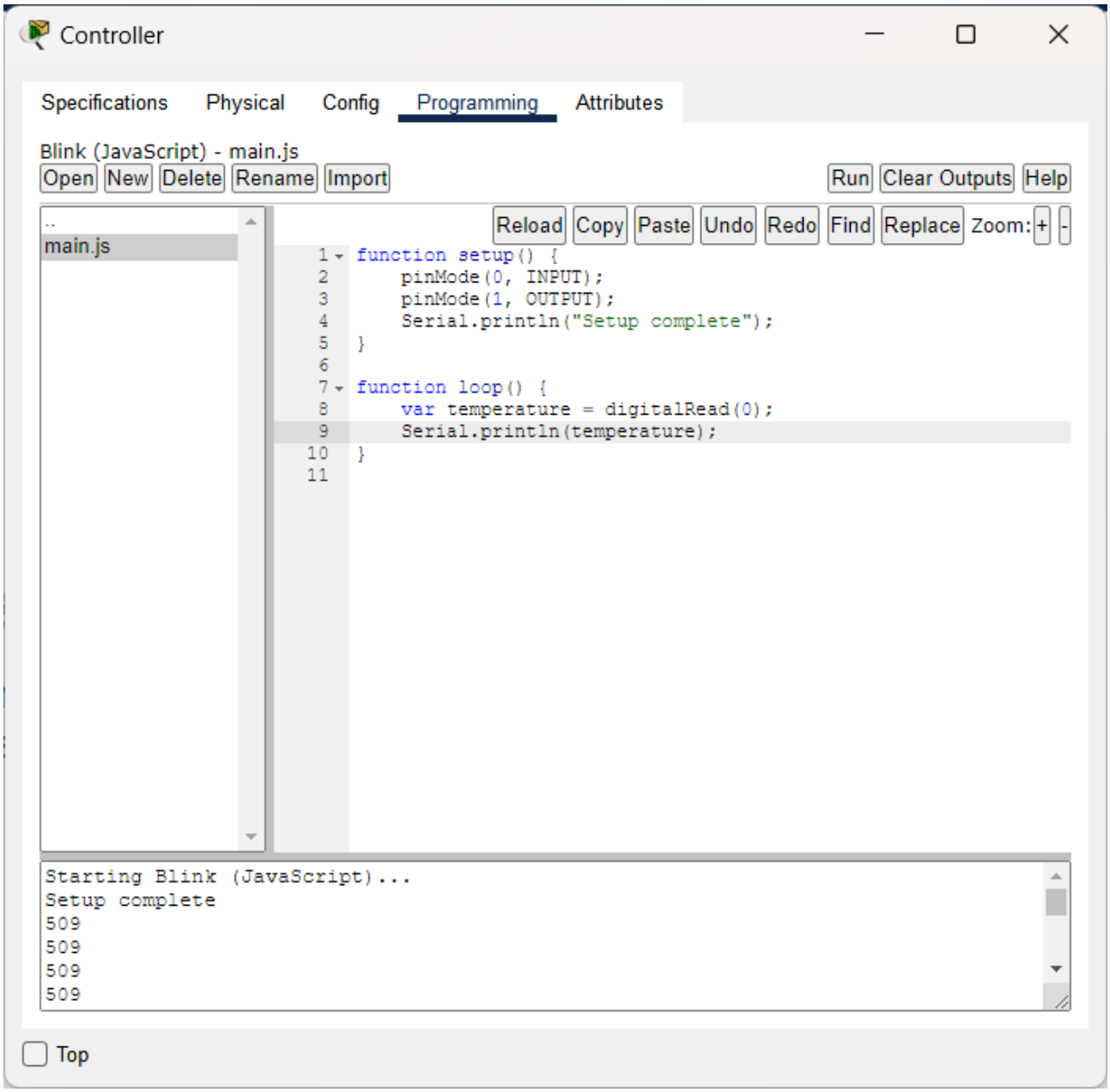

Рис. 3.4 – Програма контролера для зчитування даних

Програма складається із двох функцій. У першій відбувається налаштування пристрою, у другій знаходиться головний цикл програми, в якому знаходиться пристрій майже весь час роботи. У функції налаштування за допомогою функції pinMode налаштовується робота портів контролера. Нульовий порт, він же D0 налаштовується на прийом даних, а перший або D1 на передачу. У другій частині програми на восьмому рядку, за допомогою функції digitalRead читаються дані з нульового порту і передаються в змінну temperature, значення цієї змінної виводиться в консоль [14].

У консолі можна побачити число 509. Якщо натиснути на датчик температури, можна побачити його специфікацію, де написано, що діапазон роботи датчика – від -100 до 100 градусів цельсія, який проектується на діапазон чисел від 0 до 1023. Тобто число 509 відповідає температурі трохи менше нуля градусів.

Якщо відкрити специфікацію кондиціонера, він приймає два значення – LOW для вимкненого стану і HIGH для увімкнутого. Тобто для правильної роботи цієї мережі потрібно знайти значення в діапазоні від 0 до 1023, що відповідає температурі 25 градусів цельсія, і передавати HIGH на кондиціонер, якщо датчик подає число більше або дорівнює цьому числу, в іншому випадку передавати LOW.

Для цього можна вивести формулу

$$(25 + 100) \cdot (1023/200) = 639.375 \quad (3.1)$$

Можна округлити до 640. Далі потрібно запрограмувати мікроконтролер (рис 3.5).

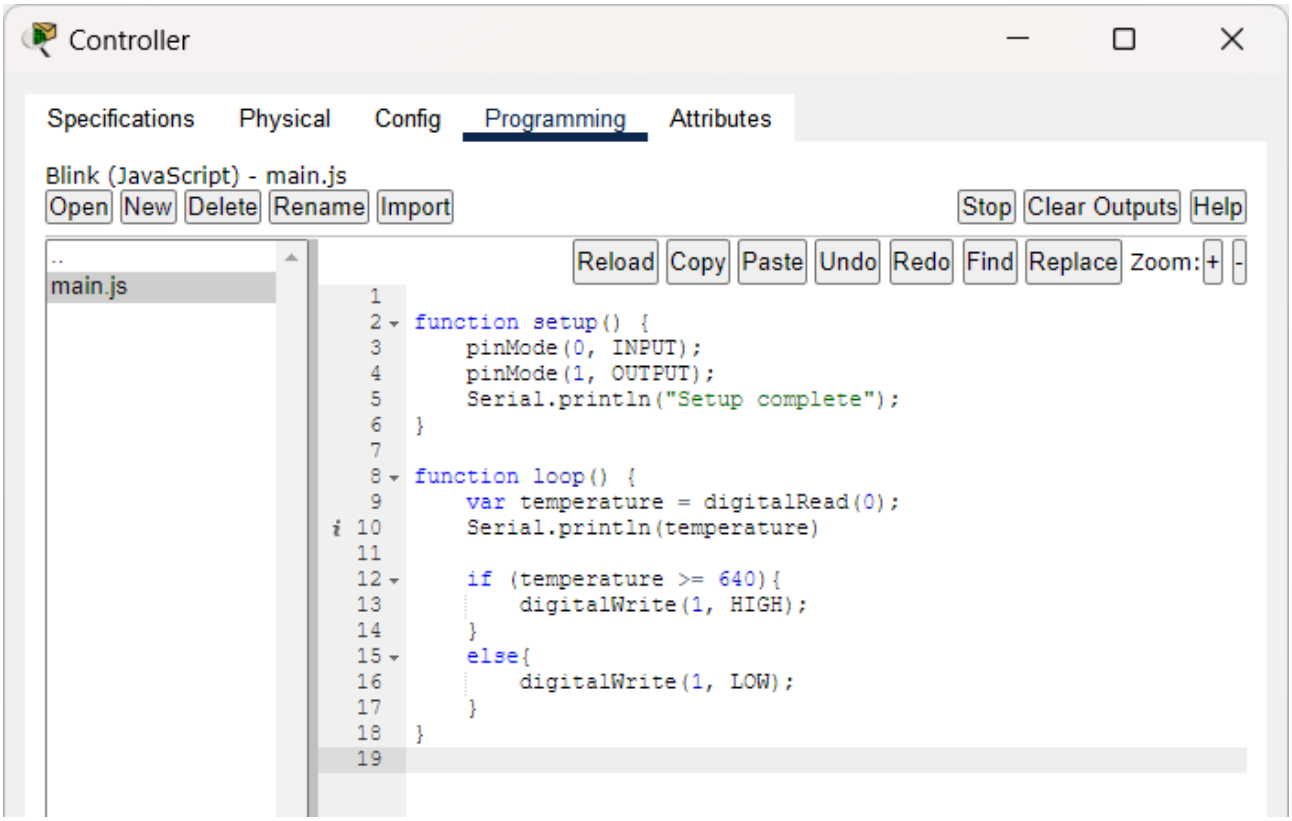

Рис. 3.5 – Повна програма контролера

На рисунку можна побачити умовний вираз, який означає, що якщо значення змінної «temperature» більше або дорівнює 640, то на порт 1 подається значення HIGH, що означає включення кондиціонера, інакше передається LOW, що означає вимкнення.

Для перевірки роботи мережі можна натиснути комбінацію клавіш shift + e, вибрати потрібну локацію зверху натиснути на кнопку «Edit» і вручну поміняти температуру (Рис 3.6).

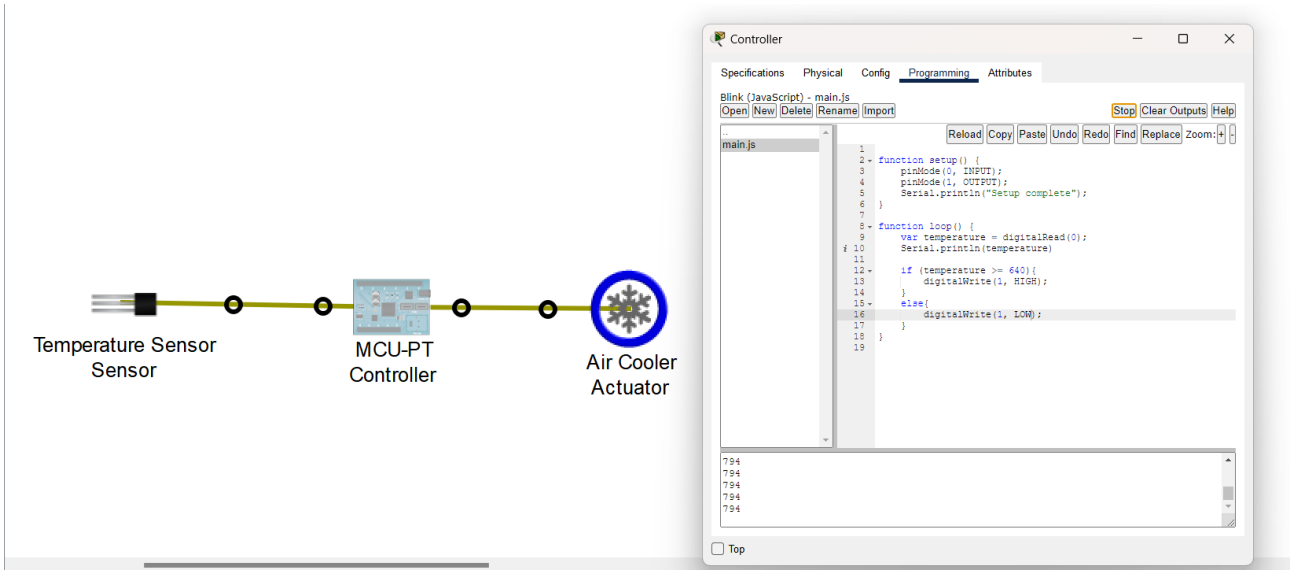

Рис. 3.6 – Робота дротового інтернету речей

У кондиціонера змінився колір, отже, він запрацював.

3.2 Бездротова мережа IoT на основі Bluetooth

Створення бездротової мережі інтернету речей складніше, ніж створення аналогічної дротової. Для створення такої мережі використовуватиметься Bluetooth. Мережа так само, як і попередня, складатиметься з датчика, контролера та актуатора. В цій мережі в якості датчика буде використовуватися датчик руху, а в якості актуатора буде лампочка. Як контролер використовуватиметься SBC Board, тому що в ньому є можливість використання Bluetooth.

Для початку знаходимо та перетягуємо потрібні елементи на робочу область (рис. 3.7).

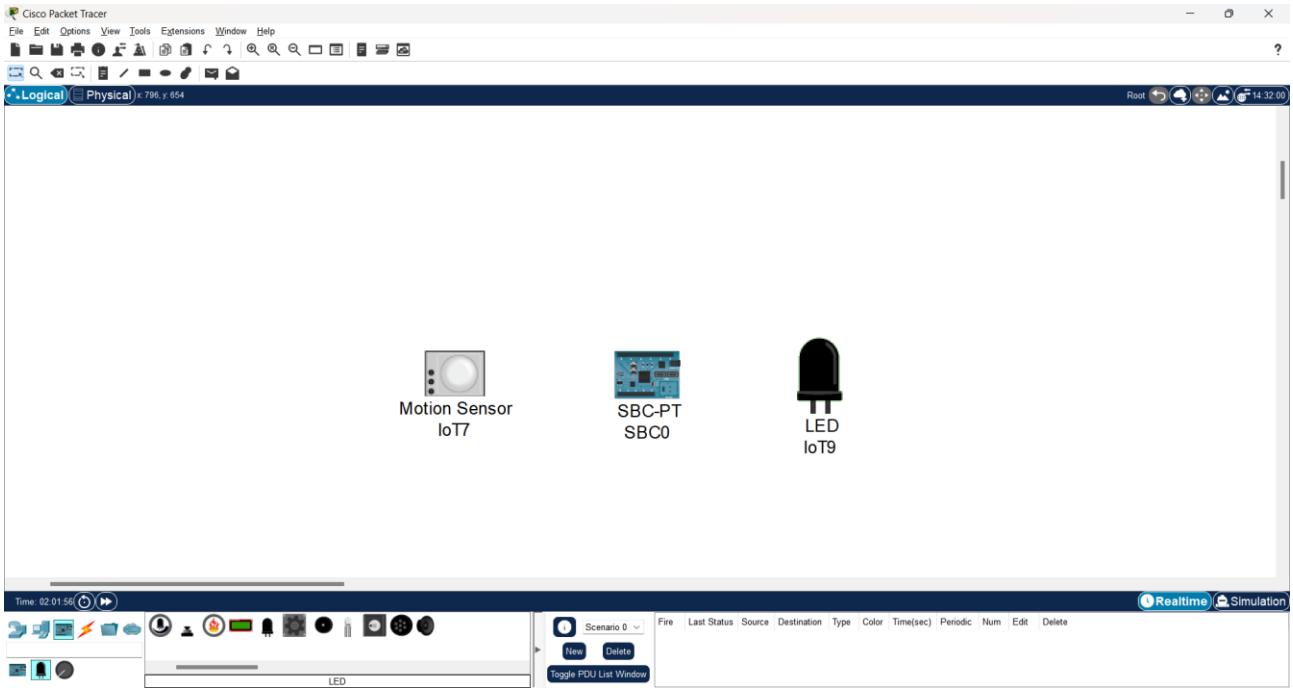

Рис. 3.7 – Розташування елементів у робочій області

Для того, щоб у пристроїв був Bluetooth, треба для датчика та лампочки виконати наступні дії: натиснути на іконку пристрою, натиснути праворуч знизу "advanced", зверху перейти у вкладку "I/O Config", в цій вкладці поставити галочку напроти Bluetooth, біля написи "Built-in", далі вибрати вкладку "Config", в ній зліва вибрати Bluetooth і напроти напису "Port Status" поставити галочку. У контролера Bluetooth увімкнено від самого початку.

Далі заходимо в налаштування контролера у вкладку Config, зліва вибираємо Bluetooth (рис 3.8).

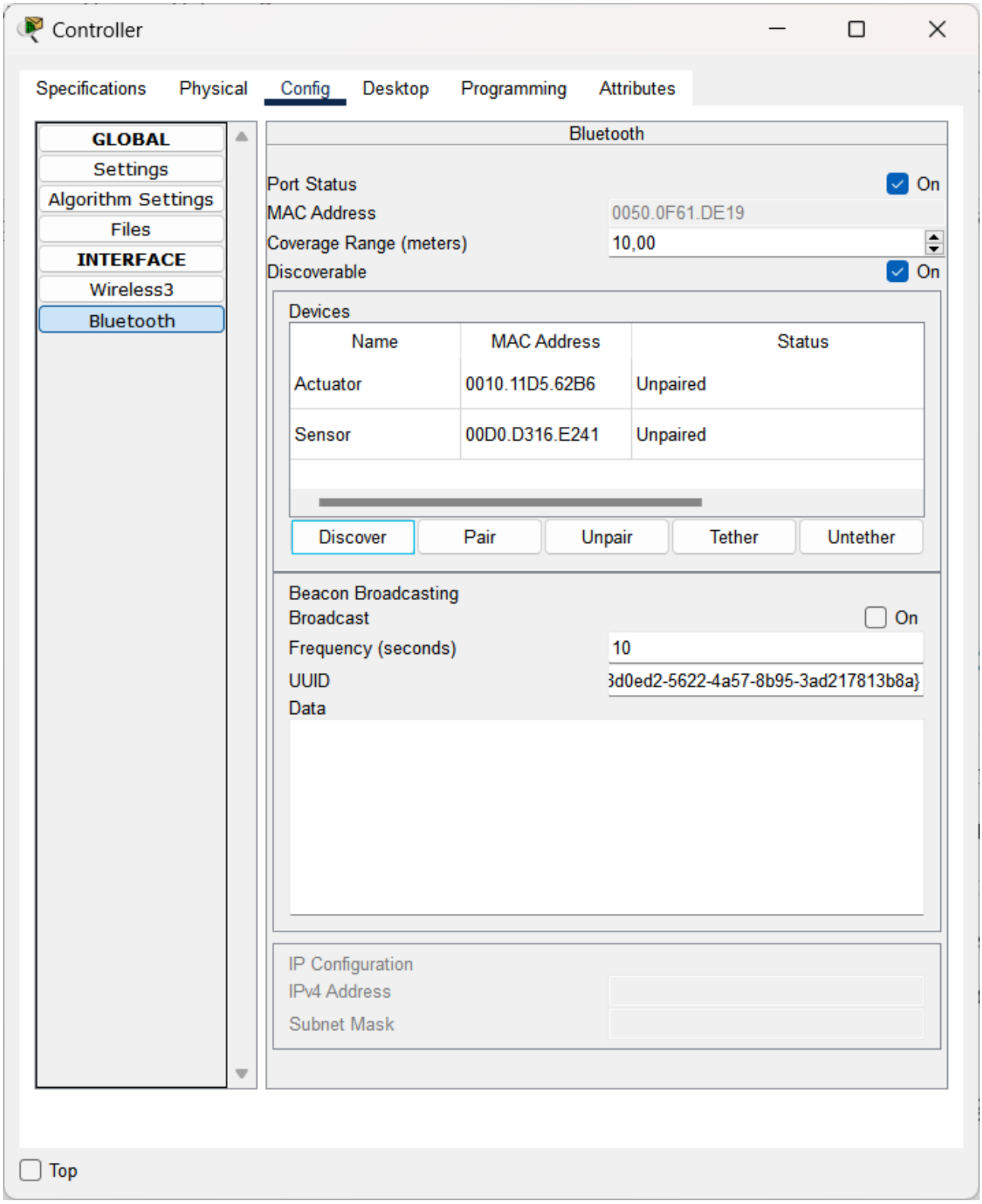

Рис 3.8 – Bluetooth панель контролера

Тут ми можемо побачити MAC адреси та UUID у Bluetooth пристроїв. Якщо натиснути кнопку «Discover» у полі «Devices», можна побачити доступні пристрої. Далі треба на них натиснути і натиснути кнопку Pair (рис 3.9)

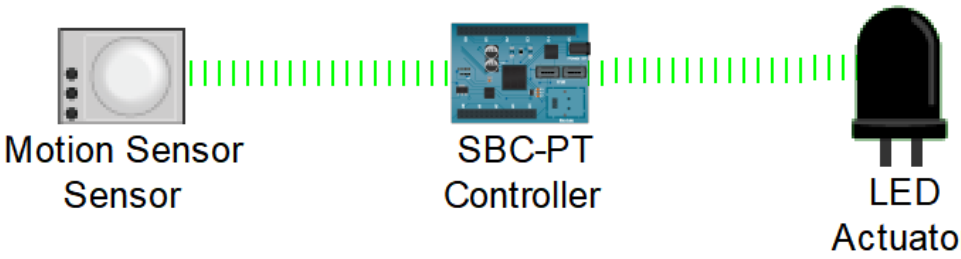

Рис. 3.9 – Елементи, з’єднані за допомогою Bluetooth

Пристрої підключені, однак спочатку вони розраховані на дротову передачу даних, тому їх потрібно перепрограмувати. Почнемо із датчика. Цього разу буде використовуватися мова програмування «Python» [15]. Для простоти всі адреси MAC і UUID будуть записані безпосередньо в коді. На рисунку 3.10 можна побачити, що датчик успішно створює та передає повідомлення контролеру.

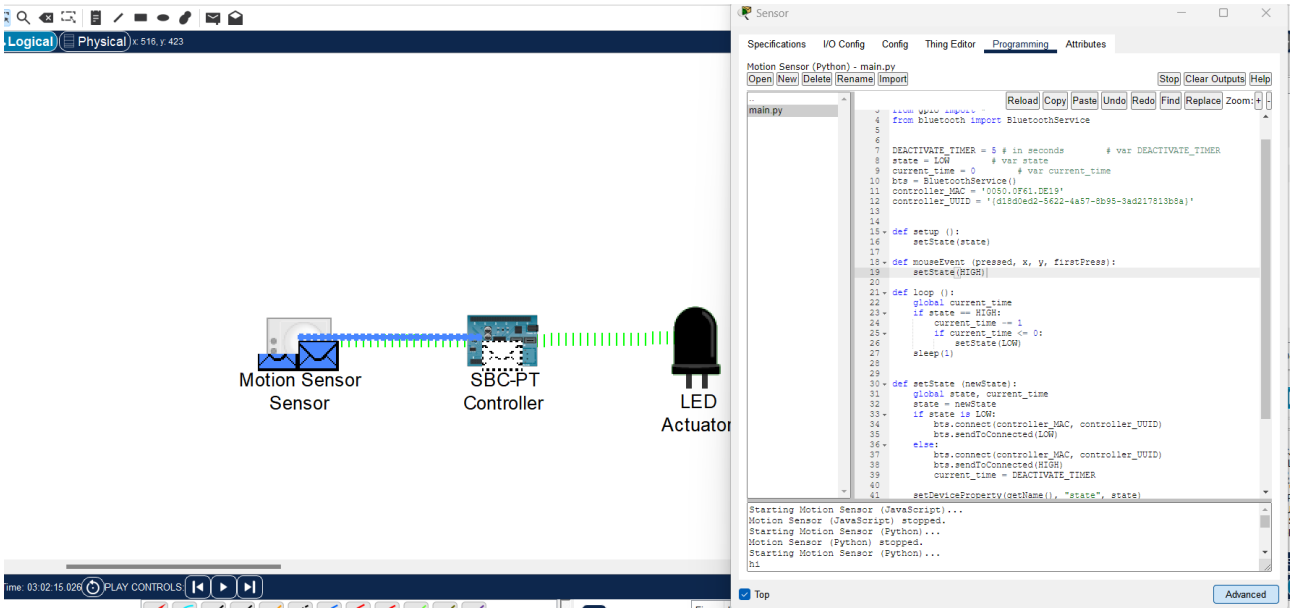

Рис. 3.10 – Передача даних від датчика контролеру

Після цього треба запрограмувати контролер на прийняття даних від датчика руху і на передачу їх лампочці. Якщо подивитися специфікації цих елементів, то можна зрозуміти, що датчик руху передає 1023, якщо засікає рух і протягом декількох секунд після цього, після чого передає 0. Лампочка ж приймає значення

від 0 до 1023, що означають яскравість. Так що можна просто передавати значення, отримане від датчика, лампочці без будь-яких змін (рис. 3.11).

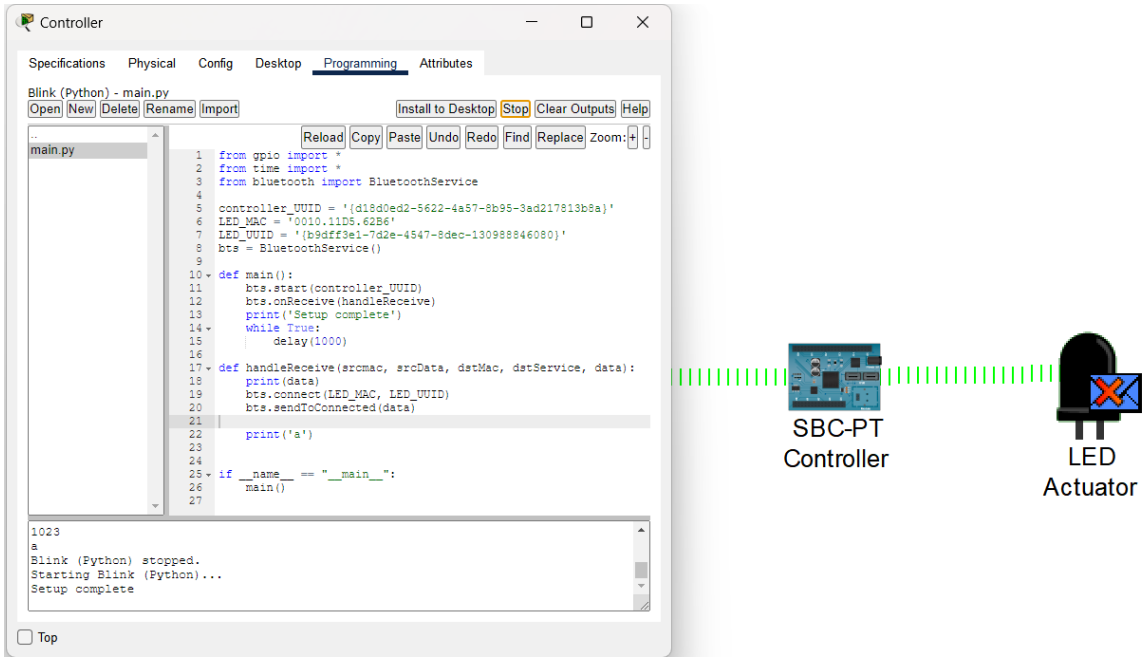

Рис. 3.11 – Лампа не приймає дані

Можна помітити, що контролер передає дані, проте Лампочка їх не приймає. Потрібно налаштувати її для прийняття даних (рис. 3.12).

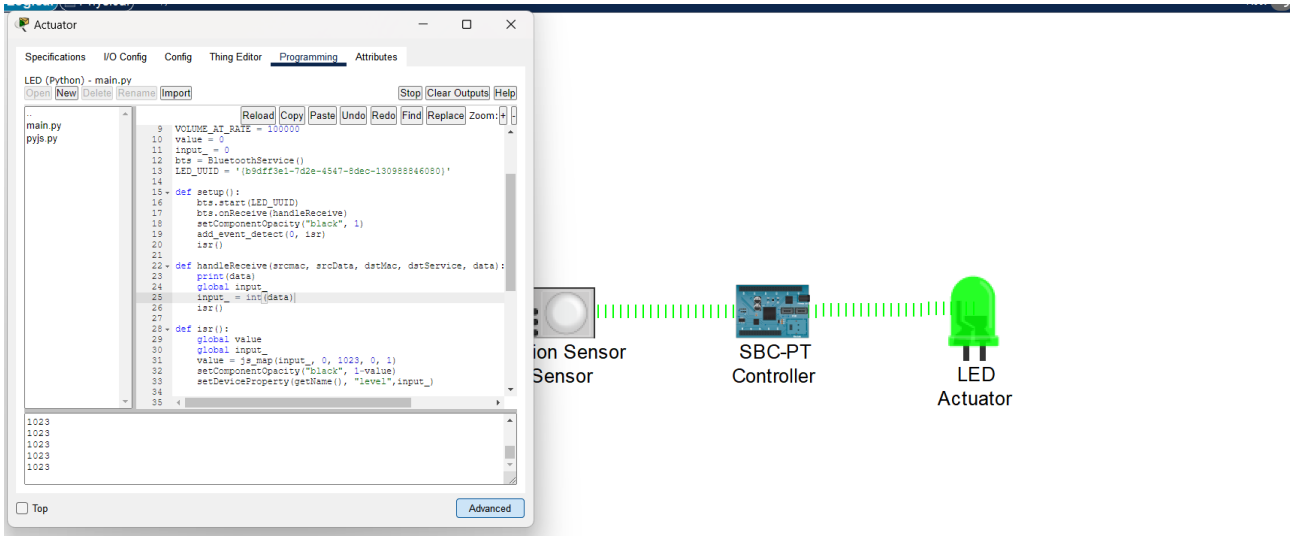

Рис. 3.12 – Робоча мережа IoT на основі Bluetooth

Якщо затиснути alt и поводити курсором біля датчику руху, то лампа загорається

3.3 Бездротова мережа IoT на основі WiFi

Для початку потрібно додати потрібні елементи до робочої області. Це Home Gateway, Humidifer, Humidity monitor та TabletPC-PT.

- Home Gateway – це точка доступу до якої будуть підключені всі пристрої та яка виконуватиме роль сервера інтернету речей.
- Humidifier – зволожувач, буде підвищувати вологість повітря, якщо вона опуститься нижче за належну.
- Humidity monitor – монітор вологості повітря.
- TablePC-PT – планшет. З нього буде керуватися мережею інтернету речей.

Після додавання елементів потрібно натиснути на маршрутизатор у вкладку «Config», обрати вид аутентифікації та ввести додаткові дані, якщо потрібно. На рисунку 3.13 можна побачити встановлення аутентифікації на маршрутизаторі.

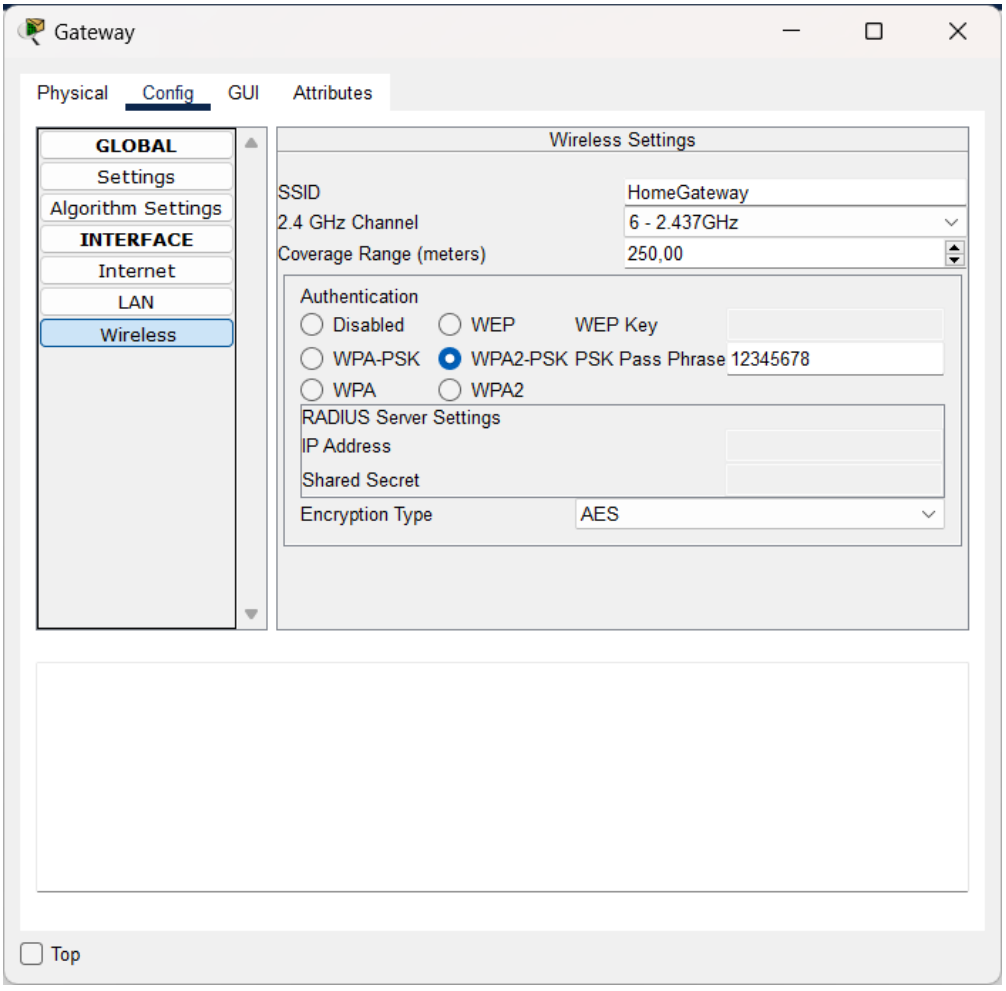

Рис. 3.13 – Налаштування маршрутизатора

Далі треба натиснути на інші елементи мережі, увійти у вкладку Config, після чого натиснути ліворуч на Wireless0, обрати потрібний вид аутентифікації і ввести потрібні дані. Також зліва натиснути на вкладку «Settings» і в полі «IoT Server» поставити Home Gateway.

Далі треба натиснути на планшет, відкрити вкладку Desktop, вибрати IoT Monitor, авторизуватися без зміни даних і відкриється список підключених пристроїв (рис 3.13). Звідси можна дивитися дані датчиків і керувати актуаторами.

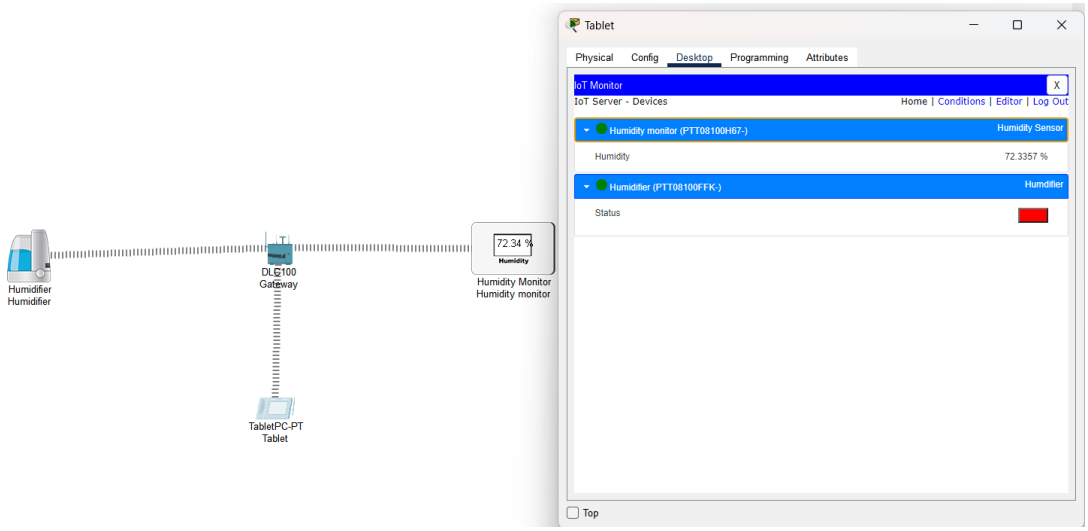

Рис. 3.14 – Пристрої інтернету речей в IoT Monitor

Щоб створювати правила праворуч зверху в IoT Monitor є кнопка Conditions, після натискання на яку відкривається вікно, де можна записувати умови активації. Тут треба натиснути кнопку «Add» і вибрати які значення яких пристроях викликають активацію яких пристроїв. Кожна така умова має мати ім'я. На рисунку 3.14 видно додане правило про включення зволожувача при вологості нижче 50%.

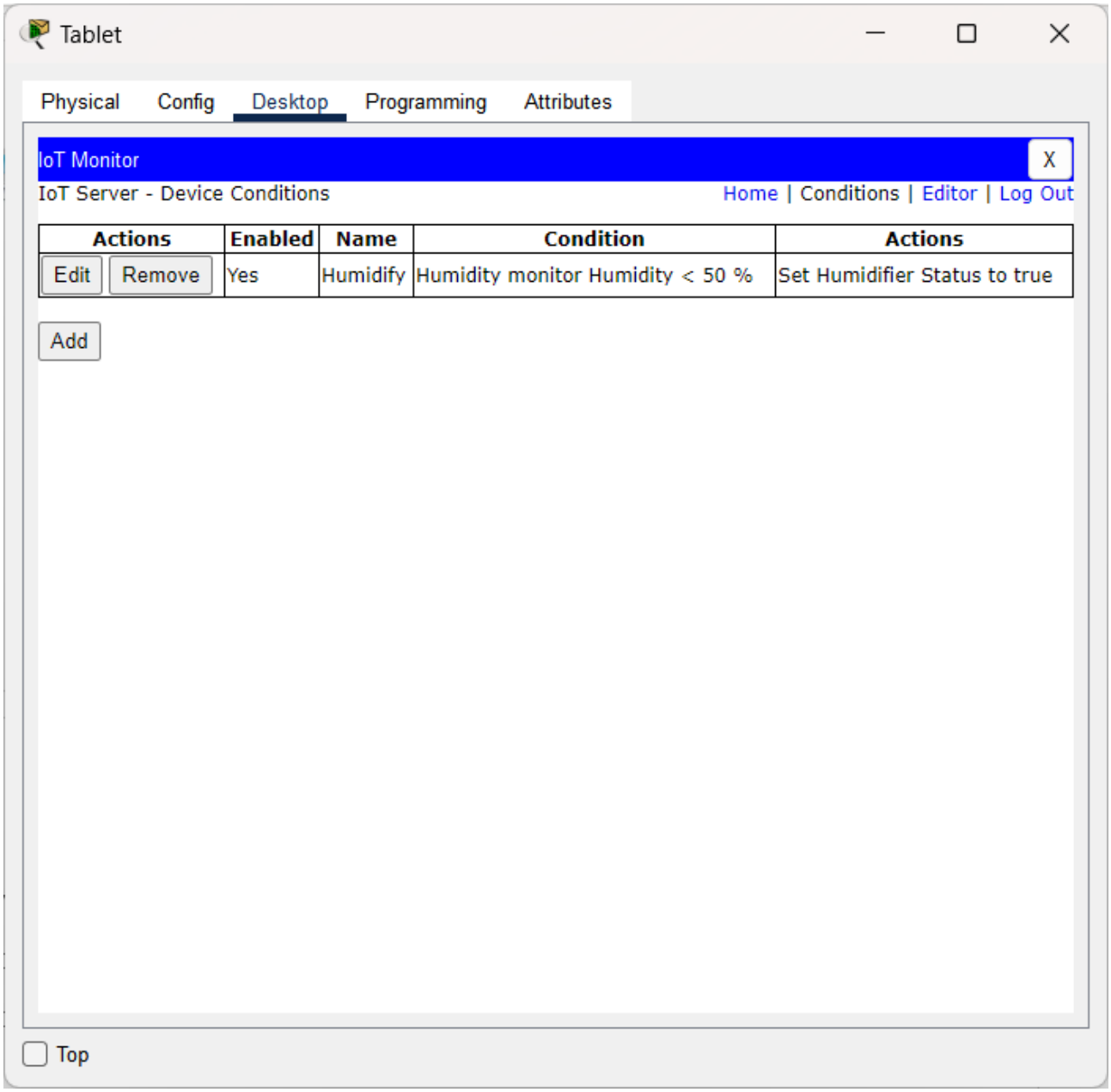

Рис. 3.15 – Вікно створення правил IoT Monitor

Перевіримо це правило, змінивши вологість у приміщенні до 40%. Як видно на рисунку 3.15, у зволожувачі горить червона лампочка, що означає, що він запрацював.

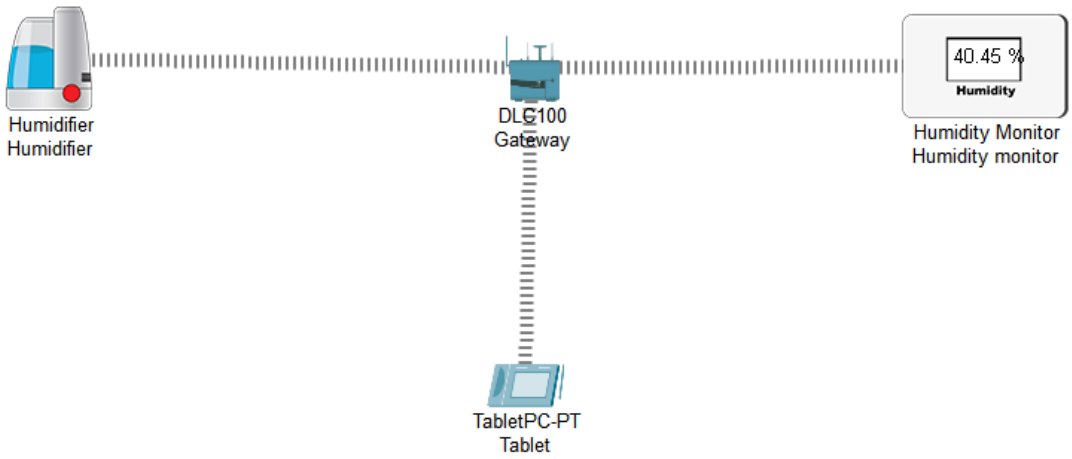

Рис. 3.16 – Робоча мережа IoT на основі Wi-Fi

До мережі можна підключити велику кількість речей, якими можна керувати з планшета, телефону чи іншого пристрою. Також можна додавати багато правил, автоматизуючи процеси. Кількість різних комбінацій речей та правил обмежується лише уявою користувача. Таким чином відбувається побудова бездротової мережі Інтернет речей. Для наочності можна додати більше речей у мережу (рис 3.16).

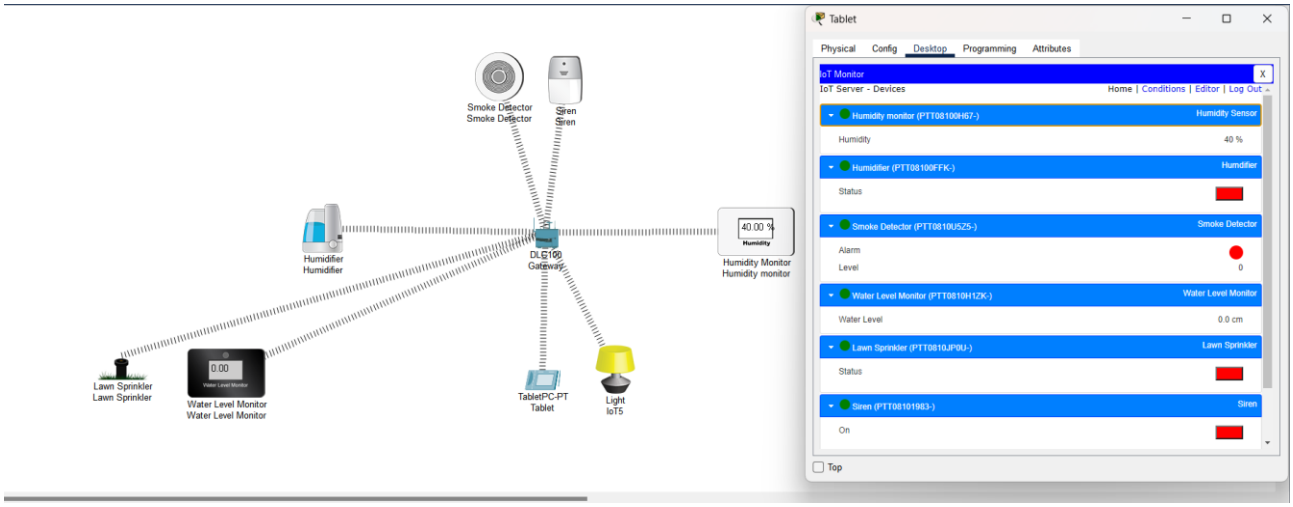

Рис 3.17 – Мережа IoT зі збільшеною кількістю речей

Для додавання нових речей використовують дії, описані раніше.

Висновок до третього розділу

У третьому розділі було розглянуто створення мережі IoT з дротовим зв’зком, Bluetooth зв’язком та Wi-Fi зв’язком в програмному забезпеченні «Packet Tracer». Були розглянуті концепти під’єднання речей та управління їх поведінкою з метою автоматизації. У перших двох випадках використовувалися мікроконтролери, вони були запрограмовані за допомогою мов програмування Python та Javascript, на отримання та передачу даних. У мережі з використанням Wi-Fi маршрутизатор використовувався як IoT-server, до якого приєднувався планшет, з якого проходило керування пристроями в мережі і створювалися правила автоматизації.

ВИСНОВОК

У рамках цієї дипломної роботи було проведено дослідження та створено бездротову мережу Internet of Things (IoT). Основними цілями роботи були дослідження архітектури, побудови інтернету речей, методів бездротової передачі даних у локальних мережах IoT, та захист інформації в них.

Головною метою IoT є забезпечення покращення ефективності, зручності та продуктивності в різних сферах життя. За допомогою сенсорів та датчиків, пристрої IoT можуть збирати дані про навколишнє середовище, стан обладнання, рух, температуру та багато іншого. Ці дані можуть бути використані для аналізу, контролю та прийняття рішень в реальному часі.

Завдяки бездротовій передачі даних, мережа інтернету речей є дуже гнучкою, можна легко додавати та видаляти компоненти мережі, а також взаємодіяти з речами через зручний інтерфейс у будь-якій точці будинку. Але бездротова мережа вразлива для атак. На відміну від дротових мереж, будь-хто може у радіусу дії може прослуховувати дані, які передаються в бездротових мережах. Через це у бездротових мережах дуже багато уваги приділено безпеці.

Були розглянуті протоколи передачі даних, такі, як Bluetooth, Zigbee, 6LoWPAN, 802.11. Було описано принципи їх роботи, види пристроїв в цих мережах, топології мереж, характеристики, такі, як швидкість передачі даних, енергоефективність, дальність роботи та інш. Окремо була розглянута безпека цих протоколів

У третьому розділі була проведена практична робота з побудови мереж інтернету речей з різними видами зв’язку за допомогою програмного забезпечення «Packet Tracer» від компанії «Cisco». Спочатку була розглянута найпростіша дротова локальна мережа IoT, після неї була побудована бездротова мережа на основі Bluetooth, далі була розглянута мережа на основі Wi-Fi.

Таким чином вважаю, що поставлене в бакалаврській роботі завдання виконане, мета досягнута.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Perry L. Internet of Things for Architects / L. Perry – Birmingham: Packt Publishing Ltd, 2018. – 499 p.

2. Gartner Glossary: Internet of Things (iot) [Електроний ресурс] – Режим доступу: World Wide Web. – https://www.gartner.com/en/information-technology/glossary/internet-of-things

3. IoT Explained — How Does an IoT System Actually Work? [Електроний ресурс] – Режим доступу: World Wide Web. – https://medium.com/iotforall/iot-explained-how-does-an-iot-system-actually-work-e90e2c435fe7

4. ITU-T Рекомендація Y.2060 [Електроний ресурс] – Режим доступу: World Wide Web. – https://www.itu.int/rec/T-REC-Y.2060-201206-I

5. Папуловская Н. В. Основы интернета вещей: учебно-методическое пособие / Н. В. Папуловская – Екатеринбург : Издательство Уральского университета, 2022. – 102 с.

6. Banafa A. Introduction to Internet of Things (IoT) / A. Banafa – Gistrup: River Publishers. 2023. – 66 p.

7. Интернет вещей в научных исследованиях [Електроний ресурс] – Режим доступу: World Wide Web. – https://cyberleninka.ru/article/v/internet-veschey-v-nauchnyh-issledovaniyah

8. Tanenbaum A. Computer Networks. Fifth Edition / A. Tanenbaum, D. Wetherall – Boston: Pearson Education, Inc., 2011. – 933 p.

9. Bluetooth Low Energy (BLE): A Complete Guide [Електроний ресурс] – Режим доступу: World Wide Web. – https://novelbits.io/bluetooth-low-energy-ble-complete-guide/

10. Laboid H. Wi-Fi, Bluetooth, Zigbee and WiMax / H. Labiod, H. Afifi, C. De Santis – Dordrecht: Springer, 2007. – 316p.

11. Garg V. Wireless Communications and Networking / V. Garg – San Francisco: Elsevier Inc., 2007. – 930 p.

12. Wi-Fi HaLow (IEEE 802.11ah) — дальнобойное беспроводное подключение с низким энергопотреблением для интернета вещей [Електроний ресурс] – Режим доступу: World Wide Web. – https://www.ixbt.com/news/2016/01/05/wi-fi-halow-ieee-802-11ah.html

13. JavaScript reference [Електроний ресурс] – Режим доступу: World Wide Web. – https://devdocs.io/javascript/

14 Arduino Language Reference [Електроний ресурс] – Режим доступу: World Wide Web. – https://www.arduino.cc/reference/en/

15. The Python Wiki [Електроний ресурс] – Режим доступу: World Wide Web. – https://wiki.python.org/moin/FrontPage

Додаток А

Програма мікроконтролера у підрозділі 3.1:

```
function setup() {
  pinMode(0, INPUT);
  pinMode(1, OUTPUT);
  Serial.println("Setup Complete");
}

function loop() {
  var temperature = digitalRead(0);
  Serial.println(temperature);

  if (temperature >= 640){
     digitalWrite(1, HIGH);
  }
  else{
     digitalWrite(1, LOW);
  }
}
```

Програма датчика у підрозділі 3.2:

```
from time import *
from physical import *
from gpio import *
from bluetooth import BluetoothService


DEACTIVATE_TIMER = 5 # in seconds        # var DEACTIVATE_TIMER
state = LOW        # var state
current_time = 0        # var current_time
bts = BluetoothService()
controller_MAC = '0050.0F61.DE19'
controller_UUID = '{d18d0ed2-5622-4a57-8b95-3ad217813b8a}'


def setup ():
    setState(state)

def mouseEvent (pressed, x, y, firstPress):
    setState(HIGH)

def loop ():
    global current_time
```

```
    if state == HIGH:
        current_time -= 1
        if current_time <= 0:
            setState(LOW)
    sleep(1)


def setState (newState):
    global state, current_time
    state = newState
    if state is LOW:
     bts.connect(controller_MAC, controller_UUID)
     bts.sendToConnected(LOW)
    else:
     bts.connect(controller_MAC, controller_UUID)
     bts.sendToConnected(HIGH)
     current_time = DEACTIVATE_TIMER

    setDeviceProperty(getName(), "state", state)


if __name__ == "__main__":
    setup()
    while True:
        loop()
        sleep(0)
```

Програма контролера у підрозділі 3.2:

```
from gpio import *
from time import *
from bluetooth import BluetoothService

controller_UUID = '{d18d0ed2-5622-4a57-8b95-3ad217813b8a}'
LED_MAC = '0010.11D5.62B6'
LED_UUID = '{b9dff3e1-7d2e-4547-8dec-130988846080}'
bts = BluetoothService()

def main():
  bts.start(controller_UUID)
  bts.onReceive(handleReceive)
  print('Setup complete')
  while True:
     delay(1000)

def handleReceive(srcmac, srcData, dstMac, dstService, data):
  print(data)
  bts.connect(LED_MAC, LED_UUID)
```

```
  #.sendToConnected(data)
  bts.send(LED_MAC, LED_UUID, data)
  print('a')


if __name__ == "__main__":
  main()
```

Програма актуатора у підрозділі 3.2:

```
from environment import *
from physical import *
from gpio import *
from time import *
from pyjs import *
from bluetooth import BluetoothService

MAX_LIGHT_PERCENT = 1
VOLUME_AT_RATE = 100000
value = 0
input_ = 0
bts = BluetoothService()
LED_UUID = '{b9dff3e1-7d2e-4547-8dec-130988846080}'

def setup():
  bts.start(LED_UUID)
  bts.onReceive(handleReceive)
  setComponentOpacity("black", 1)
  add_event_detect(0, isr)
  isr()

def handleReceive(srcmac, srcData, dstMac, dstService, data):
  print(data)
  global input_
  input_ = int(data)
  isr()

def isr():
    global value
    global input_
    value = js_map(input_, 0, 1023, 0, 1)
    setComponentOpacity("black", 1-value)
    setDeviceProperty(getName(), "level",input_)



def main():
    setup()
    while True:
        updateEnvironment()
```

```
            delay(1000)


    def updateEnvironment():
        global value
        global MAX_LIGHT_PERCENT
        global VOLUME_AT_RATE
        rate     =     float(value*MAX_LIGHT_PERCENT*VOLUME_AT_RATE)     /
Environment.getVolume()
        # rate equals limit because we want it to happen immediately
        Environment.setContribution("Visible Light", rate, rate, False)

    if __name__ == "__main__":
        main()
```