

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Радіoeлектроніки і телекомунікацій

(повне найменування інституту, факультету)

Мікро- та нанoeлектроніки

(повне найменування кафедри )

**Пояснювальна записка**

до дипломного проєкту (роботи)

Магістр

(ступінь вищої освіти)

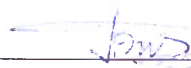
на тему: Впровадження в виробництво системи контролю за станом  
обладнання з використанням IoT

Виконав: студент ІІІ курсу, групи РТ-311м

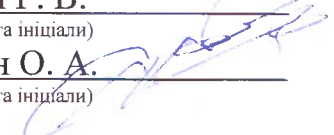
Спеціальності 153 “Мікро- та наносистемна  
техніка”

(код і найменування спеціальності)

Освітня програма (спеціалізація)  
“Мікро- та нанoeлектронні прилади та  
пристрої”

 Пехотін Д. А.  
(прізвище та ініціали)

Керівник  Сніжної Г. В.  
(прізвище та ініціали)

Рецензент  Манукян О. А.  
(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
**Національний університет «Запорізька політехніка»**  
 (повне найменування закладу вищої освіти)

Інститут, факультет Радіoeлектроніки і телекомунікацій  
 Кафедра Мікро- та наноелектроніки  
 Ступінь вищої освіти Магістр  
 Спеціальність 153 “Мікро- та наносистемна техніка”  
 (код і найменування)  
 Освітня програма (спеціалізація) “Мікро- та наноелектронні прилади та пристрої”  
 (назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

мікро- та наноелектроніки

Геннадій СНИЖНОЙ

«10» 12 2022 року

**З А В Д А Н Н Я**  
**НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА**

Пехотіну Дмитру Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Впровадження в виробництво системи контролю за станом обладнання з використанням IoT.

керівник проєкту (роботи) Сніжної Геннадій Валентинович, д.т.н, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «21» листопада 2022 року №393

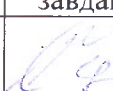
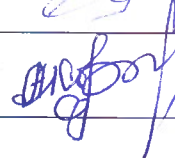
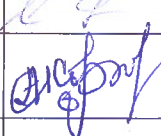
2. Строк подання студентом проєкту (роботи) 05 грудня 2022 р.

3. Вихідні дані до проєкту (роботи) Система керування IoT пристроями Home Assistant, датчики температури і вологості, USB ZigBee Sniffer, програматор CC Debugger, програма Balena Etcher, Telegram бот, програма SmartRF Flash Programmer, одноплатний міні комп'ютер Raspberry Pi 4, мережеве підключення.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Описано сучасні тенденції цифровізації підприємства, встановлена та налаштована система контролю за станом обладнання на базі Home Assistant (додано датчики вимірювання температури та вологості, зроблені автоматизації, налаштована робота попереджень о високих значеннях температури в Telegram бота), проведено моніторинг стану температури після впровадження системи в виробництво.

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

## 6. Консультанти розділів проєкту (роботи)

| Розділ         | Прізвище, ініціали та посада консультанта   | Підпис, дата                                                                        |                                                                                     |
|----------------|---------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|                |                                             | завдання видав                                                                      | прийняв виконане завдання                                                           |
| 1-3            | Сніжної Г. В., д.т.н., завідувач. каф. МІНЕ |  |  |
| Нормо-контроль | Коротун А. В., к.ф.-м.н., доцент каф. МІНЕ  |  |  |
|                |                                             |                                                                                     |                                                                                     |
|                |                                             |                                                                                     |                                                                                     |
|                |                                             |                                                                                     |                                                                                     |
|                |                                             |                                                                                     |                                                                                     |

7. Дата видачі завдання «10» жовтня 2022 року.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту (роботи)                                                     | Строк виконання етапів проєкту (роботи) | Примітка |
|-------|----------------------------------------------------------------------------------------------|-----------------------------------------|----------|
| 1     | Огляд літературних джерел та ознайомлення зі статтями за заданою тематикою.                  | 11.10 – 20.10                           | Виконано |
| 2     | Збирання та прошивка датчика вимірювання температури і вологості.                            | 21.10 – 2.11                            | Виконано |
| 3     | Встановлення та налаштування ОС Home Assistant.                                              | 3.11 – 5.11                             | Виконано |
| 4     | Розробка архітектури системи контролю за станом обладнання та налаштування роботи оповіщень. | 6.11 – 13.11                            | Виконано |
| 5     | Моніторинг стану температури після впровадження системи.                                     | 14.11 – 24.11                           | Виконано |
| 6     | Оформлення пояснювальної записки і презентації.                                              | 25.11 – 05.12                           | Виконано |
|       |                                                                                              |                                         |          |
|       |                                                                                              |                                         |          |
|       |                                                                                              |                                         |          |
|       |                                                                                              |                                         |          |
|       |                                                                                              |                                         |          |

Студент

(підпис)

Дмитро ПЄХОТІН

(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Геннадій СНІЖНОЙ

(прізвище та ініціали)

## АНОТАЦІЯ

ПЗ: 86 стор., 58 рис., 4 табл., 30 джерел.

**Об'єкт:** система контролю за станом обладнання з використанням IoT.

**Мета роботи:** впровадженні в виробництво системи автоматизації та керування IoT пристроями для контролю за станом обладнання роботи персоналу технічного обслуговування.

**Методи:** аналіз і практичне застосування технологічних рішень.

**Результати:** визначено та обґрунтовано архітектуру системи контролю за станом обладнання згідно концепції Industry 4.0.

**Рекомендації щодо впровадження:** результати роботи можуть бути використані при виборі апаратного та програмного забезпечення систем автоматизації та керування IoT пристроями, розробці проектних рішень систем для моніторингу.

**Практична цінність:** розроблена архітектура системи на базі ОС Home Assistant для автоматизації та керування IoT пристроями, застосування системи можливе з будь яким обладнання в якому є потреба контролю за станом показників таких як температура, робота ефективності системи перевірена.

## ЗМІСТ

|                                                                                                          |    |
|----------------------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....                                                                           | 6  |
| ВСТУП.....                                                                                               | 7  |
| 1 ЦИФРОВІЗАЦІЯ СУЧАСНОГО ПІДПРИЄМСТВА, ОСНОВНІ<br>ПРИНЦИПИ ІНДУСТРІЇ 4.0.....                            | 8  |
| 1.1 Індустрія 4.0 та роль IoT в системі управління підприємством.....                                    | 8  |
| 1.2 Стандарт бездротового зв'язку ZigBee.....                                                            | 14 |
| 1.3 Система керування IoT пристроями та їхньої автоматизації Home<br>Assistant.....                      | 33 |
| 2 ВПРОВАДЖЕННЯ В ВИРОБНИЦТВО СИСТЕМИ КОНТРОЛЮ ЗА<br>СТАНОМ ОБЛАДНАННЯ.....                               | 37 |
| 2.1 Обладнання яке потребує систему контролю.....                                                        | 37 |
| 2.2 Датчик температури і вологості.....                                                                  | 39 |
| 2.3 Встановлення системи контролю за станом обладнання на базі Home<br>Assistant.....                    | 50 |
| 2.4 Налаштування системи Home Assistant, додавання датчиків для<br>моніторингу за станом обладнання..... | 53 |
| 2.5 Налаштування роботи попереджень о високих значеннях<br>температури.....                              | 66 |
| 2.6 Моніторинг стану температури після впровадження системи в<br>виробництво.....                        | 73 |
| 3 УДОСКОНАЛЕННЯ СИСТЕМИ КОНТРОЛЮ ЗА СТАНОМ<br>ОБЛАДНАННЯ.....                                            | 77 |
| 3.1 Віддалений доступ до системи контролю із мережі інтернету.....                                       | 77 |
| 3.2 Перспективи розвитку платформи.....                                                                  | 80 |
| ВИСНОВКИ.....                                                                                            | 82 |
| ПЕРЕЛІК ПОСИЛАНЬ.....                                                                                    | 84 |

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

IoT – Internet of Things;

IIoT – Industrial Internet of Things;

AODV – Ad hoc On Demand Distance Vector;

H.A. – Home Assistant;

PLC – Programmable logic controller;

MQTT – Message Queue Telemetry Transport;

API – Application Programming Interface.

## ВСТУП

Індустрія 4.0 – провідний тренд «Четвертої промислової революції», яка відбувається на наших очах.

Виділяють чотири базові технології, в результаті впровадження яких очікуються революційні зміни, одна з яких – інтернет речей (Internet of Things, IoT). У цій технології інтернет використовується для обміну інформацією не тільки між людьми, але і між різними «речами», тобто машинами, пристроями, датчиками тощо. Починається все з того, що ключові компоненти обладнання забезпечуються різними датчиками, виконавчими механізмами і контролерами. Зібрані дані обробляються і надсилаються до відповідних служб підприємства, що дозволяє персоналу оперативно приймати обґрунтовані і виважені рішення.

Сучасні темпи розвитку ІТ технологій змусили задуматися о впровадженні в виробництво системи для контролю за станом обладнання та роботи персоналу технічного обслуговування.

Для рішення цього завдання запропоновано впровадження системи автоматизації та керування IoT пристроями, яка працюватиме на базі ОС Home Assistant та застосування стандарту бездротового зв'язку ZigBee.

Для роботи системи були використанні датчики виміру температури та вологості, одноплатний міні комп'ютер Raspberry Pi 4 для встановлення ОС Home Assistant та роботи USB ZigBee Sniffer, який буде використано як шлюз конвертера ZigBee в MQTT протокол.

Головною складовою впровадженої системи контролю є:

- автоматизація всіх процесів системи, комутація IoT пристроїв в єдиний центр контролю;
- надсилання оповіщень про високі показники з використання сучасних технологій автоматичної обробки і доставки повідомлень;
- запис показників до журналу подій для подальшої інспекції роботи відповідального персоналу виробництва.

# **1 ЦИФРОВІЗАЦІЯ СУЧАСНОГО ПІДПРИЄМСТВА, ОСНОВНІ ПРИНЦИПИ ІНДУСТРІЇ 4.0**

## **1.1 Індустрія 4.0 та роль IoT в системі управління підприємством**

Індустрія 4.0 – це сучасний підхід до процедури виробництва, в основі якого лежить масова інтеграція інформаційних технологій у промисловість.

Зараз ми живемо в епоху завершення третьої цифрової революції, що почалася в другій половині минулого століття. Її характерні риси — розвиток інформаційно-комунікаційних технологій, автоматизація та роботизація виробничих процесів [1].

Характерні риси Індустрії 4.0 – це повністю автоматизовані виробництва, на яких керівництво всіма процесами здійснюється в режимі реального часу. Кіберфізичні системи створюють віртуальні копії об'єктів фізичного світу, контролюють фізичні процеси і приймають децентралізовані рішення. Вони здатні об'єднуватися в одну мережу, взаємодіяти між собою, самоналагоджуватися і самонавчатися. Важливу роль відіграють інтернет-технології, що забезпечують комунікації між машинами та персоналом. Підприємства виробляють продукцію відповідну до вимог індивідуального замовника, оптимізуючи собівартість виробництва.

Виділяють чотири базові технології, в результаті впровадження яких очікуються революційні зміни [4].

1. Інтернет речей (Internet of Things, IoT). У цій технології Інтернет використовується для обміну інформацією не тільки між людьми, але і між різними «речами», тобто машинами, пристроями, датчиками і т.д. З одного боку пристрої, забезпечені датчиками, можуть обмінюватися даними і обробляти їх без участі людини. З іншого боку, людина може активно брати участь в цьому процесі, наприклад, коли мова йде про «розумний будинок».

Різновидом IoT є промисловий (індустріальний) інтернет речей (Industrial Internet of Things, IIoT). Саме він відкриває пряму дорогу до

створення повністю автоматизованих виробництв. Починається все з того, що ключові компоненти обладнання забезпечуються різними датчиками, виконавчими механізмами і контролерами. Зібрані дані обробляються і надсилаються до відповідних служб підприємства, що дозволяє персоналу оперативно приймати обґрунтовані і виважені рішення. Але завдання-максимум полягає в досягненні такого рівня автоматизації підприємства, при якому на всіх ділянках, де це можливо, машини працюють без участі людей. Роль персоналу при цьому зводиться до контролю роботи машин і реагування лише на екстрені ситуації.

2. Цифрові екосистеми. Це системи, що складаються з різних фізичних об'єктів, програмних систем і керуючих контролерів, що дозволяють уявити таке утворення як єдине ціле. Фізичні та обчислювальні ресурси в такий екосистемі тісно пов'язані, моніторинг і управління фізичними процесами здійснюється з використанням технологій IoT. Традиційні інженерні моделі гармонійно співіснують з комп'ютерними.
3. Аналітика великих даних (Data Driven Decision) або просто Великі дані (Big data). Величезні обсяги інформації, що накопичуються в результаті «оцифровування» фізичного світу, можуть бути ефективно оброблені тільки комп'ютерами (в майбутньому, можливо, квантовими), із застосуванням хмарних обчислень і технологій штучного інтелекту (Artificial Intelligence). В результаті людина, яка контролює той чи інший процес, ситуацію, обстановку має отримувати оброблені дані, максимально зручні для сприйняття, аналізу і ухвалення рішення.
4. Складні інформаційні системи, відкриті для використання клієнтами і партнерами (цифрові платформи). Це можуть бути цифрові платформи і системи для управління бізнес-процесами, для інтеграції інтернету речей в фізичні бізнес-процеси, для аналізу і прогнозування стану обладнання, що детально буде розглядатися в даному дипломному проекті.

Четверта промислова революція, крім перерахованих вище сфер прискореного розвитку, може також задіяти широке впровадження 3D-друку, друкованої електроніки, застосування розподілених реєстрів (тобто технології блокчейн, яка стала популярною після створення на її основі криптовалют), використання віртуальної і доповненої реальності і навіть розробку автономних роботів, які будуть не компонентами автоматизованих ліній, як зараз, а цілком мобільними високоінтелектуальними пристроями, здатними працювати поруч з людьми [6].

За прогнозами Всесвітнього Економічного Форуму, більшість технологій Четвертої революції стане повсякденністю вже в 2027 році. А це означає, що з'являться не тільки розумні будинки, а й розумні міста, безпілотні автомобілі на вулицях, штучний інтелект в офісах і суперкомп'ютери в кишенях.

Вперше про програму «Індустрія 4.0» мова зайшла у 2011 році на промисловій виставці в Ганновері, де уряд Німеччини поставив задачу розширити застосування інформаційних технологій у виробництві. Над створенням програми модернізації промислових підприємств країни в цьому напрямку працювала високопрофесійна команда, до якої увійшли представники бізнесу і держави. Мета програми – збереження і збільшення конкурентних переваг підприємств країни. Зацікавленість в програмі «Індустрія 4.0» і уряду, і бізнесу Німеччини легко зрозуміти. Наприклад, за оцінками консалтингової компанії Roland Berger, економіка ЄС може недоотримати в найближчі роки 605 млрд. доларів, якщо проігнорує вимоги, що висуваються Четвертої промисловою революцією. А ось у разі їх виконання потенційний прибуток може досягти 1,25 трильйонів доларів. Не дивно, що за прикладом Німеччини слідують і інші країни. В Японії створено Національний інститут просування цифрової економіки і цифрового суспільства (Japan Institute for Promotion of Digital Economy and Community, JIPDEC).

Найбільші компанії США – AT&T, Cisco, GE, IBM і Intel — в 2014 році створили Консорціум промислового Інтернету (Industrial Internet Consortium™, ІІС), відкриту некомерційну групу, яка за станом на початок 2017 року об'єднувала вже 250 компаній з 30 країн. Основне завдання Консорціуму - створення екосистеми компаній, наукових центрів і державних структур, сприятливою для впровадження індустриального Інтернету. Згідно з прогнозом компанії McKinsey, до 2025 року сукупний економічний ефект від впровадження тільки промислового інтернету складе до 11 трлн. доларів на рік. Значить, ті компанії, які вже сьогодні беруть активну участь у Четвертій промислової революції, отримають відчутні конкурентні переваги вже завтра.

На даний час перехід до індустрії 4.0 в Україні відбувається дуже активно. Створено рух «Індустрія 4.0 в Україні», велику увагу цим питанням приділяє АППА (Асоціація підприємств промислової автоматизації України).

Складовою частиною Інтернету речей є Індустриальний (або Промисловий) Інтернет речей (Industrial Internet of Things, IIoT) [2].

Сучасний Інтернет складається з тисяч корпоративних, наукових, урядових та домашніх комп'ютерних мереж. Об'єднання мереж різної архітектури і топології здійснюється за допомогою протоколу ІР. Кожному учаснику мережі (або групі учасників) присвоюється ІР-адреса, постійний або тимчасовий (динамічний).

Аналогічним чином Інтернет речей сьогодні складається з безлічі слабо пов'язаних між собою мереж, кожна з яких вирішує свої завдання. Наприклад, в офісній будівлі може бути розгорнуто відразу кілька мереж: для управління кондиціонерами, системою опалення, освітленням, безпекою і т.д. Ці мережі можуть працювати за різними стандартами, і об'єднання їх в одну мережу представляє собою нетривіальну задачу. Крім того, існуюча (четверта) версія протоколу ІР (IPv4) дозволяє використовувати всього лише 4,22 мільярда адрес, через що виникла проблема їх вичерпання. І хоча не кожному апарату, який підключається до мережі, необхідний унікальний ІР-адреса (але все одно необхідний унікальний ідентифікатор), в зв'язку з бурхливим зростанням

Інтернету речей проблема нестачі адрес може стати обмежуючим фактором. Кардинально вирішити її допоможе шоста версія протоколу, IPv6 яка забезпечить можливість використання кожним жителем Землі більше 300 млн. IP-адрес.

Очікується, що до 2023 року в світі буде від 30 до 50 млрд. об'єднаних в мережу речей, а можливості адресації протоколу IPv6 дозволять практично без обмежень ідентифікувати в Мережі будь-яку річ.

В основі Інтернету речей лежать такі технології:

1. Засоби ідентифікації.

Кожен об'єкт фізичного світу, який бере участь в Інтернеті речей, нехай навіть не підключений до Мережі, все одно повинен мати унікальний ідентифікатор. Для автоматичної ідентифікації предметів можуть використовуватися різні вже існуючі системи: радіочастотна, при використанні якої до кожного об'єкту прикріплюється радіочастотна мітка, оптична (штрих-коди, Data Matrix, QR-коди), інфрачервоні мітки тощо. Але для забезпечення унікальності ідентифікаторів різних типів доведеться провести роботу по їх стандартизації.

2. Засоби вимірювання.

Завдання засобів вимірювання – забезпечити перетворення інформації про зовнішнє середовище в дані, придатні для передачі їх засобам обробки. Це можуть бути як окремі датчики температури, освітленості тощо, так і складні вимірювальні комплекси. Для досягнення автономності засобів вимірювання бажано забезпечити електроживлення датчиків за рахунок засобів альтернативної енергетики (сонячні батареї тощо), щоб не витрачати час і кошти на підзарядку акумуляторів або заміну батарей.

3. Засоби передачі даних.

Для передачі даних може бути використана будь-яка з існуючих технологій. У разі використання бездротових мереж особливу увагу приділяють підвищенню надійності передачі даних. При використанні дротових мереж активно використовують технологію передачі даних по лініях

електропередач, оскільки багато «речей» (такі як торгові автомати, банкомати тощо) підключені до електромереж.

#### 4. Засоби обробки даних.

Тридцять і більше мільярдів пристроїв, підключені в 2020 році до Інтернету, генерують 44 мільярди терабайт даних. Це приблизно в сім разів перевищує кількість оцифрованої інформації в усьому світі станом на 2010-ті роки. Тому в компанії Microsoft вважають, що головна частина Інтернету речей – це не датчики і засоби передачі даних, а хмарні системи, що забезпечують високу пропускну здатність і здатні швидко реагувати на певні ситуації (наприклад, вміти за показаннями датчиків з'ясувати, що в будинку вже п'ять хвилин нікого немає, а входні двері залишилися відкритими). Допоможуть впоратися з величезними потоками інформації також туманні обчислення, які будуть не конкурувати з хмарними, а ефективно їх доповнювати.

#### 5. Виконуючі пристрої.

Це пристрої, здатні перетворювати цифрові електричні сигнали, що надходять від інформаційних мереж, в дії. Наприклад, для того, щоб через смартфон можна було включити систему опалення в будинку, вона повинна мати відповідний пристрій. Виконавчі пристрої часто конструктивно поєднуються з датчиками [8].

Передбачається, що до 2022 року інтернет речей буде застосовуватися в самих різних галузях. Перш за все це промисловість, транспорт (220 млн. підключених автомобілів), розумний будинок, комунальні служби (мільярд датчиків, істотне зниження втрат енергії), охорону здоров'я (646 млн. пристроїв, які збирають дані про здоров'я людей), аграрний сектор (75 млн. датчиків для моніторингу стану ґрунту). Крім того, Інтернет речей буде застосовуватися в торгівлі, логістиці, громадському харчуванні, готельному бізнесі, банківській системі, будівництві та в збройних силах (126 тис. дронів і роботів).

У віддаленій перспективі «розумними» стануть не тільки будинки, а й міста, і навіть (деякі) держави. Але на даному етапі розвитку технологій і суспільства Інтернет речей активно впроваджується не в глобальних масштабах, а всередині компаній, що займаються виробництвом товарів, енергії, транспортними перевезеннями тощо — там, де за рахунок нових технологій очікується підвищення продуктивності та конкурентоспроможності. Складність масштабування цього досвіду обумовлена тим, що необхідно інтегрувати в єдине ціле багато систем від різних постачальників, а налагодити їх злагоджену роботу не просте завдання.

## **1.2 Стандарт бездротового зв'язку ZigBee**

ZigBee – це відкритий стандарт бездротового зв'язку для різних систем автоматизації: «Розумний дім», «Інтелектуальна будівля», системи управління технологічними та бізнес-процесами, системи безпеки тощо.

Технологія ZigBee дозволяє створювати бездротові мережі, що самоорганізуються і самовідновлюються, з автоматичною ретрансляцією повідомлень. Мережі ZigBee при відносно невеликих швидкостях передачі забезпечують гарантовану доставку пакетів і захист інформації, що передається.

Стандарт ZigBee передбачає використання частотних каналів у діапазонах 868 МГц, 915 МГц та 2,4 ГГц. Найбільші швидкості передачі даних і найвища стійкість до перешкод досягаються в діапазоні 2,4 ГГц. Тому більшість виробників мікросхем випускають приймач саме для цього діапазону, в якому передбачено 16 частотних каналів з кроком 5 МГц.

Швидкість передачі разом зі службовою інформацією в ефірі становить 250 кбіт/с. При цьому середня пропускну здатність вузла для корисних даних, залежно від завантаженості мережі та кількості ретрансляцій, може лежати в межах 5...40 кбіт/с. Відстань між вузлами мережі становлять десятки метрів

при роботі всередині приміщень та сотні метрів на відкритому повітрі. За рахунок ретрансляцій зона покриття мережі може значно збільшуватись.

Дещо про історію створення стандарту ZigBee.

У різних галузях є потреба у створенні бездротових мереж із великою кількістю датчиків і виконавчих механізмів, де потрібно високих швидкостей передачі, але на першому плані виступають надійність, стійкість (здатність до самовідновлення) і простота розгортання та експлуатації.

Найважливішими вимогами до таких мереж є також:

- тривала робота джерел живлення датчиків;
- низька вартість;
- компактність;
- можливість створення пористих мереж для зв'язку між великою кількістю пристроїв.

Таке поєднання вимог призвело до створення стандартів IEEE 802.15.4 і ZigBee, що забезпечують прості, стійкі, масштабовані, багатокрокові бездротові мережі з можливістю підтримки різних додатків. Альянс ZigBee, заснований у 2002 році, є спільнотою компаній (понад 300), що об'єдналися з метою розробки ефективних протоколів для бездротової мережі та забезпечення сумісності пристроїв різних виробників. У жовтні 2007 року було прийнято альянсом специфікацію ZigBee® Pro Feature Set 2006, вона надає деякі переваги порівняно з першою специфікацією, тому вона є основною на сьогодні. Наведені нижче описи належать до цієї версії стандарту.

Переваги застосовуваних мережевих рішень:

- низька ціна;
- безпека;
- надійність та самовідновлення;
- гнучкість;
- практично не обмежена можливість розширення;
- низьке енергоспоживання;

- простота та дешевизна розгортання;
- вбудований алгоритм налаштування мережі та маршрутизації.

Стандарти IEEE 802.15.4 та ZigBee 2007 Specification визначають стандартизовані протоколи, які забезпечують мережну інфраструктуру, необхідну для бездротових мережних додатків з великою кількістю датчиків та виконавчих механізмів.

IEEE 802.15.4 визначає фізичний та MAC рівні, а ZigBee визначає мережевий рівень та рівень додатків.

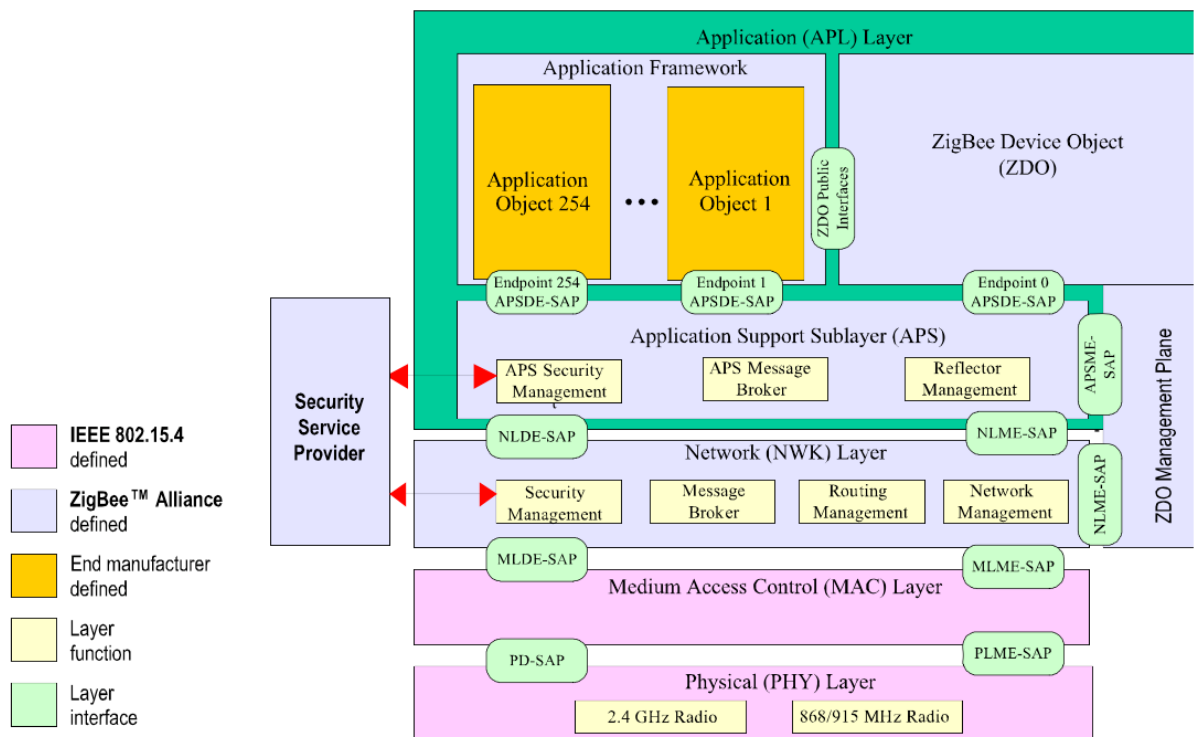


Рисунок 1.1 – Стек протоколів ZigBee

Специфікація ZigBee регламентує стек протоколів взаємодії вузлів мережі, в якому протоколи верхніх рівнів використовують послуги, що надаються протоколами нижніх рівнів [7].

Рівень програм (APL) складається з фреймворка програм (Application Framework), об'єкта пристрою ZigBee (ZD0) та підрівня підтримки програм (APS).

Фреймворк додатків (Application Framework) містить опис порядку створення профілів у стеку ZigBee, що гарантує їхню сумісність, а також визначає:

- стандартні типи даних для профілів;
- дескриптори, які допомагають у виявленні служб;
- формати кадрів для транспортування даних та значення пар ключів.

Це дозволяє швидко розробляти прості профілі на основі атрибутів.

Об'єкти програм (Application Objects) – програмні модулі, що управляють пристроями ZigBee у кінцевих точках. Одна базова станція ZigBee може підтримувати до 240 об'єктів програм. Об'єкти програм підтримують кінцеві точки, пронумеровані від 1 до 240. Кінцева точка 0 зарезервована для об'єкта пристрою ZigBee (ZD0).

Об'єкт пристрою ZigBee (ZigBee Device Object – ZD0):

- визначає роль пристрою ZigBee: координатор, маршрутизатор або кінцевий пристрій;
- ініціює запити пошуку та приєднання та/або відповідає на такі запити;
- встановлює безпечний зв'язок між пристроями ZigBee.

Він також надає багатий набір команд керування, які визначаються у профілі пристрою ZigBee (використовуються при введенні в експлуатацію).

План управління ZD0 (ZD0 Management Plane) підтримує зв'язок ZD0 з підрівнем підтримки програм (APS) та мережевим рівнем (NWK). Дозволяє ZD0 обробляти запити додатків на доступ до мережі та забезпечення безпеки, використовуючи повідомлення профілю пристрою ZigBee (ZigBee Device Profile – ZDP).

Підрівень підтримки програм (Application Support Sublayer – APS) відповідає за надання даних програмам та профілям пристрою ZigBee. Підрівень також керує приєднаннями в мережі ZigBee та зберігає дані про такі приєднання у своїй таблиці.

Постачальник послуг безпеки (SSP) забезпечує механізми безпеки для рівнів, що використовують шифрування – NWK і APS. Ініціюється та конфігурується об'єктом пристрою ZD0.

Мережевий рівень (Network Layer – NWK) обробляє мережеві адреси та маршрутизацію за викликами MAC рівня. До його завдань також входять:

- запуск мережі (якщо пристрій є координатором);
- присвоєння мережевих адрес;
- додавання та видалення мережевих пристроїв;
- маршрутизація повідомлень;
- застосування політики безпеки;
- здійснення пошуку маршрутів.

Рівні, що визначаються стандартом IEEE 802.15.4:

Рівень керування доступом до середовища (Medium Access Control Layer – MAC) відповідає за надійний зв'язок пристрою з безпосередніми сусідами, допомагає вирішувати колізії та підвищувати ефективність. MAC рівень відповідає також за асемблювання та декомпозицію кадрів та пакетів даних. Фізичний рівень (Physical Layer – PHY) забезпечує інтерфейс до фізичного середовища передачі.

Фізичний рівень складається з двох рівнів, що працюють у різних діапазонах частот. Рівень, що працює за менших частот, покриває діапазони 868 МГц для Європи та 915 МГц для таких країн, як США та Австралія. Високочастотний фізичний рівень працює на частоті 2,4 ГГц та застосовується практично у всьому світі.

Мережі ZigBee включають такі типи пристроїв (базових станцій):

- координатори;
- маршрутизатори;
- кінцеві пристрої.

Координатор - цей пристрій запускає мережу та керує нею. Координатор формує мережу, є центром керування мережі та довірчим центром (trust-

центром) – встановлює політику безпеки та задає налаштування під час підключення пристрою до мережі, відповідає за ключі безпеки.

Маршрутизатор розширює область покриття мережі, здійснює динамічну маршрутизацію в обхід перешкод, відновлює маршрути у випадках навантаження мережі або відмови будь-якого пристрою. Маршрутизатори можуть приєднуватися до координатора та інших маршрутизаторів, а також підтримувати дочірні пристрої. Маршрутизатори здійснюють маршрутизацію пакетів по мережі та повинні бути готові до передачі даних у будь-який момент часу. Тому ці вузли не використовують режими зниженого енергоспоживання та мають стаціонарне живлення. Їхня кількість у мережі має бути достатньою для обслуговування необхідної кількості сплячих вузлів. Максимальна кількість сплячих вузлів, які обслуговує маршрутизатор – 32.

Кінцевий пристрій може приймати та надсилати повідомлення, але не може транслювати пакети та здійснювати маршрутизацію. Кінцеві пристрої підключаються до маршрутизатора або координатора і не можуть підтримувати дочірні пристрої.

Як особливий тип пристроїв, у мережах ZigBee можна виділити так звані "сплячі пристрої". По суті, це кінцеві пристрої, переведені в сплячий режим.

Сплячі пристрої використовують режими зниженого енергоспоживання. Як правило, це вузли з батарейним живленням. Їхня кількість визначається конкретним додатком. На час сну маршрутизатори, до яких ці пристрої підключені, представляють їх у мережі.

Стандарт ZigBee описує профіль стека, який визначає мережу, служби програм та параметри безпеки для всієї мережі. Профіль стека для конкретної ZigBee мережі вибирається мережевим адміністратором (проектувальником), виходячи з сфери застосування (автоматизація будівлі, управління техпроцесом, позиціонування). Кожен такий профіль стека потім визначатиме мережеві параметри, такі як:

- загальні розміри мережі;
- алгоритми маршрутизації, що підтримуються;

- розміри таблиць маршрутизації та прив'язки додатків;
- характер безпеки для цієї мережі.

Особливий інтерес з точки зору топології мережі представляє специфікація трьох параметрів: максимальна глибина мережі (`nwkMaxDepth`), максимальна кількість дочірніх зв'язків маршрутизатора (`nwkMaxChildren`) та максимальна кількість зв'язків з дочірніми маршрутизаторами (`nwkMaxRouters`). Разом вони визначають загальну структуру мережі.

Відповідно до стандарту IEEE 802.15.4, кожен мережевий пристрій має 64-розрядну (довгу) IEEE адресу, яка унікально ідентифікує цей пристрій. Щоб продовжити життя акумуляторних батарей, для скорочення розмірів пакетів і відповідно часу активної взаємодії пристроїв використовуються більш короткі адреси. Стандарт ZigBee вимагає, щоб обмін повідомленнями після приєднання до мережі здійснювався з 16-бітовими (короткими) мережевими адресами. 16-бітова мережна адреса присвоюється пристрою, що приєднуються знову під час обміну повідомленнями в процесі приєднання.

Координатор ZigBee завжди має адресу 0000. Координатор за встановленим стандартом ZigBee алгоритмом визначає, виходячи з параметрів профілю стека, діапазони адрес, які присвоюються пристрою, що приєднуються відповідно до їх розташування в мережевій ієрархії. Хоча ці діапазони адрес різняться, залежно від параметрів профілю стека, діапазони адрес завжди однакові для конкретного набору параметрів профілю стека.

Однією з важливих особливостей мереж ZigBee є можливість вибору топології мережі, що найбільше відповідає конкретному застосуванню. Точна структура мережі ZigBee в більшості випадків не може бути наперед визначена, оскільки залежить від взаємного розташування пристроїв і проходження радіохвиль під час формування мережі. Однак три згадані параметри профілю стека диктуватимуть загальну структуру (топологію) мережі.

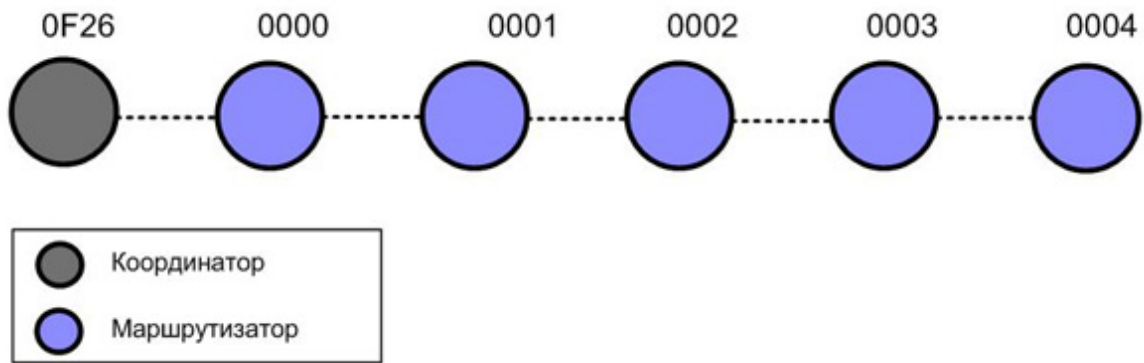


Рисунок 1.2 – Лінійна топологія мережі

Найбільш перспективною у мережі ZigBee є сітчаста топологія (mesh-топологія).

Сітчаста мережа – це мережа взаємопов'язаних маршрутизаторів і кінцевих пристроїв, в якій кожен маршрутизатор має принаймні два зв'язки і може транслювати повідомлення для сусідніх пристроїв. Як показано на рис. 1.3, сітчаста мережа складається з одного координатора і декількох маршрутизаторів і кінцевих пристроїв.

У цій мережі кожен пристрій може зв'язуватися з будь-яким іншим пристроєм як безпосередньо, так і через проміжні вузли мережі. Сітчаста топологія підтримує «багатокроковий» (multi-hop) зв'язок, при якому дані проходять кроками від одного пристрою до іншого, використовуючи найбільш надійні лінії зв'язку та найбільш ефективні маршрути, поки не досягнуть мети.

Здатність до багатокрокової передачі допомагає забезпечити живучість мережі (здатність до самовідновлення). Якщо один із пристроїв відмовляє або перебуває під впливом перешкод, мережа здатна перемаршрутизуватися, використовуючи пристрої, що залишилися.

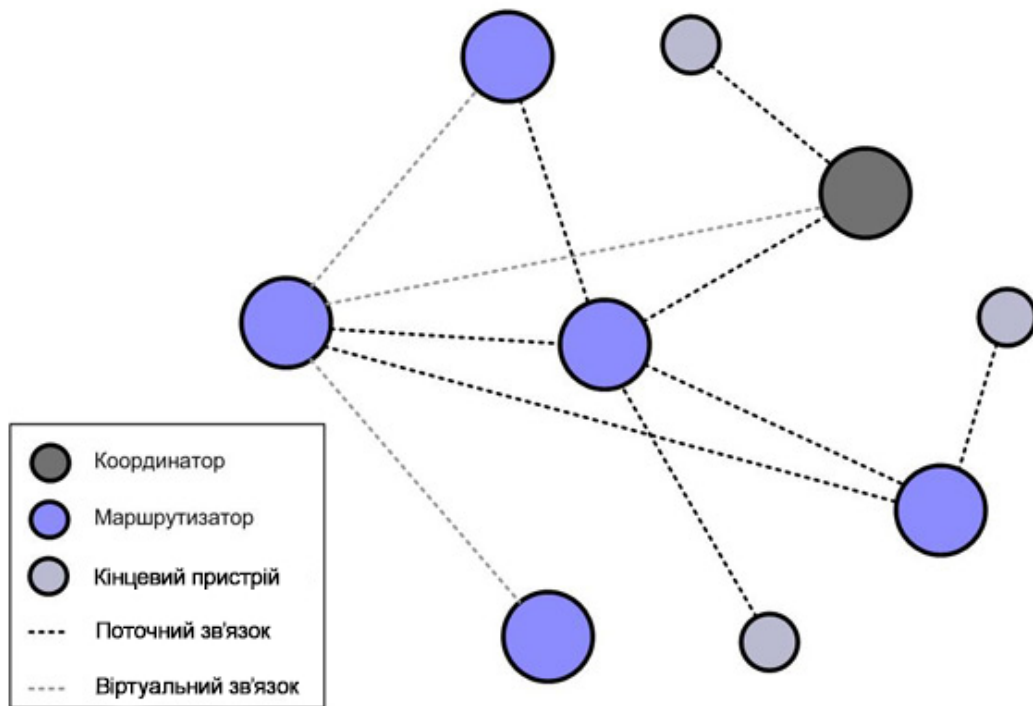


Рисунок 1.3 – Сітчаста топологія (mesh-топологія) мережі ZigBee

Принципова відмінність мереж ZigBee від інших бездротових мереж, таких як IEEE 802.11/WiFi, полягає в тому, що замість централізованої зіркоподібної структури мережі ZigBee допускають ієрархію приєднання до мережі. В результаті може бути досягнуто декілька рівнів приєднання. Така ієрархія пристроїв служить для створення бездротових мереж.

Для порівняння, IEEE 802.11/WiFi пропонує технологію точки бездротового доступу: бездротове підключення здійснюється між, скажімо, ноутбуком або телевізором та точкою доступу, яка не є бездротовим маршрутизатором. Тому далі потік даних спрямовується на дроти (наприклад, по дротовій мережі до інших дротових пристроїв або до іншої точки бездротового доступу). Мережа ZigBee здатна доставляти пакети повністю через бездротову мережу через кілька проміжних бездротових пристроїв.

Переваги mesh-топології:

- Сітчаста топологія має високу живучість і надійність. Якщо маршрутизатор стає недоступним, можуть бути знайдені та використані альтернативні маршрути.

- Використання проміжних пристроїв при передачі даних робить мережу широко масштабованою.
- Слабкі сигнали та мертві зони можуть бути легко виключені простим додаванням додаткових маршрутів.

Будь-яка мережа ZigBee починається з її формування. Пристрій, здатний бути координатором (воно призначається при проектуванні мережі), визначає вільний від перешкод канал і настраює себе координатором. У ZigBee цей пристрій називається PAN координатором – координатором персональної (на відміну від локальної – LAN) мережі.

Пристрої, які хотіли б приєднатися до мережі (підключитися, увійти до складу мережі), надсилають широкомовний запит. Спочатку цей запит відповідає лише PAN координатор. Надалі, крім PAN координатора, надати приєднання до мережі можуть маршрутизатори ZigBee, які вже приєдналися до PAN координатора безпосередньо або через інші маршрутизатори. Після того, як пристрій, який бажає приєднатися до мережі, отримує відповідь на широкомовний запит, далі йде обмін повідомленнями, які визначають, чи може пристрій приєднатися до мережі. Ключовим фактором визначення можливості є здатність маршрутизатора обслужити нові пристрої на додаток до раніше підключених.

Існує два шляхи вступу пристроїв до мережі ZigBee: MAC асоціація та повторне мережне приєднання (NWK rejoin).

Як очевидно з назви, MAC асоціація складає MAC рівні. Здатністю до MAC асоціації повинен за замовчуванням мати кожен пристрій ZigBee.

Механізм MAC асоціації наступний:

- Координатор або маршрутизатор, який збирається дозволити іншим пристроям приєднатися до нього, повинен виставити на свій рівень MAC дозвіл на приєднання NLME-PERMIT-JOINING.request.
- У свою чергу пристрій, який бажає вступити в мережу, виставляє на свій MAC рівень запит на приєднання NLME-NETWORK-DISCOVERY.request.

- Після цього пристрій, що вступає в мережу, передає в ефір ширококомовний запит маячка і отримує цей маячок від маршрутизатора(ів), готового(их) підключити новий пристрій.
- Отримавши маячок, пристрій визначає, в яку мережу він хоче вступити і до якого конкретно пристрою приєднатися, після чого виставляє на свій MAC рівень вимогу NLME-JOIN.request із встановленим на значення FALSE атрибут «повторне приєднання» (rejoin).
- Зробивши це, пристрій надсилає на обраний маршрутизатор запит на приєднання і отримує відповідь, що містить присвоєну йому мережеву адресу.

Повторне мережне приєднання (NWK rejoin) всупереч назві може використовуватися і при першому підключенні пристрою до мережі. Це означає, по-перше, що стандартний механізм MAC асоціації в цьому випадку може застосовуватися (ZigBee маршрутизатор виставить дозвіл на приєднання NLME-PERMIT-JOINING.request), або ні. По-друге, це означає, що обмін пакетами може бути безпечним, якщо пристрій знає поточний мережевий ключ. Це можливо, якщо пристрій дійсно перепідключається, або якщо він вмикається вперше, але отримав мережевий ключ через якийсь сторонній механізм (наприклад, при налаштуванні).

При такому підключенні пристрій, який бажає вступити в мережу, виставляє на свій мережевий рівень (NWK) запит на приєднання NLME-JOIN.request і обмінюється з маршрутизатором, що підключає його, пакетами «запит підключення» – «відповідь на запит підключення».

Крім процесу підключення нових пристроїв до зміни структури мережі, можуть призвести інші процеси. Пристрої можуть, наприклад, залишати мережу або повторно підключатися до мережі в інших місцях (наприклад, при перезавантаженні пристрою). Однією з важливих переваг мережі ZigBee є здатність відстежувати пристрої та топологію мережі в режимі частих підключень, відключень та перепідключень пристроїв.

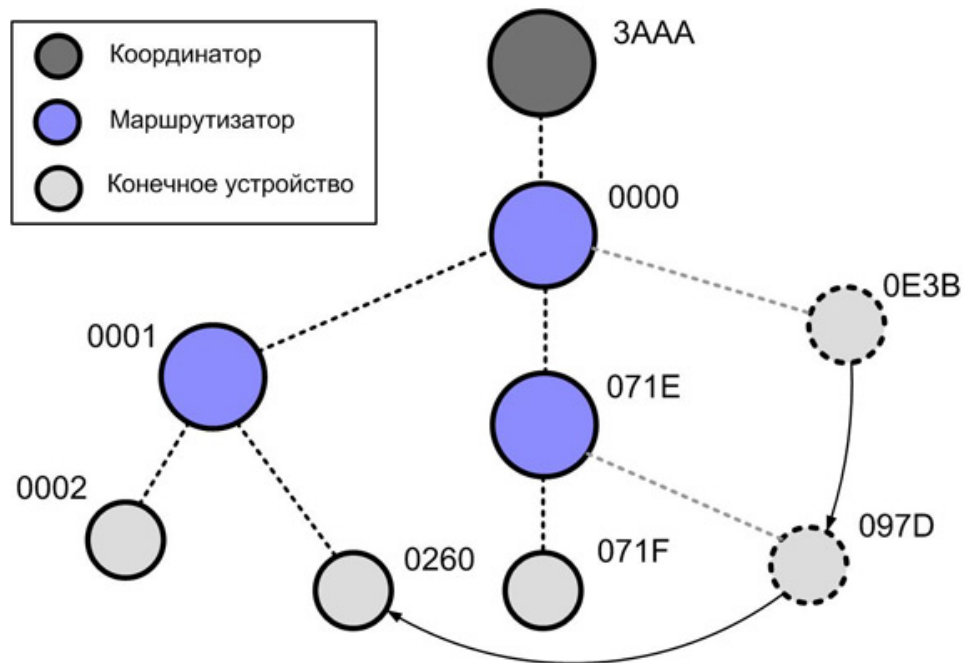


Рисунок 1.4 - Переключення кінцевого пристрою в деревоподібній мережі

Основний алгоритм маршрутизації в мережах ZigBee - Ad hoc On Demand Distance Vector (AODV) заснований на понятті "вектор відстані" маршруту, коли кожен маршрутизатор, що бере участь у трансляції запиту маршруту від конкретного джерела до конкретного пункту призначення створює свій запис у маршрутній таблиці. Цей запис як мінімум містить «логічну відстань» від джерела запиту та адресу попереднього маршрутизатора.

Алгоритм працює наступним чином:

1. Коли пристрою "джерело" (И) необхідно відправити пакет на пристрій "адресат" (А), джерело ініціює процес виявлення маршруту шляхом розсилки широкомовного пакета "запит маршруту", запитуючи маршрут до пристрою А;

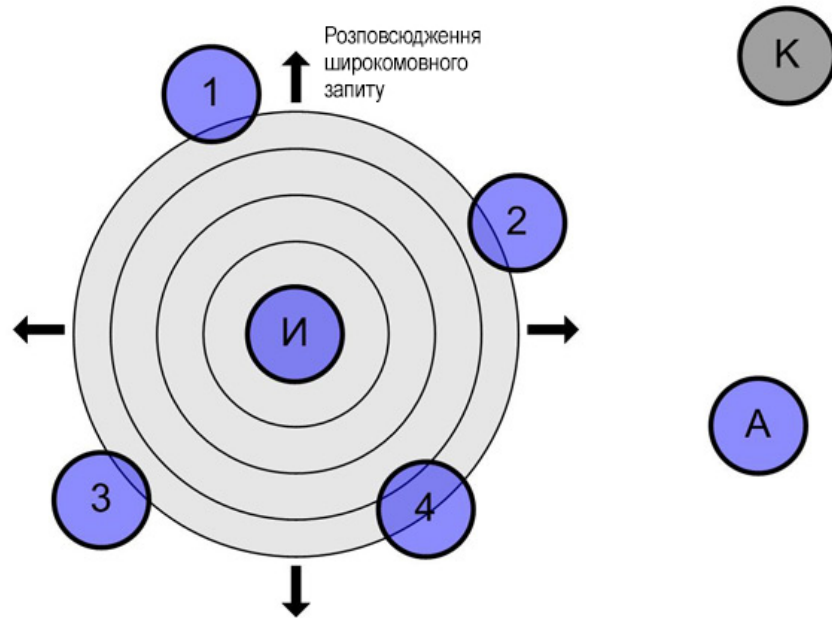


Рисунок 1.5 - Пошук маршруту

2. Кожен пристрій, який отримав цей запит маршруту, вносить свій запис до таблиці маршрутизації і може транслювати цей запит, включивши в запит таблицю зі своїм записом;
- 3.

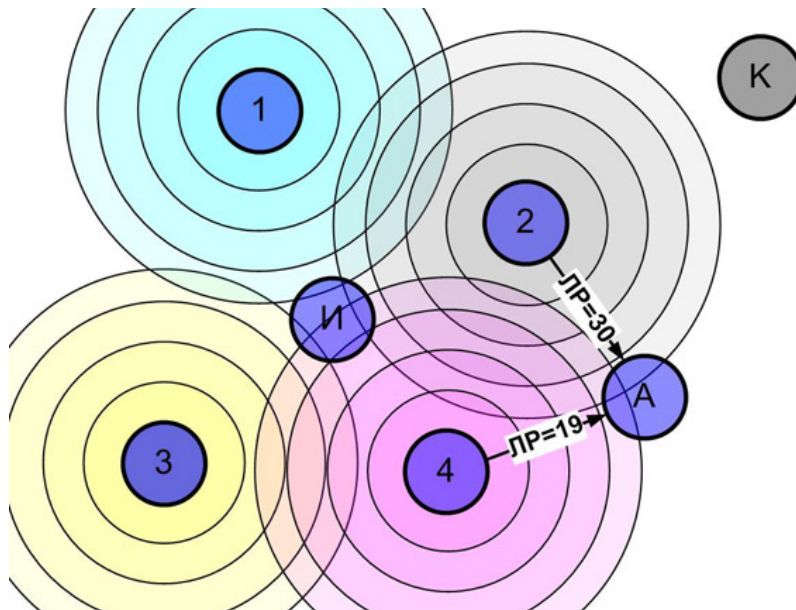


Рисунок 1.6 - Пошук маршруту (ЛР – логічна відстань)

4. Наступні пристрої, що отримали ретрансльований запит, транслюють його далі, і з кожним кроком «логічна відстань», пройдена запитом, збільшується на величину, яка залежить від якості зв'язку на ділянці;
5. Процес триває поки запит не досягає пристрою А, причому зазвичай пристрою А досягає кілька ретрансльованих запитів (від різних маршрутизаторів) та «логічні відстані» (ЛР) маршрутів цих пакетів різні;
6. Пристрій А відправляє відповідь за маршрутом, що має мінімальну «логічну відстань»;

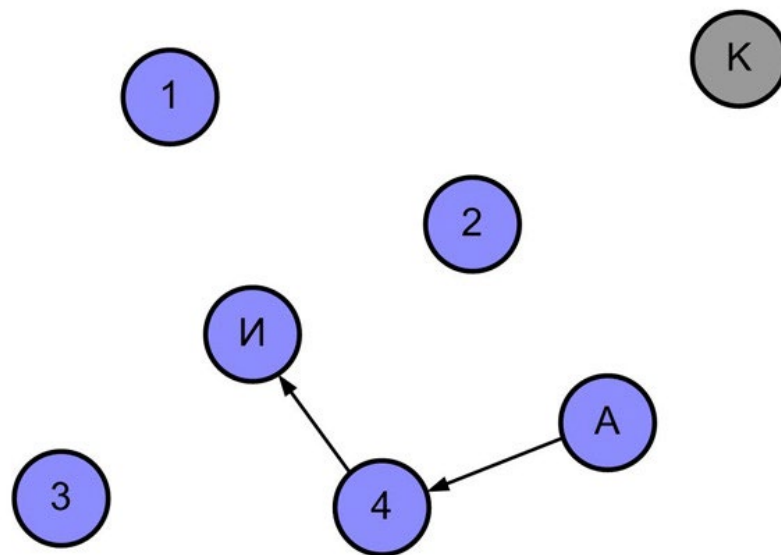


Рисунок 1.7 – Відправка відповіді най коротшим маршрутом

7. Відповідь повертатиметься у відповідність до таблиці маршруту шляхом, пройденим початковим запитом, поки не прибуде на пристрій И.
8. Ця серія відповідей, що передаються у зворотному напрямку, формує прямий маршрут для майбутньої передачі пакетів від И до А. Механізм «логічної відстані» дозволяє джерелу та вузлам, розташованим на шляху запиту, вибрати мінімальну «логічну відстань» маршруту від джерела до місця призначення.

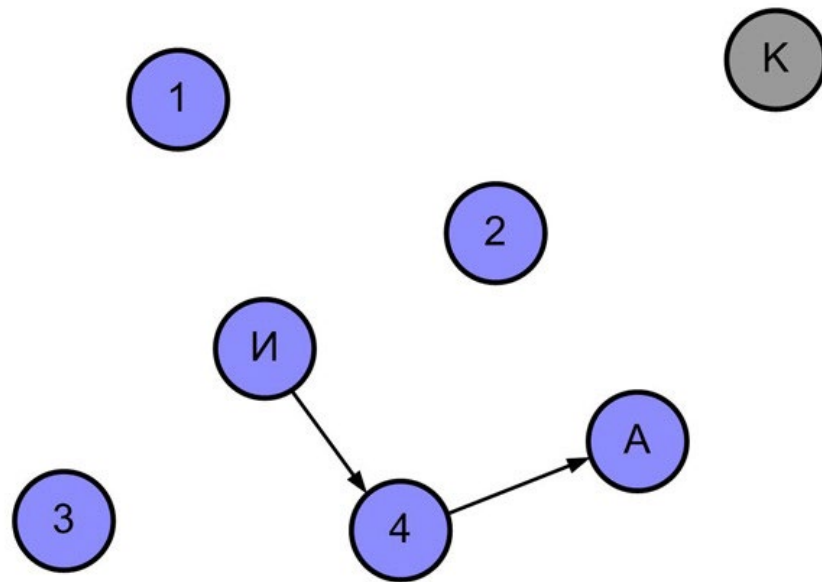


Рисунок 1.8 – Відповідь повертається до початкового пристрою

Описаний базовий алгоритм ефективний та універсальний, проте потребує значного обсягу пам'яті для зберігання маршрутів, що ускладнює та подорожчає пристрої.

Тому в мережах ZigBee реалізовано й інший алгоритм, який дозволяє знизити вимоги до обсягу пам'яті, а іноді й зменшити мережевий трафік, необхідний пошуку маршрутів. Дія алгоритму полягає в тому, що адреси в ZigBee мережі розподіляються ієрархічно, починаючи з координатора. Пристрій, що не має можливостей маршрутизації або пристрій, можливості маршрутизації якого вичерпані, може скористатися ієрархічною маршрутизацією – менш ефективною, проте цілком практичною.

У процесі формування ZigBee мережі алгоритм розподілу адрес задає діапазони адрес мережевим пристроям в ієрархічному порядку, починаючи з координатора. У результаті будь-який пристрій у мережі, знаючи свою адресу та адресу одержувача пакета, може визначити, чи належить конкретна мережна адреса до «низхідної» гілки (і до якої саме), чи знаходиться в іншому місці в ієрархії пристроїв. Виходячи з цього, будь-який пристрій може прийняти просте рішення маршрутизації: передавати пакет "вгору" - у напрямку координатора або "вниз" - до дочірнього пристрою.

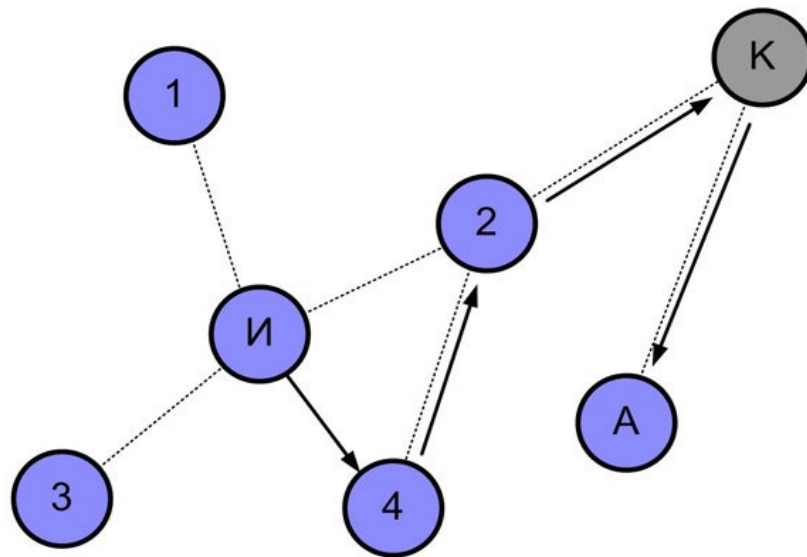


Рисунок 1.9 – Приклад ієрархічної маршрутизації

Як у випадку на рис. 1.7, пакет, відправлений пристроєм И, призначений для пристрою А. Однак пристрій 4 вичерпав свої можливості маршрутизації, тому не може транслювати пакет безпосередньо на пристрій А, а замість цього, використовуючи ієрархічну маршрутизацію, направляє цей пакет «вгору» ієрархічними сходами – на пристрій 2. Далі пакет транслюється на координатор К, який передає його на адресу А.

Переваги ієрархічної маршрутизації полягають у її простоті та меншому використанні ресурсів, що дозволяє створювати дуже недорогі пристрої без можливостей маршрутизації, які можуть брати участь у будь-якій ZigBee сумісній мережі. Недолік алгоритму в тому, що пакети будуть рухатися вгору і вниз до координатора і назад навіть у тому випадку, коли між джерелом та приймачем можливий прямий зв'язок.

Профіль додатків, кластери та кінцеві точки. Профіль програми описує набір пристроїв, що використовуються для певної програми, та опосередковано схеми обміну повідомленнями між цими пристроями. Наприклад, є профілі додатків для «розумного дому», автоматизації будівлі,

системи безпеки тощо. Для унікальної ідентифікації програм кожному додатку виділяється ідентифікатор профілю.

Існує два типи профілів додатків:

- загальні профілі програм – сумісне прикладне програмне забезпечення, розроблене альянсом ZigBee, яке виконує конкретне завдання;
- профілі виробників обладнання – приватні профілі програм, розроблені компаніями для роботи пристроїв ZigBee.

Пристрої в межах профілю програми спілкуються один з одним у вигляді кластерів, які можуть бути входами або виходами пристрою. Наприклад, у профілі «розумного будинку» є кластер, призначений для підсистем контролю освітлення. Ідентифікатор кластера унікально ідентифікує кластер у межах конкретного профілю.

Кінцева точка визначає об'єкт зв'язку всередині пристрою, через який реалізується певний додаток. Наприклад, пульт дистанційного керування може виділити кінцеву точку 6 для контролю освітлення в спальні, кінцеву точку 8 для керування системою обігріву та кондиціонування повітря та кінцеву точку 12 для керування системою безпеки. Це дозволяє пульту дистанційного керування незалежно зв'язуватися з цими пристроями та визначати, які пакети призначені для кожної програми та пристрою.

У будь-якому пристрої ZigBee можна використовувати до 240 кінцевих точок (1 – 240). Кінцева точка нуль виділена для об'єкта пристрою ZigBee (ZD0), який забезпечує команди контролю та управління самим пристроєм ZigBee.

ZCL – це бібліотека кластерів, яку можна використовувати будь-яким додатком. Це дозволяє використовувати кластери у різних сферах. Зокрема, кластери освітлення можуть використовуватися в будь-якій програмі, яка потребує управління освітленням, наприклад, в «розумному будинку» або в системі автоматизації комерційної будівлі.



Рисунок 1.10 - Кожен кластер складається з клієнта та сервера.

Система безпеки ZigBee, яка заснована на 128-бітному алгоритмі AES, доповнює модель безпеки, що надається IEEE 802.15.4. Служби безпеки ZigBee включають методи створення ключів та транспорту, управління пристроями та захисту пакетів даних. Специфікація ZigBee визначає безпеку для MAC, NWK та APS рівнів. Безпека програм зазвичай забезпечується через профілі програм.

Центр керування безпекою визначає, чи потрібно дозволити чи заборонити приєднання нових пристроїв до своєї мережі.

Центр керування безпекою може періодично оновлювати ключ мережі та переходити на новий ключ. Спочатку він транслює новий ключ, зашифрований за допомогою старого ключа мережі. Потім він повідомляє всі пристрої про перехід на новий ключ. Центром управління безпекою зазвичай є координатор мережі, але це може бути виділений пристрій.

Центр управління відіграє такі ролі у забезпеченні безпеки:

- довірений менеджер з автентифікації пристроїв, які бажають приєднатися до мережі;
- менеджер мережі, що підтримує та розповсюджує мережеві ключі;
- менеджер конфігурації, що забезпечує безпеку взаємодії пристроїв;
- ключі контролю доступу.

ZigBee використовує три типи ключів для керування безпекою: головний ключ, мережевий ключ та ключ каналу зв'язку.

Головні ключі:

Ці додаткові ключі не використовуються для шифрування пакетів. Вони використовуються як початковий секретний код, що розділяється двома пристроями, коли пристрої виконують процедуру генерації ключа каналу зв'язку. Головні ключі, які створюються центром керування безпекою, називаються головними ключами центру безпеки, тоді як інші ключі називаються основними ключами рівня додатків.

Мережеві ключі:

Ці ключі забезпечують безпеку мережного рівня мережі ZigBee. Мережевий ключ мають усі пристрої у мережі ZigBee. Мережні ключі високої безпеки повинні надсилатися бездротовими каналами тільки в зашифрованому вигляді, тоді як стандартні мережеві ключі безпеки можуть відправлятися як зашифрованими, так і не зашифрованими.

Ключі каналів зв'язку:

Ці додаткові ключі забезпечують безпеку одноадресної передачі між двома пристроями на рівні додатків.

Важливим етапом є введення в експлуатацію. Під введенням в експлуатацію розуміється фізичне розгортання конкретного рішення та логічна прив'язка вузлів на формування функціональної мережі. У широкому значенні цього слова введення в експлуатацію охоплює широке коло завдань, включаючи:

- промір радіо та фізичного середовища;
- розміщення пристроїв;
- конфігурацію їх параметрів;
- прив'язку додатків;
- оптимізацію мережі та параметрів пристроїв;
- тестування та підтвердження правильності роботи.

Крім того, при введенні в експлуатацію часто доводиться вирішувати суміжні питання, такі як наявність знаючого і досвідченого персоналу,

наявність необхідного обладнання, функціональна сумісність пристроїв, що використовуються, а також їх сумісність з іншими бездротовими або провідними системами.

### **1.3 Система керування IoT пристроями та їхньої автоматизації**

#### **Home Assistant**

Home Assistant — це універсальна безкоштовна операційна система для керування пристроями інтернету речей різних виробників, а також їхньої автоматизації. Це локальне рішення, що не залежить від хмари чи наявності інтернету, та підтримує велику кількість протоколів: Wi-Fi, Zigbee, Z-wave, Bluetooth, Bluetooth-mesh, IR etc.

Існує велика кількість сценаріїв використання, наприклад:

- керування безпекою: камери, сигналізації, датчики задимленості, відкриття, руху, вібрації;
- керування освітленням: регулювання яскравості, кольору, теплоти, підсвітка;
- керування побутовою технікою: кондиціонер, пилосос, телевізор, чайник;
- керування електроживленням: розетки, вимикачі, реле, лічильники;
- моніторинг: температури, вологості, тиску, CO<sub>2</sub>, протікання.

Одна з найбільших переваг сервісу – його доступність. Встановити Home Assistant на Windows, macOS, Linux або навіть на Raspberry Pi не складе труднощів. Система працює на пристроях із більшістю ОС.

Для керування пристроями Н.А. використовує окремі компоненти (components) та інтеграцію (integrations). Програма дозволяє об'єднати IoT-пристрої в рамках 1800 готових інтеграцій у 50+ категоріях – таких як Сигналізація, Автоматизація, Клімат, Відеоспостереження, Освітлення, Виявлення Присутності та Здоров'я.

Кількість доступних інтеграцій дуже велика, однак якщо серед всього цього різноманіття не знайдеться потрібної системи/пристрою, є можливість реалізувати сторонню інтеграцію без складнощів. До того ж, швидше за все, це не забере багато часу – плагіни для багатьох девайсів вже можуть бути доступні завдяки великій спільноті Home Assistant.

Інтерфейс програми цілком зрозумілий і переведений на безліч мов, включаючи українську. Кожен користувач може змінювати картки на панелі керування за власним бажанням.

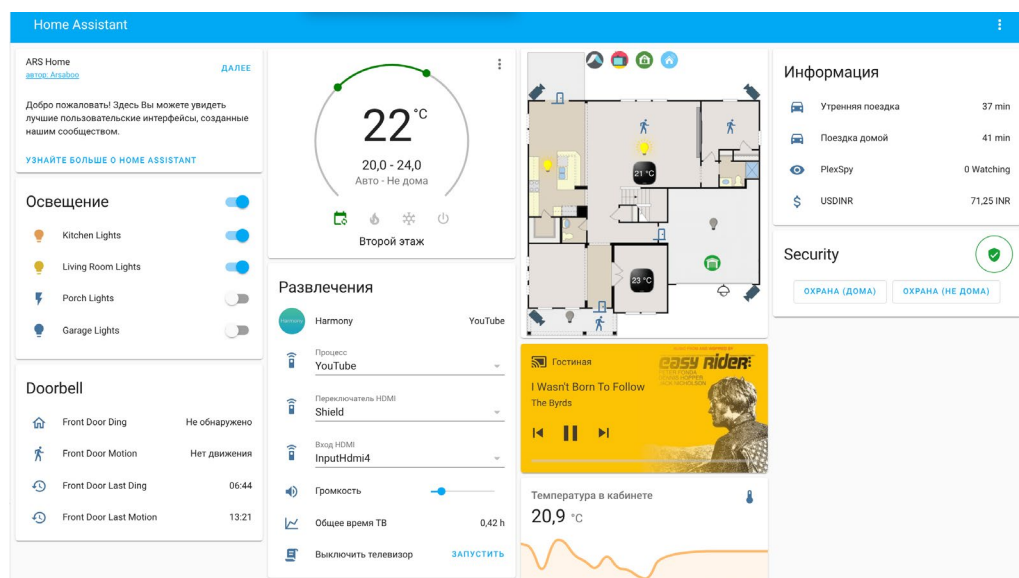


Рисунок 1.11 – Приклад інтерфейсу Home Assistant

Редагування карток можливе як у візуальному редакторі так і у текстовому.

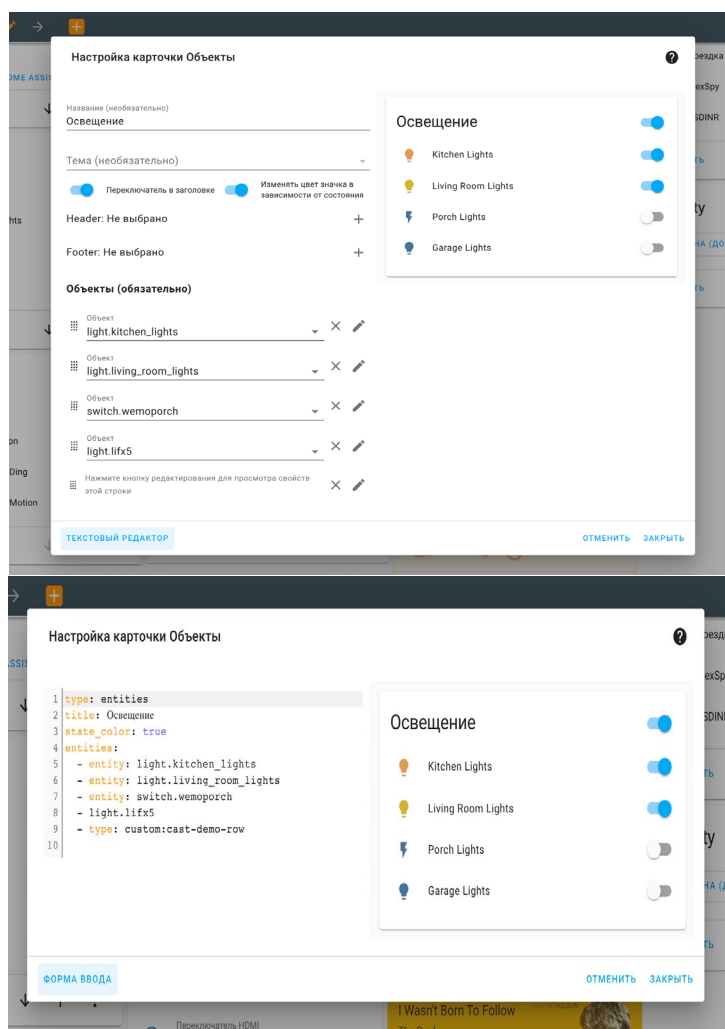


Рисунок 1.12 – Візуальний та текстовий редактори

### Автоматизація:

Автоматизації в Home Assistant дозволяють задати дію, яка буде застосовуватись у момент виникнення певної події. Так, кожна автоматизація складається з тригера та дії. Іноді сюди додається ще й умова. Наприклад, «коли температура повітря більше 60 градусів» - це триггер, «відправити оповіщення на телефон» - це дія.

### Оповіщення:

Надсилення користувачеві повідомлень про значні події та сигнали від пристроїв не тільки важливе, але й просто необхідне для своєчасної зворотної реакції. І тут Н.А. пропонує кілька можливостей:

1. Внутрішні сповіщення Н.А. (persistent notifications) – дозволяють налаштувати надсилання повідомлень безпосередньо в інтерфейс Home Assistant.
2. Вбудований компонент notify – дозволяє надсилати повідомлення у сторонні платформи через встановлення додаткових інтеграцій – наприклад, Slack, Telegram, Facebook Messenger.

## 2 ВПРОВАДЖЕННЯ В ВИРОБНИЦТВО СИСТЕМИ КОНТРОЛЮ ЗА СТАНОМ ОБЛАДНАННЯ

### 2.1 Обладнання яке потребує систему контролю

Впровадження системи контролю виконувалось на виробництві, яке виготовляє меблі. Обладнання яке використовується на виробництві – це верстати з ЧПУ управлінням та різноманітним електронним вмістом, яке має схильність до високого нагріву в процесі експлуатації. Тому важливим кроком було розробити систему для контролю за нагрівом, яка буде збирати потрібну інформацію, обробляти та у разі збільшення температури до критичних значень, відправляти оповіщення відповідному персоналу. Проведено дослідження стану температури після впровадження системи та зроблені висновки про ефективність її роботи.

Основне обладнання в якому буде встановлено датчики, це верстати з числовим програмним управлінням типу Biesse Rover a 1632 та KDT KS-828C.



Рисунок 2.1 – Верстати з ЧПУ Biesse Rover a 1632 та KDT KS-828C

В обладнанні такого типу використовується високотехнологічна система управління, яка включає багато електронних компонентів таких як:

- програмовані логічні контролери (PLC);
- частотні перетворювачі або інвертори;

- високовольні сервоприводи;
- ЕОМ;
- блоки живлення та інші технічні компоненти.



Рисунок 2.2 – Внутрішня конструкція електронно-комутаційної шафи KDT KS-828C на підприємстві



Рисунок 2.3 - Внутрішня конструкція електронно-комутаційної шафи Biesse Rover a 1632 на підприємстві

Компоненти цих систем при своїй роботі нагріваються і виділяють багато тепла. Щоб уникнути сильного нагріву компонентів в електронно-комутаційних шафах, виробний верстатів встановлює теплообмінники. Це пристрій який складається з алюмінієвого радіатора та вентиляторів, їх основною задачею являється охолодження повітря в середині шафи за рахунок зовнішнього, більш холодного повітря.

Також виробники верстатів використовують системи охолодження на базі кондиціонерів, це більш ефективна система, але і більш схильна до виходу з ладу.

Так як будь-яка система охолодження потребує контролю, оскільки може забитися або вийти з ладу якийсь із її частин, що призведе до критичного нагрівання компонентів верстата і можливого виходу їх з ладу. Датчики виміру температури і вологості працюватимуть як «запобіжник», система контролю за станом обладнання буде збирати показники з датчиків і у разі виходу хоча б одного з показників за межі норми, робити оповіщення, це сигналізує о необхідності провести обслуговування системи охолодження верстатів. Збір показників буде проводитися з двох верстатів Biesse Rover a 1632 та двох KDT KS-828C.

## **2.2 Датчик температури і вологості**

Для повноцінної роботи системи контролю за станом обладнання потрібен пристрій для збору інформації. Цим пристроєм являться датчик температури та вологості.

На даний час представлена велика кількість готових рішень, які працюють за допомогою протоколів Wi-Fi або Bluetooth. Але пристрої на базі цих протоколів не підходять за важливими критеріями, для Wi-Fi – це низька енергоефективність, для Bluetooth – мала дальність роботи та відсутність mesh-топології.

Ідеальне рішення для цього проекту являється протокол ZigBee, який було описано в теоретичній частині. Його основні переваги, такі як:

- надійність та самовідновлення;
- практично не обмежена можливість розширення (mesh-топології);
- низьке енергоспоживання;
- вбудований алгоритм налаштування мережі та маршрутизації.

Робить вибір протоколу ZigBee обґрунтованим та правильним.

Готові рішення на базі протоколу ZigBee також присутні у продажу. Але було прийнято рішення скористатися проектом бездротового датчика вимірювання температури і вологості повітря на чіпі CC2530 Efekta THP\_LR \ THP.

Це рішення передбачає:

- замовлення друкованої плати за готовим дизайном у постачальника послуг з виробництва друкованих плат;
- придбання потрібних елементів та радіомодулів для роботи пристрою;
- пайка елементів на друковану плату;
- прошивка радіомодулю;
- виготовлення корпусу для плати на 3D принтері або іншим методом.

Вибір цього рішення обумовлено кількома причинами, по-перше це економічна вигідність. Ціна готового датчика з аналогічними характеристиками складає від 22\$ за один пристрій. Ціна всіх компонентів датчика із проекту CC2530 Efekta THP\_LR \ THP складає 14\$, що помітно дешевше ніж готові рішення.

Досвід використання готових датчиків показав, що є кілька нюансів, які потрібно було вирішити:

- батареї, які використовують у готових датчиках, погано тримають заряд;
- дальність передачі при використанні на великих ділянках теж частіше недостатня;

- також дуже важливим моментом при використанні власного датчика є програмна частина, з'являється можливість вирішувати, що, як і з яким часовим інтервалом датчик читатиме сенсори і надсилатиме дані.

Характеристики датчика:

Датчик можливо використовувати в двох модифікаціях. Варіант датчик працює на радіомодулі CC2530 з підсилювачем сигналу та без нього. У варіанті з підсилювачем можливо встановлення радіомодуля з антеною на друкованій платі або з виносною. Усі модифікації датчика працюють на батареях ААА, це забезпечує тривалий термін роботи на одному комплекті батарейок.

Плата датчика має розмір 29мм x 67мм. Місце під напайку радіомодуля зроблено універсальним для можливості встановлення декількох видів радіомодулів: EBYTE E18-MS1-PCB (200м), EBYTE E18-MS1PA2-PCB (800м), EBYTE E18-2G4Z27SI IPEX (2500м) [10].

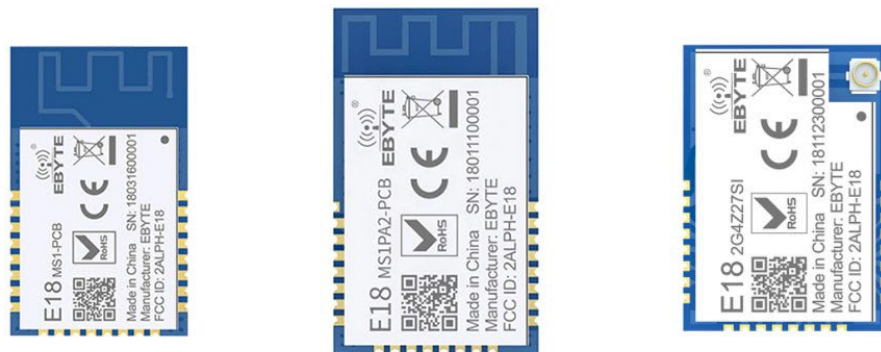


Рисунок 2.4 – Радіомодулі EBYTE E18-MS1-PCB(200м), EBYTE E18-MS1PA2-PCB (800м), EBYTE E18-2G4Z27SI IPEX (2500м).

Характеристики радіомодулів:

- робоча частота: 2400-2480 MHz;
- потужність передавача: 4 dbm, з підсилювачем 20 dbm;
- чутливість приймача: -96.4 dbm, з підсилювачем -98 dbm;
- швидкість передачі: 250kbps;

- протокол: zigbee;
- енергонезалежна пам'ять: 256 kb;
- енергозалежна пам'ять: 8 kb;
- ядро: мікроконтролер 8051;
- напруга живлення: 2.0 - 3.6v;
- споживання під час передачі: 28ma, з підсилювачем 98ma;
- споживання прийому: 27ma, з підсилювачем 36ma;
- споживання в режимі сну: 1.2µa, з підсилювачем 2µa;
- діапазон робочої температури -40 - +95°C.

Для зручності збірки датчика всі електронні компоненти розташовані на одній стороні плати, за винятком утримувача батареї, який розташований на зворотній стороні плати. У проекті використовувалися цифрові сенсори BME280, BMP280. У першому варіанті датчика використовувався сенсор bme280, але через кризу виробництва напівпровідників і, як наслідок, зростання цін надалі до проекту було додано сенсор BMP280. Це найдешевший варіант, вартість bmp280 в інтернеті становить приблизно 30 гривень. Цей сенсор вимірює тільки атмосферний тиск і температуру, щоб вимірювати ще й вологість в проект був доданий сенсор SHTC3, який коштує трохи дорожче, але дешевше ніж BME280.

Технічні параметри сенсорів:

BME280 (Bosch)

- напруга живлення: 1.71v - 3.6v;
- інтерфейс обміну даними: i2c чи spi;
- діапазон вимірювання атмосферного тиску: 300hpa – 1100hpa (арр ±0.12hpa), що еквівалентно діапазону від -500 до 9000 м над рівнем моря;
- діапазон вимірювання температури: -40 ° c ... +85 ° c (at ± 0.5 ° c);
- діапазон виміру вологості: 0% ... 100% (arh ±3%);
- енергоспоживання: режим вимірювань: 340 – 714 uA; у сплячий режим: - 0.1 uA.

### BMP280 (Bosch)

- напруга живлення: 1.71V - 3.6V;
- інтерфейс обміну даними: I2C чи SPI;
- діапазон вимірювання атмосферного тиску: 300hPa – 1100hPa ( $\Delta p \pm 0.12\text{hPa}$ ), що еквівалентно діапазону від -500 до 9000 м над рівнем моря;
- діапазон вимірювання температури:  $-40\text{ }^{\circ}\text{C} \dots +85\text{ }^{\circ}\text{C}$  ( $\Delta t \pm 0.5\text{ }^{\circ}\text{C}$ );
- енергоспоживання: режим вимірювань: 325 - 720uA; у сплячий режим: - 0.1 uA.

### SHTC3 (Sensirion)

- напруга живлення: 1.62V - 3.6V;
- інтерфейс обміну даними: I2C;
- діапазон вимірювання температури:  $-40\text{ }^{\circ}\text{C} \dots +125\text{ }^{\circ}\text{C}$  ( $\Delta t \pm 0.2\text{ }^{\circ}\text{C}$ );
- діапазон виміру вологості: 0% ... 100% ( $\Delta rh \pm 2\%$ );
- енергоспоживання: режим вимірювань: 270 - 430uA нА; у сплячий режим: - 0.3 uA.



Рисунок 2.5 – Цифрові сенсори BME280, BMP280 та SHTC3

На платі встановлений світлодіод для індикації режимів роботи: вхід у мережу, вихід з мережі, відправлення даних. Також є дві кнопки: кнопка скидання (reset) та кнопка користувача (mode) - (вхід у мережу, вихід з мережі, зчитування та відправка даних не за розкладом) та порт програмування.

На мою думку датчик простий і легкий у виготовленні, час складання датчика при ручному паянні становить 10-15 хвилин, схема датчика складається всього з 11 елементів, включаючи радіомодуль. Все це робить його використання доцільним та виправданим.

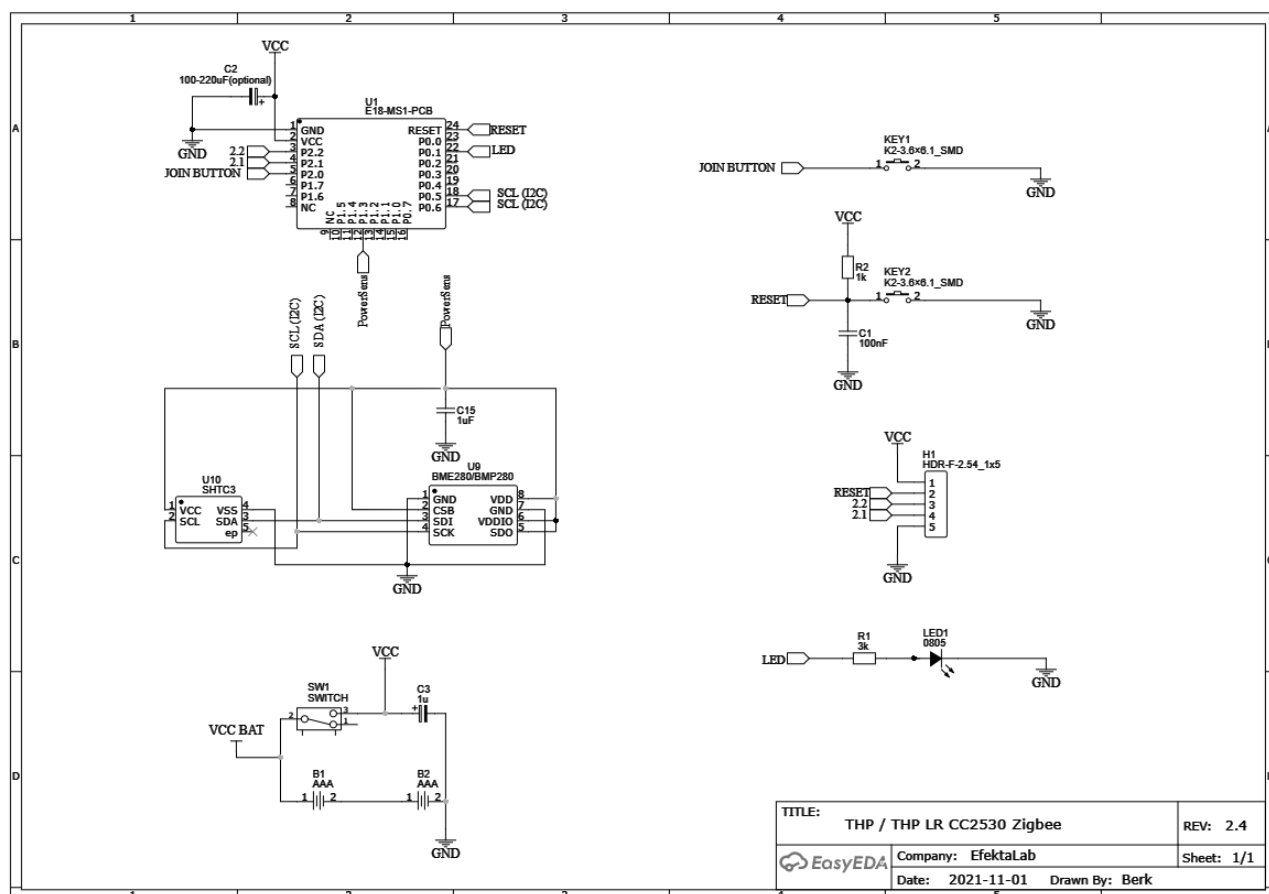


Рисунок 2.6 – Принципова схема датчика CC2530 THP\_LR \ THP

Таблиця 2.1 – Перелік елементів датчика та їх ціна

| Найменування елементу            | Кількість | Ціна           |
|----------------------------------|-----------|----------------|
| Конденсатор C1 100nF             | 1         | 4 грн          |
| Конденсатор C2 100uF             | 1         | 16,20 грн      |
| Конденсатор C3 1uF               | 1         | 12,10 грн      |
| Резистор R1 3k                   | 1         | 1,20 грн       |
| Світлодіод LED1 0805             | 1         | 12,50 грн      |
| Кнопка KEY1                      | 2         | 5 грн          |
| Радіомодуль<br>EBYTE E18-MS1-PCB | 1         | 216 грн        |
| Сенсор SHTC3                     | 1         | 80 грн         |
| Перемикач SW1                    | 1         | 13 грн         |
| Друкована плата та корпус        | 1         | 180 грн        |
| Всього:                          | 11        | 540 грн ~ 14\$ |

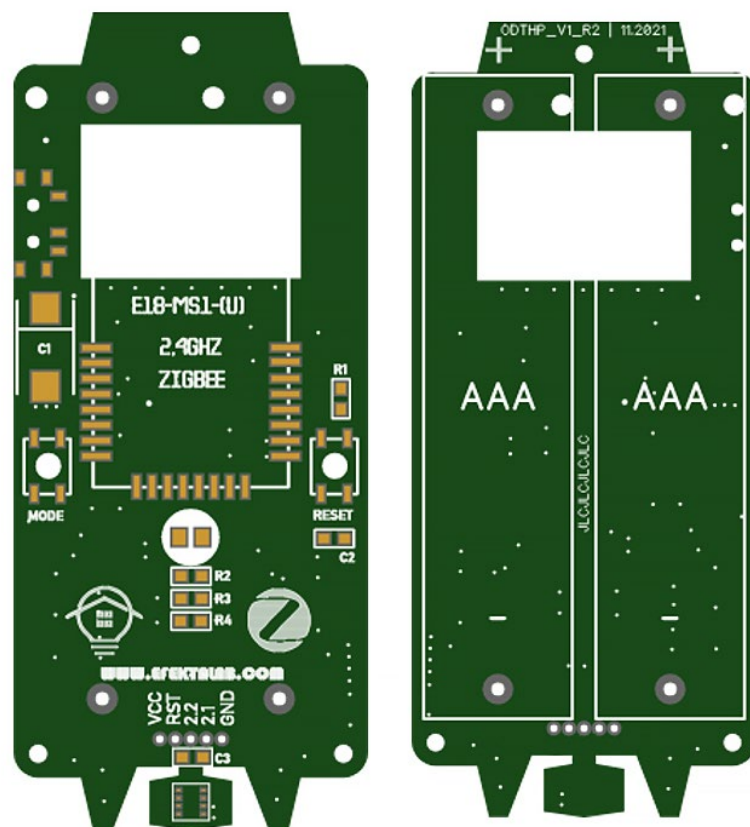


Рисунок 2.7 – Друкована плата датчика, передня та задня частини

Прошивка радіомодулю:

Програмна частина цього проекту доступна на репозиторію github проекту у вигляді вихідного коду, а також у вигляді скомпільованої прошивки. Була використана скомпільована прошивка для варіанта датчика з найдоступнішим радіомодулем EBYTE E18-MS1-PCB. Було встановлено інтервал читання сенсора один раз на хвилину та надсилення даних при зміні значень на 0.33 градуси для температури, на 2.5 відсотка для вологості. Надсилення інформації про рівень заряду батарейок відбувається один раз на 6 годин.

Для завантаження прошивки в радіомодуль використовувався програматор SmartRF04EB CC Debugger.



Рисунок 2.8 – Програматор CC Debugger

Для прошивки радіомодуля програматор треба підключити до комп’ютера за допомогою usb входу, а модуль ZigBee підключити до програматора за допомогою дротів для налагодження за схемою:

Таблиця 2.2 – Схема підключення радіомодуля до програматора

| CC2530 | Плата SmartRF04EB |
|--------|-------------------|
| P22    | DC                |
| P21    | DD                |
| RST    | RESET             |
| GND    | GND               |
| VCC    | 3.3V              |

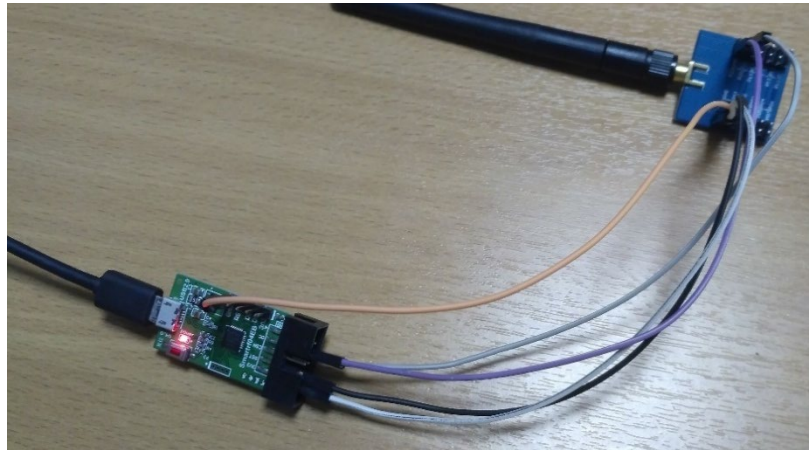


Рисунок 2.9 – Підключення програматора до радіомодулю за допомогою дротів

При підключенні програматора до USB-порту комп'ютера в програмі SmartRF Flash Programmer відразу з'явиться підключений пристрій, програму попередньо треба встановити на комп'ютер.

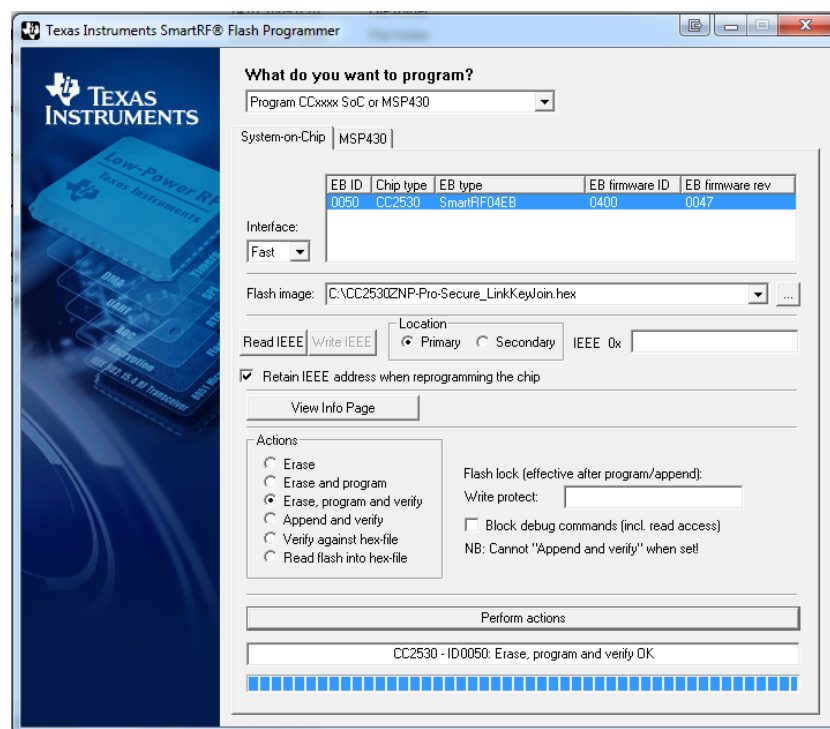


Рисунок 2.10 – Програма SmartRF Flash Programmer

Потрібно лише вказати шлях до файлу-прошивки та натиснути кнопку “Perform Actions”.

Приклад коду прошивки радіомодулю:

```
{
  zigbeeModel: ['EFEKTA_THP_LR'],
  model: 'EFEKTA_THP_LR',
  vendor: 'Custom devices (DiY)',
  description: 'DIY outdoor long-range sensor for temperature, humidity
and atmospheric pressure',
  fromZigbee: [fz.temperature, fz.humidity, fz.pressure, fz.battery],
  toZigbee: [tz.factory_reset],
  configure: async (device, coordinatorEndpoint, logger) => {
    const endpoint = device.getEndpoint(1);
    await reporting.bind(endpoint, coordinatorEndpoint, [
      'genPowerCfg', 'msTemperatureMeasurement',
'msRelativeHumidity', 'msPressureMeasurement']);
    const overrides = {min: 0, max: 64800, change: 0};
    await reporting.batteryVoltage(endpoint, overrides);
    await reporting.batteryPercentageRemaining(endpoint, overrides);
    await reporting.temperature(endpoint, overrides);
    await reporting.humidity(endpoint, overrides);
    await reporting.pressureExtended(endpoint, overrides);
    await endpoint.read('msPressureMeasurement', ['scale']);
  },
  exposes: [e.battery(), e.temperature(), e.humidity(), e.pressure()],
},
```

Корпус для датчика був розроблений під друк на FDM 3D принтері. У корпусі передбачено місце для виведення зовнішньої антени, товщина стінок в цьому місті зменшена, якщо потрібно встановити зовнішню антену для покращення рівню сигналу, треба буде висвердлити отвір.

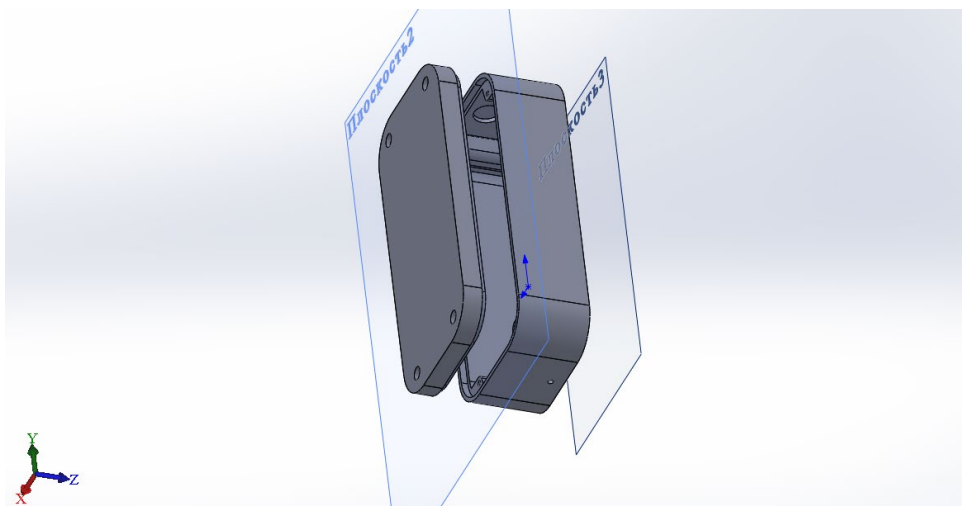


Рисунок 2.11 – 3D модель корпусу датчика

Після збірки друкованої плати і встановлення в корпус, датчик CC2530 THP\_LR \ THP має такий вигляд:



Рисунок 2.12 – Фото датчика, друкована плата та датчик в корпусі з зовнішньою антеною

Робота датчика перевірена, працездатність дуже стабільна. Так як інформація про датчик вже додана в проєкт Zigbee2MQTT, зовнішній конвертер для налаштування роботи з системою Home Assistant не потрібен.

## 2.3 Встановлення системи контролю за станом обладнання на базі Home Assistant

За допомогою системи автоматизації можливо налаштувати спільну роботу датчиків моніторингу стану обладнання. Головною задачею системи Home Assistant, буде збирання даних про температуру с датчиків встановлених в системи управління верстатом.

Для того щоб розпочати інтегрування датчиків CC2530 Efekta THP\_LR \ THP в систему потрібно встановити ОС Home Assistant на комп'ютер.

Ядром системи обрано одноплатний міні комп'ютер Raspberry Pi 4. Технічні характеристики Raspberry Pi 4:

- частота: 1500 МГц, Broadcom BCM2711, Cortex-A72;
- кількість ядер: Чотири;
- оперативна пам'ять: 2 ГБ;
- інтегрована графіка: VideoCore VI 500 МГц;
- інтерфейс: Wi-Fi 802.11ac, Bluetooth 5.0 LE, гігабітовий Ethernet RJ-45;
- швидкість передачі даних через LAN: 1000 Мб Ethernet RJ-45;
- порти: microHDMI, UART, SPI, I2C, аудіовиход 3.5 мм, USB 3.0, USB 2.0
- живлення: 5 В, 3 А через microUSB.



Рисунок 2.13 - Raspberry Pi 4 та допоміжні елементи для роботи комп'ютера

В роботі ОС Н.А. доволі часто робить звернення до накопичувача на якому встановлена, що може призвести до передчасного виходу з ладу MicroSD. Для того щоб подовжити стабільну роботу всій системи використовується SSD накопичувач, який має помітно більший ресурс роботи.

Запис образу на завантажувальний носій:

1. Підключаємо завантажувальний носій Home Assistant до комп'ютера.
2. Завантажуємо та запускаємо програму Balena Etcher.
3. Вибираємо «Flash із URL».
4. Прописуємо URL адресу образу ОС [https://github.com/homeassistant/operatingsystem/releases/download/9.3/haos\\_rpi4-64-9.3.img.xz](https://github.com/homeassistant/operatingsystem/releases/download/9.3/haos_rpi4-64-9.3.img.xz). і натискаємо OK.
5. Balena Etcher тепер завантажить образ, після цього натискаємо Select target.
6. Вибираємо SSD накопичувач, який буде використовуватися для Raspberry Pi.
7. Натискаємо "Flash!" щоб почати писати образ.
8. Коли Balena Etcher закінчить записувати образ, буде відображатися повідомлення "Flash Complete!".

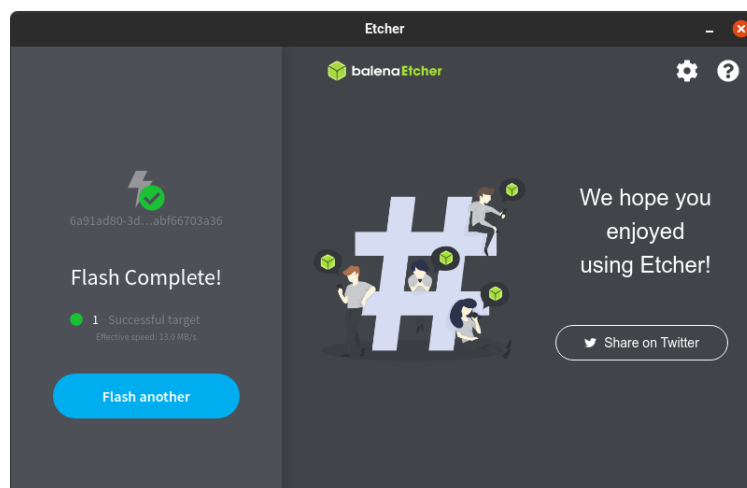


Рисунок 2.14 – Успішне закінчення запису образу на завантажувальний носій

На цьому етапі накопичувач з встановленою системою готовий до роботи.

Щоб розпочати роботу в Home Assistant потрібно:

1. підключити тільки що створений накопичувач до одноплатного комп'ютера;
2. підключіть кабель Ethernet до мережі;
3. підключіть кабель живлення;
4. через кілька хвилин у браузері системи Windows отримаємо доступ до системи автоматизації Н.А. за адресою `homeassistant.local:8123`

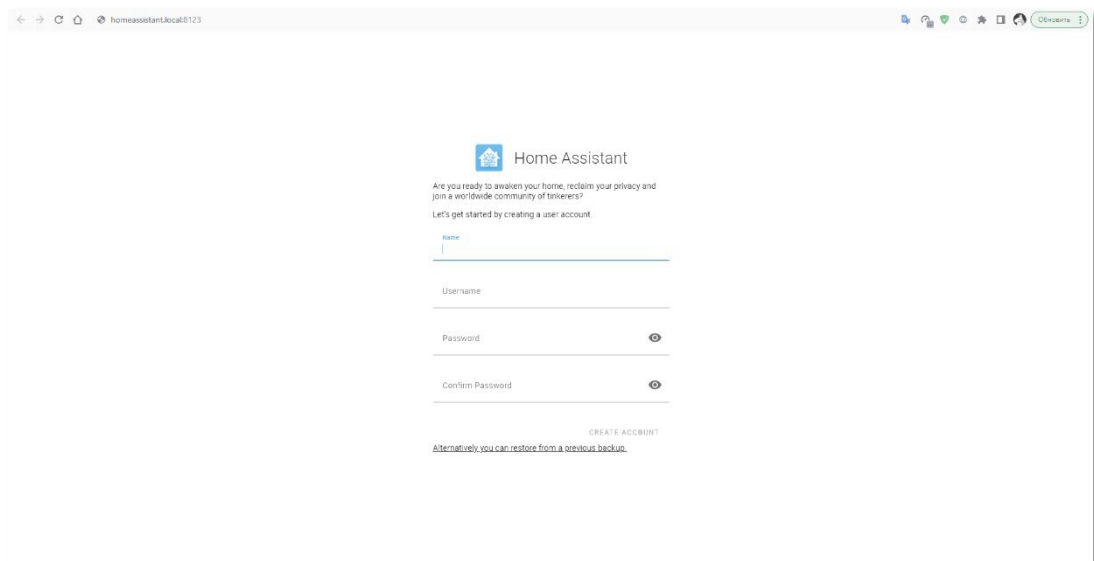


Рисунок 2.15 – Веб сторінка реєстрації в системі

На сторінці реєстрації можливо створити обліковий запис власника Home Assistant. Цей обліковий запис буде адміністратором і має повний доступ к налаштуванням системи. Вводимо назву, ім'я користувача, пароль та натискаємо кнопку «Створити обліковий запис».

Після створення облікового запису з'явиться веб-інтерфейс Home Assistant. На цьому екрані буде показано всі пристрої та їх карту розташування. Отже, настав час налаштування інтерфейсу та додавання датчиків для моніторингу за станом обладнання.

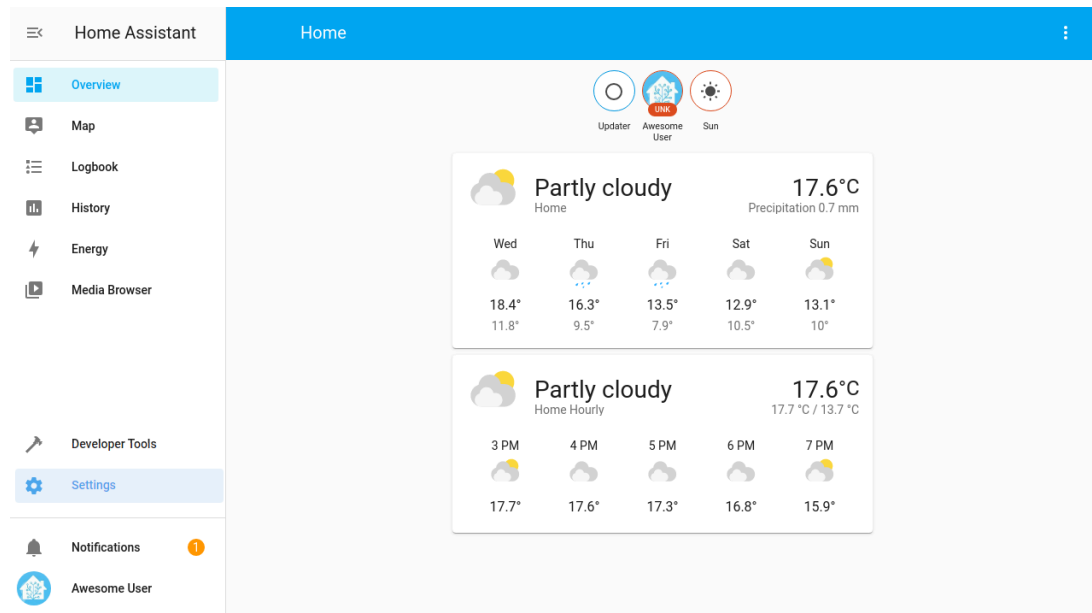


Рисунок 2.16 – Базовий інтерфейс Home Assistant без налаштування

## 2.4 Налаштування системи Home Assistant, додавання датчиків для моніторингу за станом обладнання

Для роботи системи Home Assistant з Zigbee протоколом повинен бути встановлений і налаштований MQTT брокер, для цього буде використовуватися Mosquitto MQTT сервер «Home Assistant Mosquitto».

MQTT (він же MQ Telemetry Transport) — це «міжмашинний» протокол або протокол підключення «Інтернет речей» на основі TCP/IP. Це забезпечує надзвичайно легкий транспорт повідомлень для роботи пристроїв.

Додавання MQTT до екземпляра Home Assistant можна здійснити через інтерфейс користувача, щоб налагодити роботу MQTT і Home Assistant треба вибрати MQTT брокера. Найбільш безпечним варіантом є запуск власного брокера MQTT.

Встановлення додатку Mosquitto MQTT Server:

1. У бічному меню вибираємо Hass.io, у верхньому меню є кнопка ADD-ON STORE, потрібно натиснути її та вибрати із списку додатків Mosquitto broker.

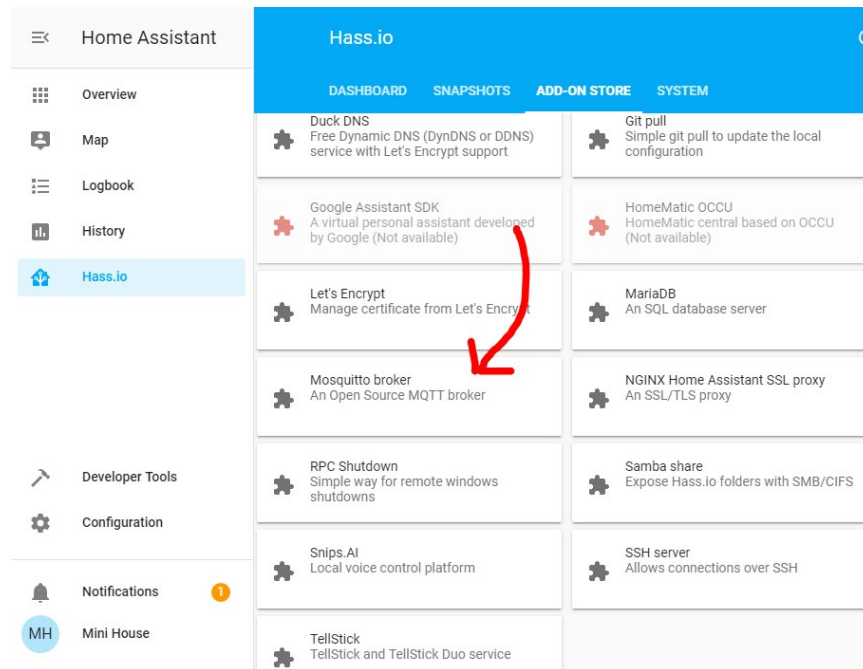


Рисунок 2.17а – Встановлення додатку Mosquitto MQTT Server

2. Натискаємо кнопку Install, щоб встановити додаток.
3. Додаємо наступний код у текстове поле конфігурації:

```
{
  "logins" : [
    {
      "username" : "MQTT" ,
      "password" : "12345678"
    }
  ] ,
  "anonymous" : false ,
  "customize" : {
    "active" : false ,
    "folder" : "mosquitto"
  } ,
  "certfile" : "fullchain.pem" ,
  "keyfile" : "privkey.pem" ,
}
```

Ключи "username", "password" треба змінити, вони використовуються для підключення до брокера MQTT.

4. Зберігаємо зроблені налаштування.
5. Натискаємо кнопку START, щоб запустити новий брокер MQTT.
6. В меню Configuration вибираємо пункт Integrations.
7. У цьому списку ми бачимо MQTT який очікує приєднання. Натискаємо кнопку Configure.

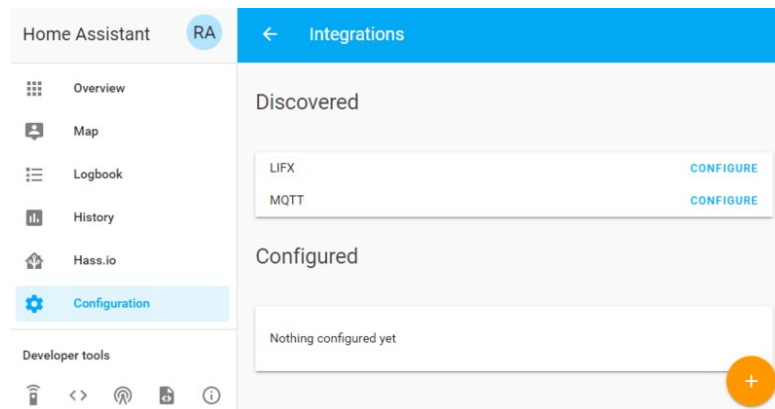


Рисунок 2.17б - MQTT додаток очікує приєднання

8. Натискаємо SUBMIT, потім FINISH.

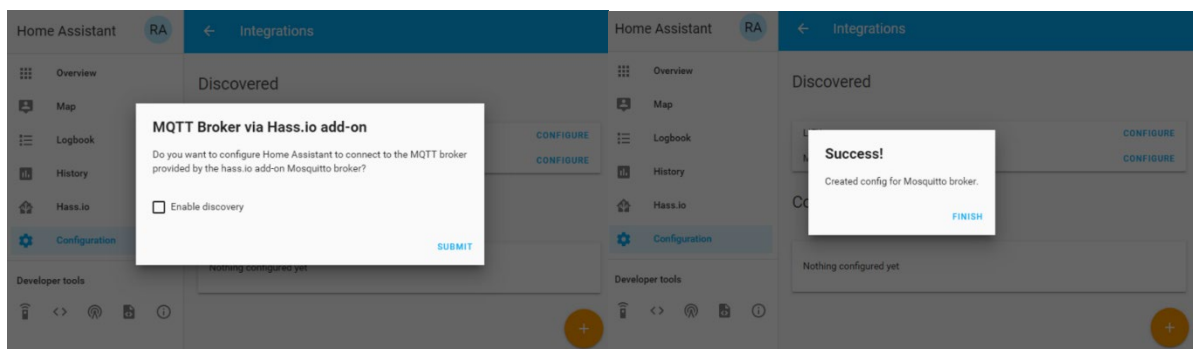


Рисунок 2.17в - Власний сервер MQTT встановлено

Тепер власний сервер MQTT налаштовано, він пов'язаний із сервером Home Assistant, з будь-якого клієнта MQTT є можливість взаємодіяти з каналами, використовуючи IP-адресу Home Assistant як хост, а також ім'я

користувача та пароль MQTT, які було встановлено. Це працюватиме лише у локальній мережі та не потребує ніяких сторонніх сервісів MQTT.

Наступним етапом являється прошивка, налаштування та встановлення USB ZigBee Sniffer, це пристрій який було використано як шлюз конвертера ZigBee в MQTT.

Є кілька способів прошивки, для цього завдання було використано програматор CC-Debugger та спеціалізований кабель.

Підключаємо програматор через кабель до ZigBee Sniffer, а пристрій в USB порт комп'ютера для живлення.



Рисунок 2.18 – Підключення USB ZigBee Sniffer до програматора

Завантажуємо прошивку zigbee2mqtt за посиланням <https://1wire.com.ua/download/ha/CC2531ZNP-Prod.zip>.

За допомогою вже знайомої програми прошиваємо ZigBee Sniffer.

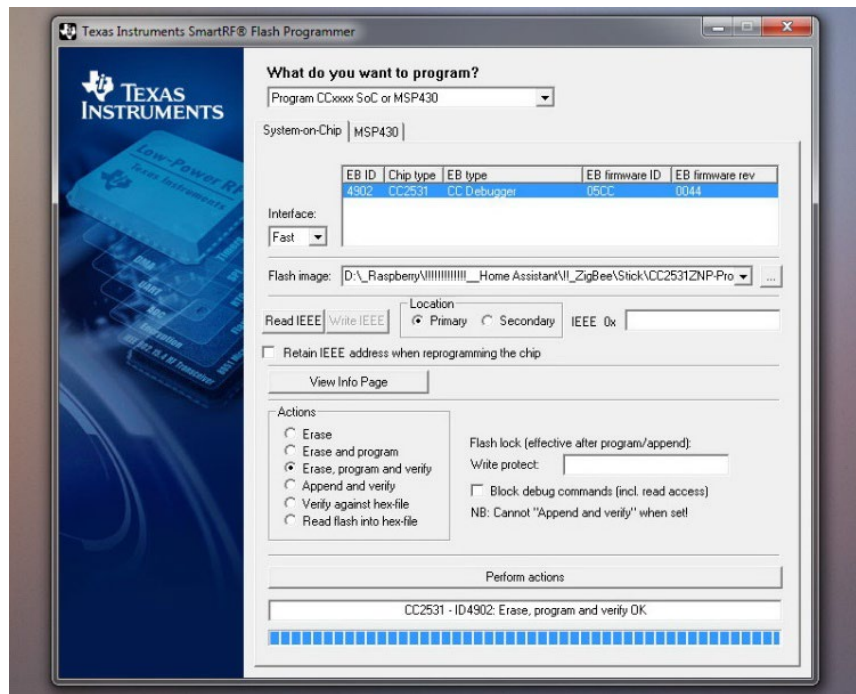


Рисунок 2.19 – Прошивка USB ZigBee Sniffer

Ця прошивка представляє собою шлюз для конвертування протоколу ZigBee в протокол MQTT, який є універсальним і використовується всіма платформами для розумних речей.

Після цього підключаємо ZigBee Sniffer до порту Raspberry PI. Тепер необхідно дізнатися, на якому порту знаходиться сам Sniffer, для цього в панелі Hass.io переходимо в меню System і натискаємо кнопку Hardware.

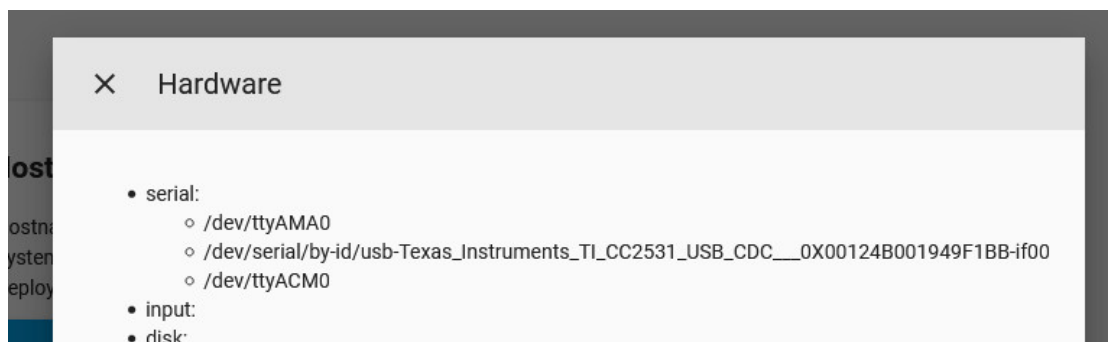


Рисунок 2.20 – Серійний порт на якому працює USB ZigBee Sniffer

У моєму випадку серійний порт USB Sniffer має адресу /dev/ttyACM0.

Після цього необхідно встановити плагін Zigbee2mqtt. Переходимо на вкладку Hass.io на закладку ADD-ON STORE і додаємо новий репозиторій - <https://github.com/danielwelch/hassio-zigbee2mqtt>.

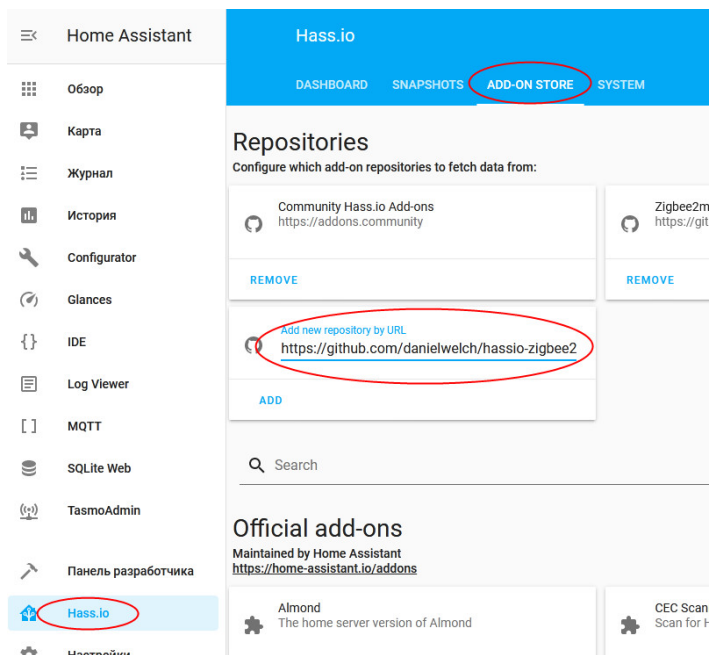


Рисунок 2.21а – Встановлення плагіну Zigbee2mqtt

Встановлюємо плагін Zigbee2mqtt та переходимо на вкладку Dashboard.

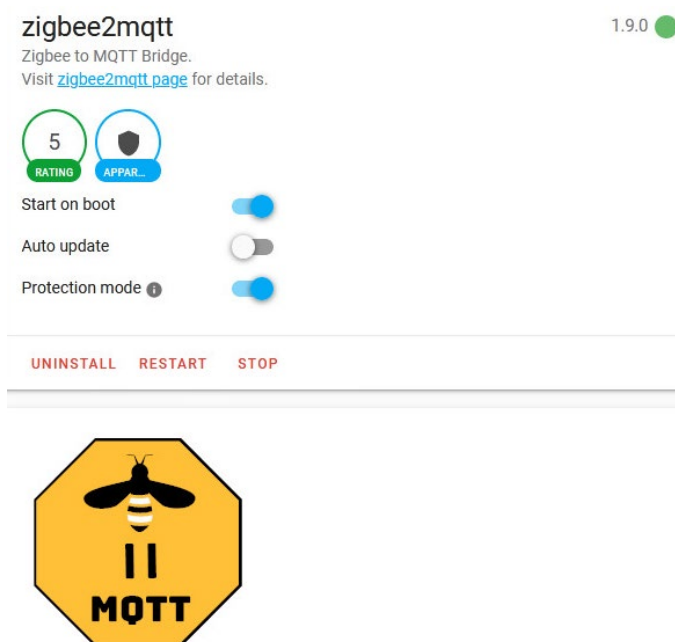


Рисунок 2.21б – Налаштування плагіну Zigbee2mqtt

У полі Config вписуємо наступну конфігурацію, вказуємо адресу Н.А. сервера, ім'я та пароль від сервера mqtt, порт USB ZigBee Sniffer.

```
{
  "data_path": "/share/zigbee2mqtt",
  "devices": "devices.yaml",
  "groups": "groups.yaml",
  "homeassistant": true,
  "permit_join": false,
  "mqtt": {
    "base_topic": "zigbee2mqtt",
    "server": "mqtt://192.168.1.7:1883",
    "user": " mqttt ",
    "password": "12345678"
  },
  "serial": {
    "port": "/dev/ttyACM0"
  },
  "advanced": {
    "pan_id": 6754,
    "channel": 11,
    "network_key": [
      1,3,5,7,9,11,13,15,0,2,4 6,8,10,2,13
    ],
    "availability_blacklist": []
  },
  "ban": [],
  "whitelist": [],
  "queue": {},
  "socat": {
```

```

    "enabled": false,
    "master": "pty,raw,echo=0,link=/dev/ttyZ2M,mode=777",
    "slave": "tcp-listen:8485,keepalive,nodelay,reuseaddr,
keepidle=1,keepintvl=1,keepcnt=5",
    "restartdelay": 1,
    "initialdelay": 1,
    "options": "-d -d",
    "log": false
  }
}

```

Зберігаємо та робимо перезавантаження плагіну.

Тепер необхідно додати конфігурацію в файл `configuration.yaml`. Цей файл містить інтеграції, для яких потрібно завантажити додаткові конфігурації. Найпростішим варіантом редагування `configuration.yaml` є використання надбудови Studio Code Server.

Використовуючи плагін додаємо наступний код в файл конфігурації:

```

discovery:
mqtt:
  broker: 192.168.1.7
  port: 1883
  username: mqtt
  password: 12345678
  discovery: true
  discovery_prefix: homeassistant
sensor:
  - platform: mqtt
    name: Zigbee-Bridge state
    state_topic: "zigbee2mqtt/bridge/state"

```

```
icon: mdi:router-wireless
```

Для автоматичного пошуку нових пристроїв додано цей код:

```
# Zigbee2mqtt
input_boolean:
  zigbee_permit_join:
    name: Allow devices to join
    initial: off
    icon: mdi:cellphone-wireless
timer:
  zigbee_permit_join:
    name: Time remaining
    duration: 600 # Updated this to the number of seconds you wish
```

На головній сторінці створюємо картку для зручного додавання нових пристроїв. Щоб додати картку до інтерфейсу користувача, натискаємо «Menu», а потім вибираємо «Edit info panel». Натискаємо кнопку «Add Card» в нижньому правому куті та вибираємо «Conditional» із засобу вибору карток.

Додаємо в поле конфігурації цей код:

```
entities:
  - entity: input_boolean.zigbee_permit_join
  - entity: timer.zigbee_permit_join
  - entity: sensor.zigbee_bridge_state
show_header_toggle: false
title: Zigbee
type: entities
```

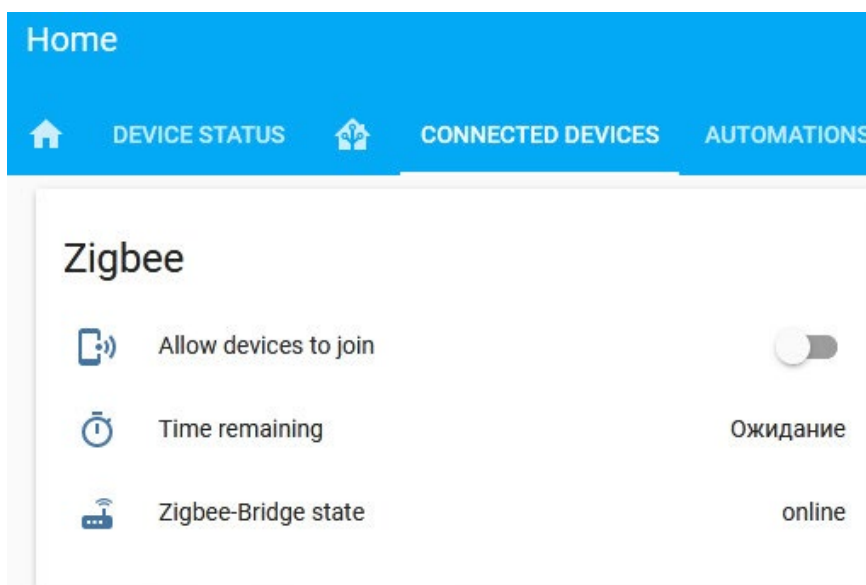


Рисунок 2.22 – Картка для додавання нових пристроїв

Перезавантажуємо Home Assistant. Тепер для пошуку нових пристроїв ZigBee достатньо натиснути перемикач «Allow devices to join». Через 10 хв. пошук припиниться. Тепер все готове для підключення нових пристроїв!

Підключаємо датчики моніторингу температури та вологості.

Включаємо пошук пристроїв «Allow devices to join». На пристрої натискаємо на кілька секунд кнопку mode «створення пари», до загоряння індикатора.

На датчику температури та вологості кнопка mode знаходиться зверху, через кілька секунд індикація припиниться. Пристрій з'явиться в розділі Інтеграції та в розділі Пристрої.

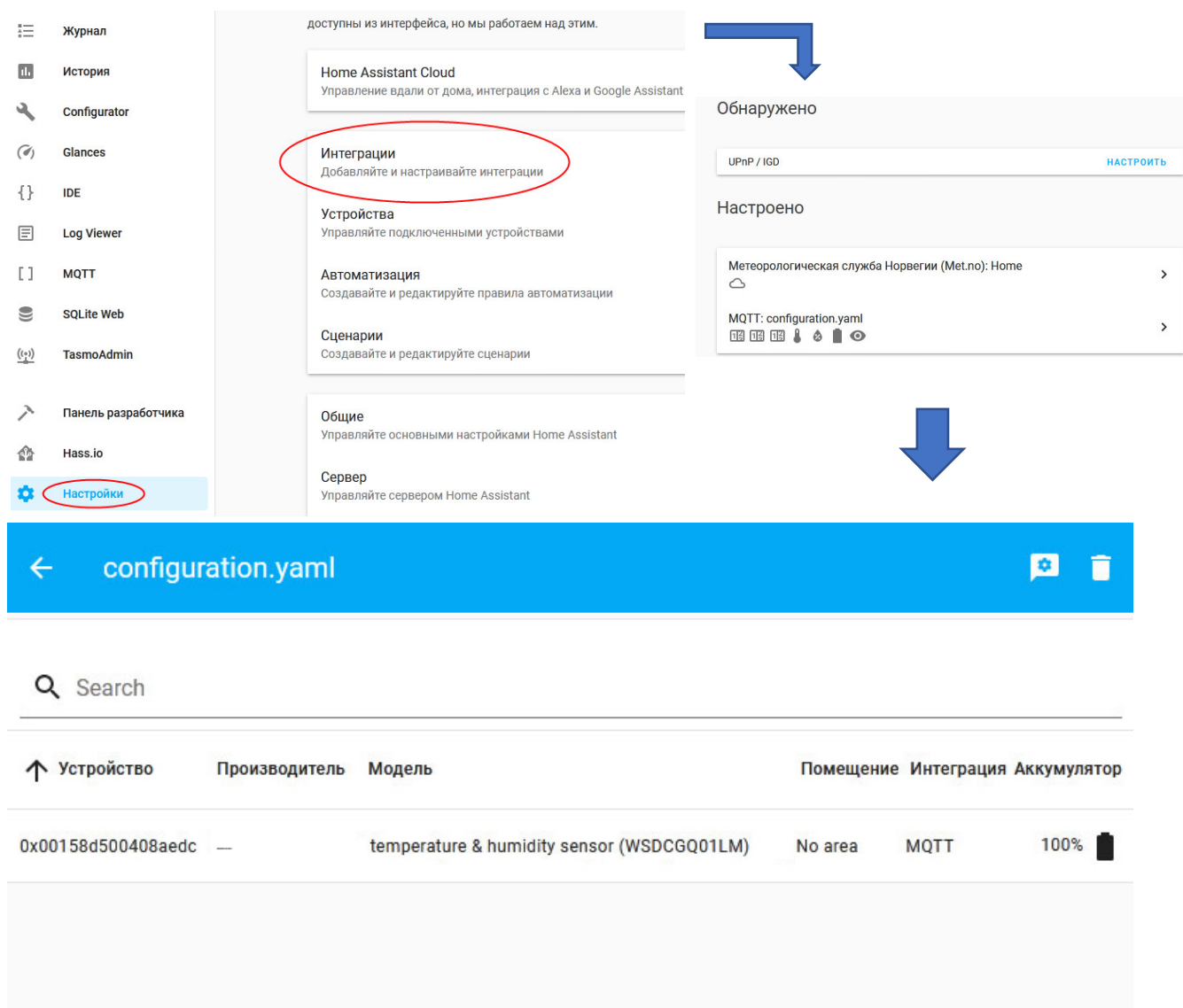


Рисунок 2.23 – Система побачила та додала новий пристрій

Натискаємо на поле с назвою датчика та потрапляємо в меню налаштування датчика. Кожному значенню необхідно додати зрозумілу назву. Для цього потрібно натиснути піктограму шестерні і ввести потрібні значення.

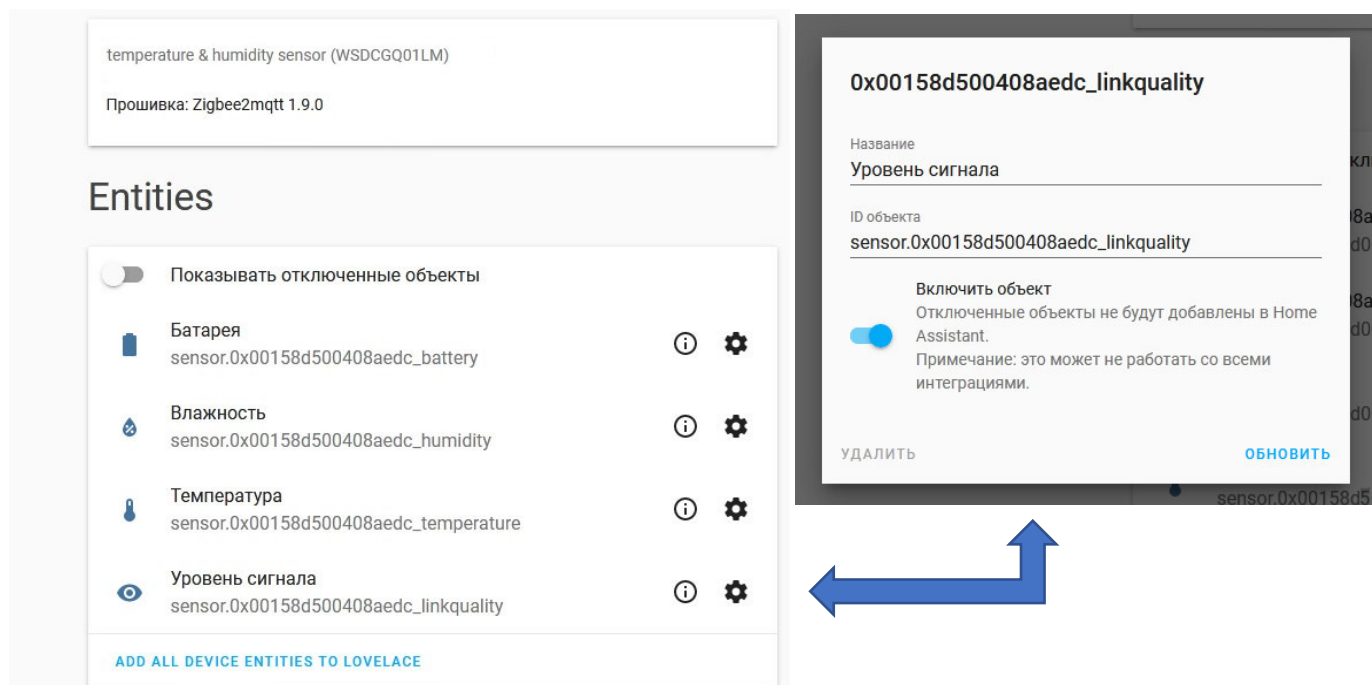


Рисунок 2.24 – Меню налаштування датчика

Тепер потрібно вивести дані нового пристрою на головний екран, це найпростіше зробити з меню натиснувши кнопку «ADD TO LOVELACE».

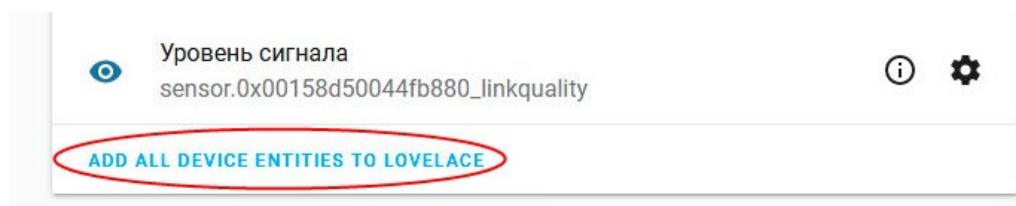


Рисунок 2.25 – Виведення даних датчика на головний екран

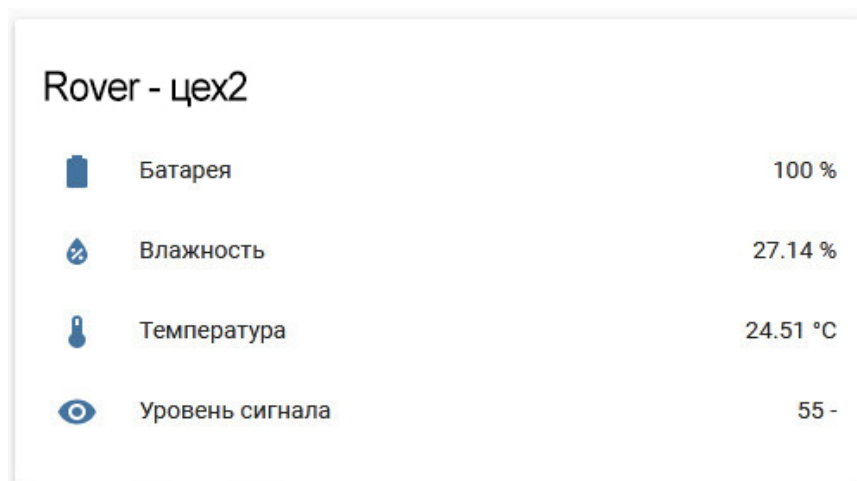


Рисунок 2.26 – Данні датчику на головній сторінці Home Assistant

У такий же спосіб підключено по одному датчику у двох верстатах Biesse Rover a 1632 та у двох KDT KS-828C.

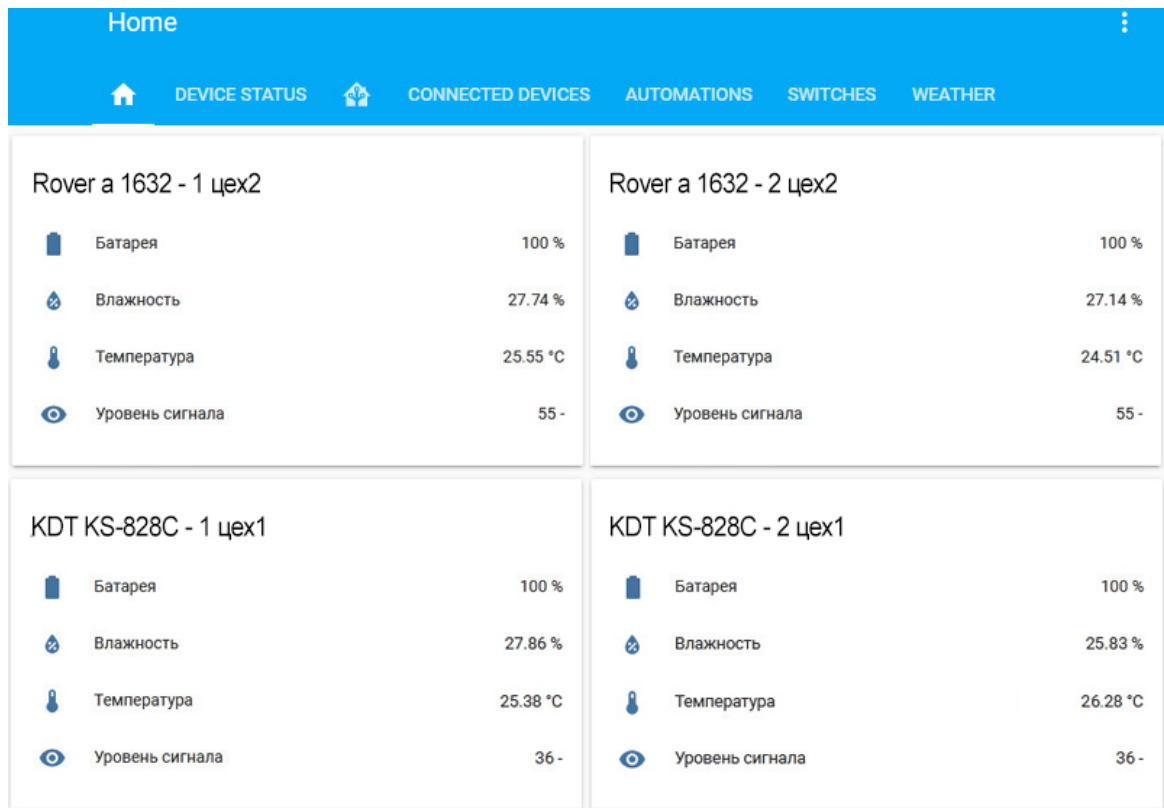


Рисунок 2.27 – Показники з 4 датчиків, 2 у верстатах Biesse Rover a 1632 та 2 у KDT KS-828C

Для зручності контролю всіх пристроїв ZigBee в панелі управління на головній сторінці Н.А. є розділ Zigbee Map. Мапа дає можливість в реальному часі спостерігати як формується мережа Zigbee та робити висновки щодо ефективності розташування датчиків та координатору, робити зміни в розташуванні пристроїв для уникнення «сірих зон».

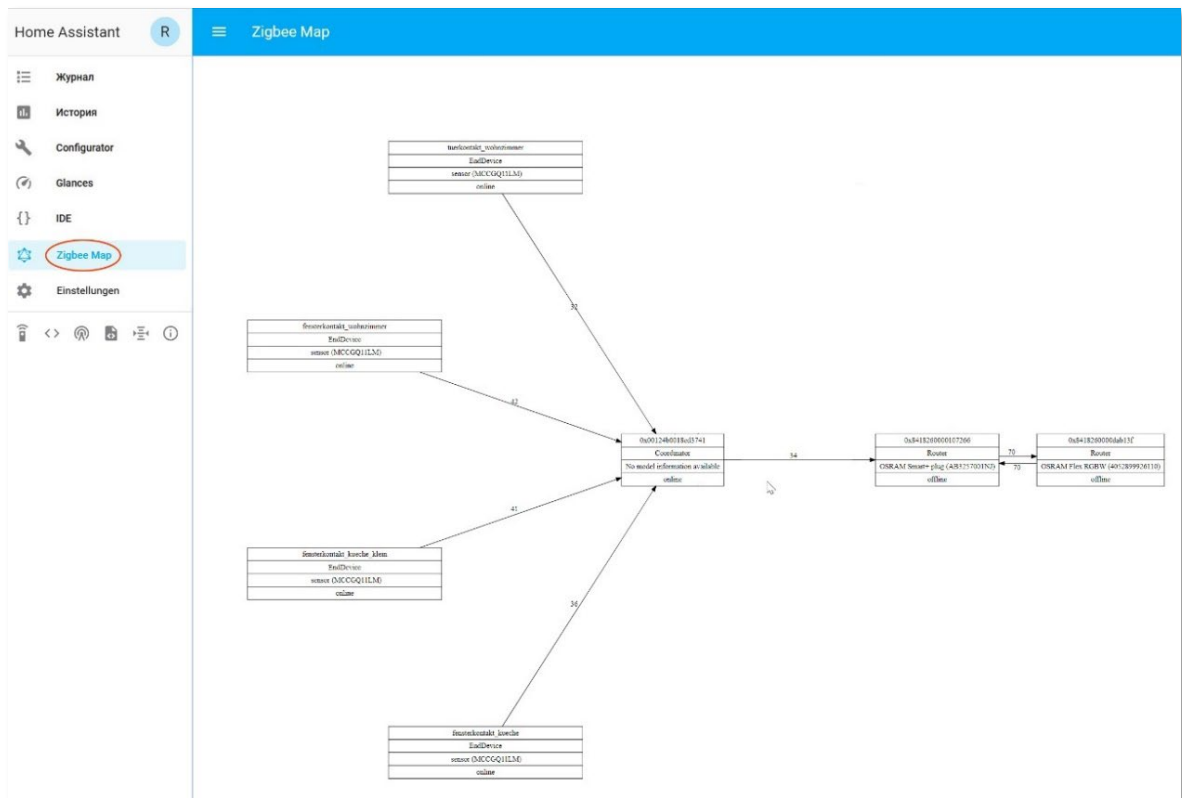


Рисунок 2.28 – Побудова зв'язків ZigBee мережі

## 2.5 Налаштування роботи попереджень о високих значеннях температури

Необхідною частиною роботи системи оповіщення є надсилання повідомлень користувачам. У Н.А. такі повідомлення називаються notifications і складається з двох базових типів повідомлень.

Перший – це внутрішні сповіщення (persistent notifications). Для їхнього відправлення використовується вбудований сервіс «persistent\_notification.create». Список таких повідомлень доступний через іконку дзвіночка в графічному інтерфейсі.

Другим, більш цікавим інструментом є вбудований модуль notify, який через встановлення додаткових модулів дозволяє передавати повідомлення, використовуючи сторонні платформи.

Розглянемо модуль для Telegram.

Підключення складається з кількох етапів:

- створення Telegram бота та отримання токена;
- дізнатися свій ID в Telegram;
- вписати в конфігураційні файли YAML Home Assistant необхідні налаштування;
- перевірити, що повідомлення надходять;
- почати використовувати (написати автоматизації які надсилають повідомлення або додати відправку повідомлень у вже існуючі автоматизації).

Для деяких інтеграцій обов'язково необхідно, щоб сервер Home Assistant був доступний зовні (із мережі), але у випадку Telegram це не обов'язково. Можна налаштувати надсилання повідомлень до Telegram незалежно від того, чи є доступ до Home Assistant зовні або немає.

Почнемо створення Telegram боту.

Для цього заходимо в програму Telegram на будь якому пристрої, реєструємось в системі та знаходимо бота під назвою BotFather. Пишемо /start для виведення доступних команд. Далі створюємо свого бота за допомогою команди / Newbot. Вибираємо ім'я, яке відображатиметься у списку контактів та адресу бота. Після вдалого створення бота BotFather надійшла відповідь з токеном для управління ботом [15].

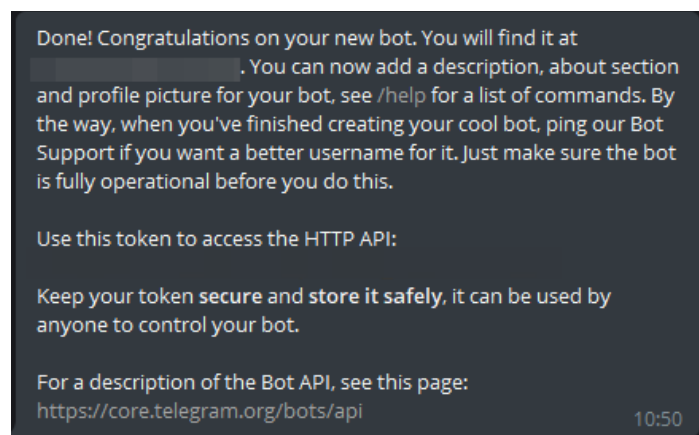


Рисунок 2.29 – Відповідь від BotFather, про вдале створення нового бота та його токен

Дізнаємося id бота.

Для цього можна скористатися іншими ботами, такими як @myidbot або @get\_id\_bot. Але також є інший спосіб. Переходимо в браузері за посиланням <https://api.telegram.org/bot-«id бота»BotFather/getMe>.

На сторінці є розділ де знаходиться запис "id":1234567890, це id бота, його необхідно зберегти.

Налаштовуємо Home Assistant для роботи оповіщень.

Відкриваємо файл configuration.yaml та додаємо туди наступний код:

```
telegram_bot:
```

```
- platform: polling
```

```
api_key: «API_KEY»
```

```
allowed_chat_ids:
```

```
- CHAT_ID_1 «example: 123456789 for the chat_id of a user»
```

```
- CHAT_ID_2 «example: -987654321 for the chat_id of a group»
```

Де !secret telegram\_bot\_api\_key - це той самий токен, який видав BotFather.

!secret\_telegram\_bot\_allowed\_chat\_id\_1 і !secret\_telegram\_bot\_allowed\_chat\_id\_2 - це id акаунтів, в які потрібно надсилати повідомлення.

Далі додаємо в інтеграцію notify два блоки, що вказують кому надсилати повідомлення. У результаті одержуємо сервіс notify.telegram\_1.

```
notify:
```

```
- platform: telegram
```

```
name: NOTIFIER_NAME
```

```
chat_id: CHAT_ID_1
```

Перевіряємо роботу оповіщень.

Потрібно зайти в розділ "Services" (<http://IP:8123/developer-tools/service>) в Home Assistant і почати вводити назву нового сервісу notify.me, вибрати його із списку і в додаткове поле вводимо:

message: asdf

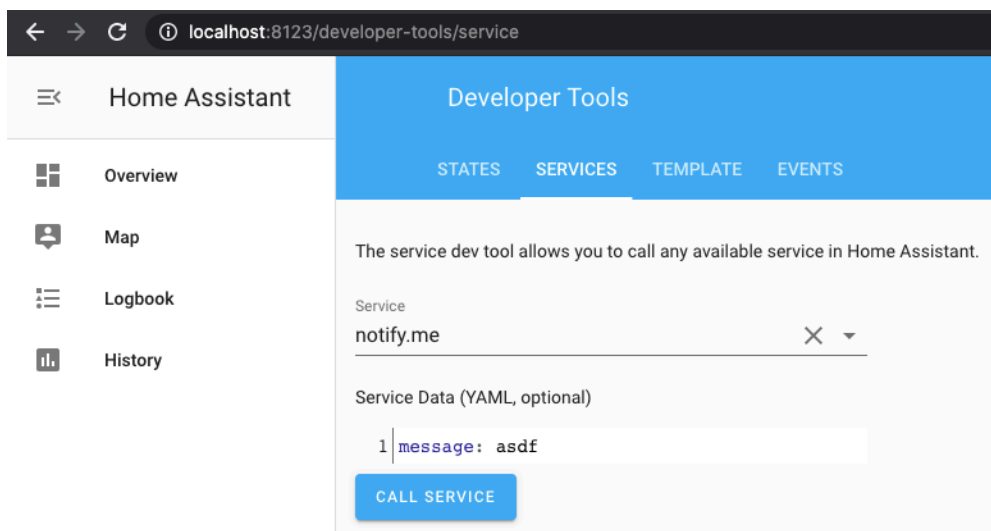


Рисунок 2.30 – Перевірка роботи оповіщень в Telegram

Після того як все налаштовано, при натисканні на кнопку «Call service», в Telegram прийде перевірочне повідомлення "asdf" від бота.

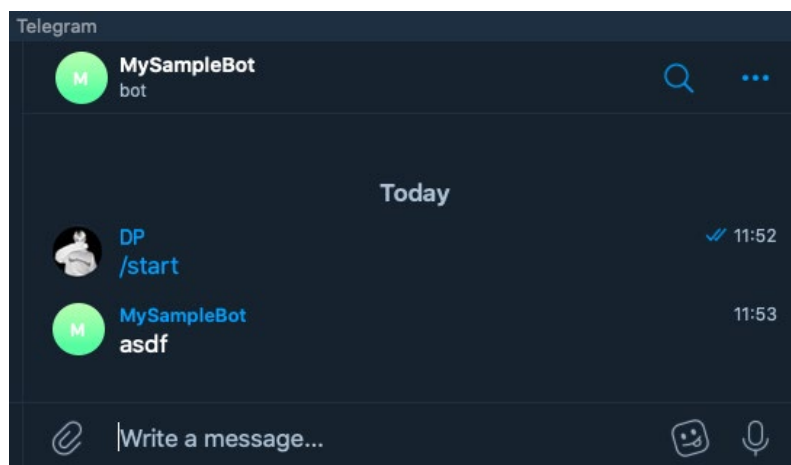


Рисунок 2.31 – Перевірка роботи оповіщень

Налаштування надсилання повідомлень до Telegram з автоматизацій.

Отже, зараз в Home Assistant з'явився сервіс, за допомогою якого можна надсилати повідомлення в Telegram. Далі його потрібно почати використовувати в автоматизаціях.

Структура автоматизацій складається із трьох елементів:

- trigger;
- condition;
- action.

"trigger" - описує умову коли автоматизація має запуститися;

"condition" - описує додаткові умови, які повинні бути щоб автоматизація виконала свою дію;

"action" - головне навіщо створюється автоматизація - виконання дії.

Існує два способи як можна створити або змінити автоматизацію:

- використовувати веб інтерфейс <http://IP:8123/config/automation>;
- працювати з текстовими файлами конфігурації.

Налаштування автоматизацій я виконував за допомогою редагування файлів конфігурацій.

Home Assistant має головний файл з налаштуваннями системи - файл `/config/configuration.yaml`. Цілком можливо записувати всі налаштування Н.А. тільки в цей файл і не використовувати інші. Але в файлі `configuration.yaml`, який створюється при першому запуску Home Assistant, пропонується створювати автоматизації в окремому файлі `automations.yaml`, що і було задіяно.

Тепер треба додати конфігураційний код автоматизації в файл `automations.yaml`:

```
- id: '1571074941908' # ID автоматизації
  alias: automation.temperatur_sensor # Ім'я автоматизації
  trigger: # Умови спрацювання автоматизації
    # Об'єкт, що визначає спрацювання
- entity_id: sensor.temperatur-1
```

```

from: Running # Початковий стан об'єкта
# Тип автоматизації – зміна стану об'єкта
platform: state
above: 50
condition: [] # Додаткові умови
action: # Виконувати автоматизацією дії
- service: notify.notify_send # Дія – виклик сервісу
data: # Аргументи, що передаються під час виклику сервісу
title: Rover a 1631-1 цех2 # Заголовок
message: "Зафіксовано високу температуру. " # Повідомлення

```

Додаємо автоматизації до інтерфейсу Н.А., для цього активуємо перемикач навпроти автоматизації.

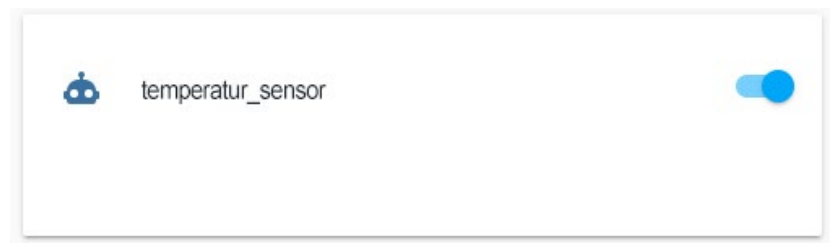


Рисунок 2.32 – Додавання автоматизації до інтерфейсу Н.А.

Перемикач впливає на те, включена ця автоматизація або ні. Якщо її вимкнути, то вона не виконуватиме своїх дій. Іноді буває зручно винести деякі автоматизації в інтерфейс, щоб можна було перевірити їх працездатність.

Якщо натиснути назву автоматизації в інтерфейсі Н.А., то відкривається вікно в якому є інформація коли ця автоматизація востаннє запускалася, також з'являється кнопка Execute для перевірки роботи автоматизації.

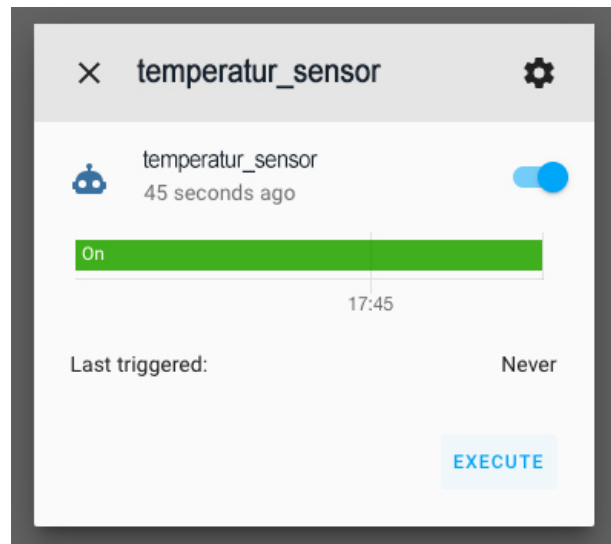


Рисунок 2.33 – Перевірка роботи автоматизації

Заходимо в автоматизацію `automation.temperatur_sensor` та натискаємо на кнопку `Execute`, результатом цього буде примусове спрацювання автоматизації та відправка повідомлення в Telegram бот «Rover a 1631-1 цех2 - Зафіксовано високу температуру.».

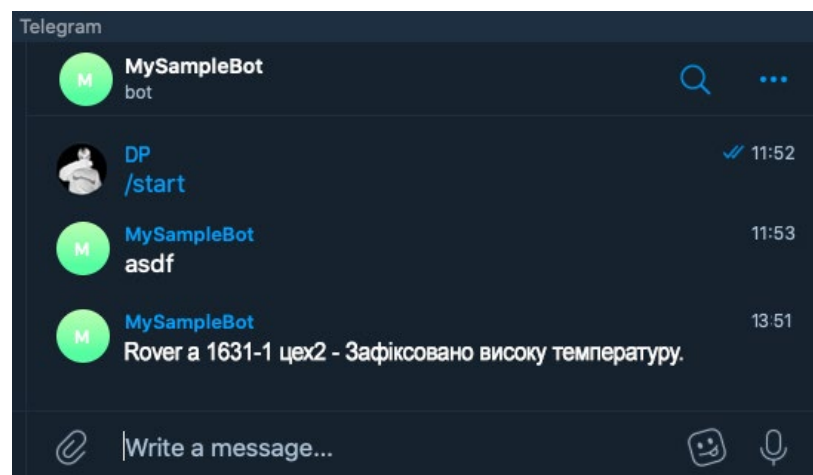


Рисунок 2.34 – Повідомлення в Telegram бот успішно доставлено

У нормальному режимі роботи, автоматизація спрацює коли температура на датчику буде більше 50 градусів і тільки потім відправить відповідне повідомлення.

Таку ж автоматизацію зроблено для інших датчиків які було встановлено в верстатах.

Завдяки автоматизаціям, відповідальна особа отримає повідомлення на свій телефон про високу температуру в електронно-комутаційній шафі на верстатах. Також повідомлення буде відображатися в системі Н.А. і записуватиметься в журнал подій для подальшої інспекції роботи відповідального персоналу виробництва. Обслуговування обладнання буде проводитися своєчасно, що в перспективі зможе подовжити безперебійну роботу верстатів.

## **2.6 Моніторинг стану температури після впровадження системи в виробництво**

Для перевірки ефективності роботи системи контролю, зроблено заміри температури в середині електронно-комутаційній шафі верстата Biesse Rover a 1632 протягом 10 робочих днів. Інформація о стані температури отримана з журналу показників датчика в системі Н.А. Показники температури датчика записуються в журнал системи кожну годину.

Для побудови графіку «Зміна показників температури в середині електронно-комутаційної шафи верстата Biesse Rover a 1632 протягом робочої зміни» вибрано діапазон з 8.00 до 17.00.

Для побудови графіку «Зміна показників температури в середині електронно-комутаційної шафи верстата Biesse Rover a 1632 протягом 10 робочих днів» взято середнє арифметичне значення температури за кожен робочий день та середнє значення температури повітря в цеху де розташовані верстати.

Графіки було побудована за допомогою програми Excel.

Таблиця 2.3 – Температура ( $t_1$ ) в середині електронно-комутаційної шафи верстата Biesse Rover a 1632 протягом робочої зміни

|                  |      |      |       |       |       |       |       |       |       |       |
|------------------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| $t_1$ в градусах | 20,4 | 37,7 | 42,2  | 47,8  | 51    | 36,3  | 51,4  | 53,5  | 55,7  | 55,2  |
| Час              | 8.00 | 9.00 | 10.00 | 11.00 | 12.00 | 13.00 | 14.00 | 15.00 | 16.00 | 17.00 |

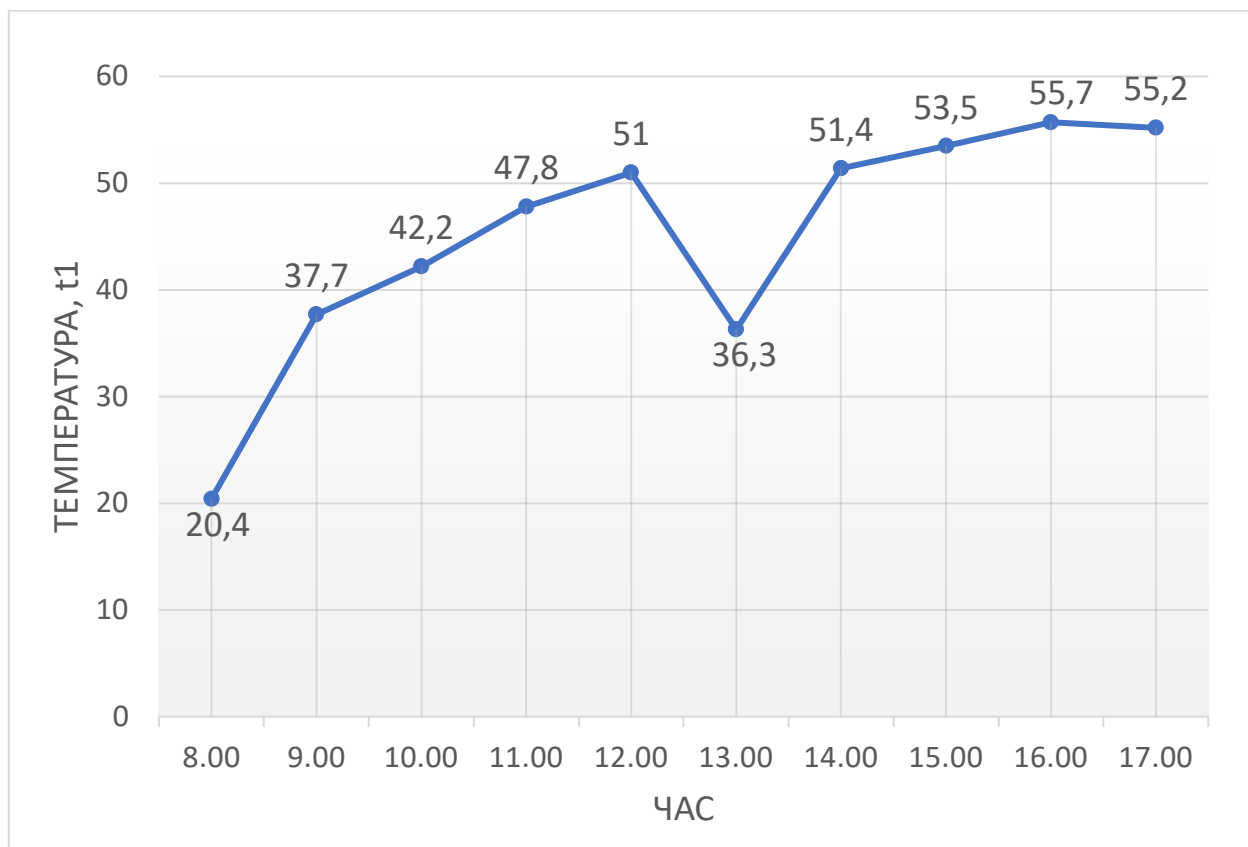


Рисунок 3.35 - Графік зміни показників температури в середині електронно-комутаційної шафи верстата Biesse Rover a 1632 протягом робочої зміни

Таблиця 2.4 – Середнє арифметичне значення температури за робочу зміну протягом 10 робочих днів ( $\mu t_2$ ) та середнє значення температури повітря в цеху ( $t_3$ )

|           |      |      |      |      |      |      |      |      |      |      |
|-----------|------|------|------|------|------|------|------|------|------|------|
| $\mu t_2$ | 45,1 | 51,8 | 54,2 | 45,4 | 51,6 | 54,7 | 45,9 | 51,3 | 54,5 | 46,2 |
| $t_3$     | 18,2 | 19,3 | 19,8 | 18,7 | 19,4 | 20   | 19,2 | 18,9 | 19,8 | 20,2 |
| День      | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   |

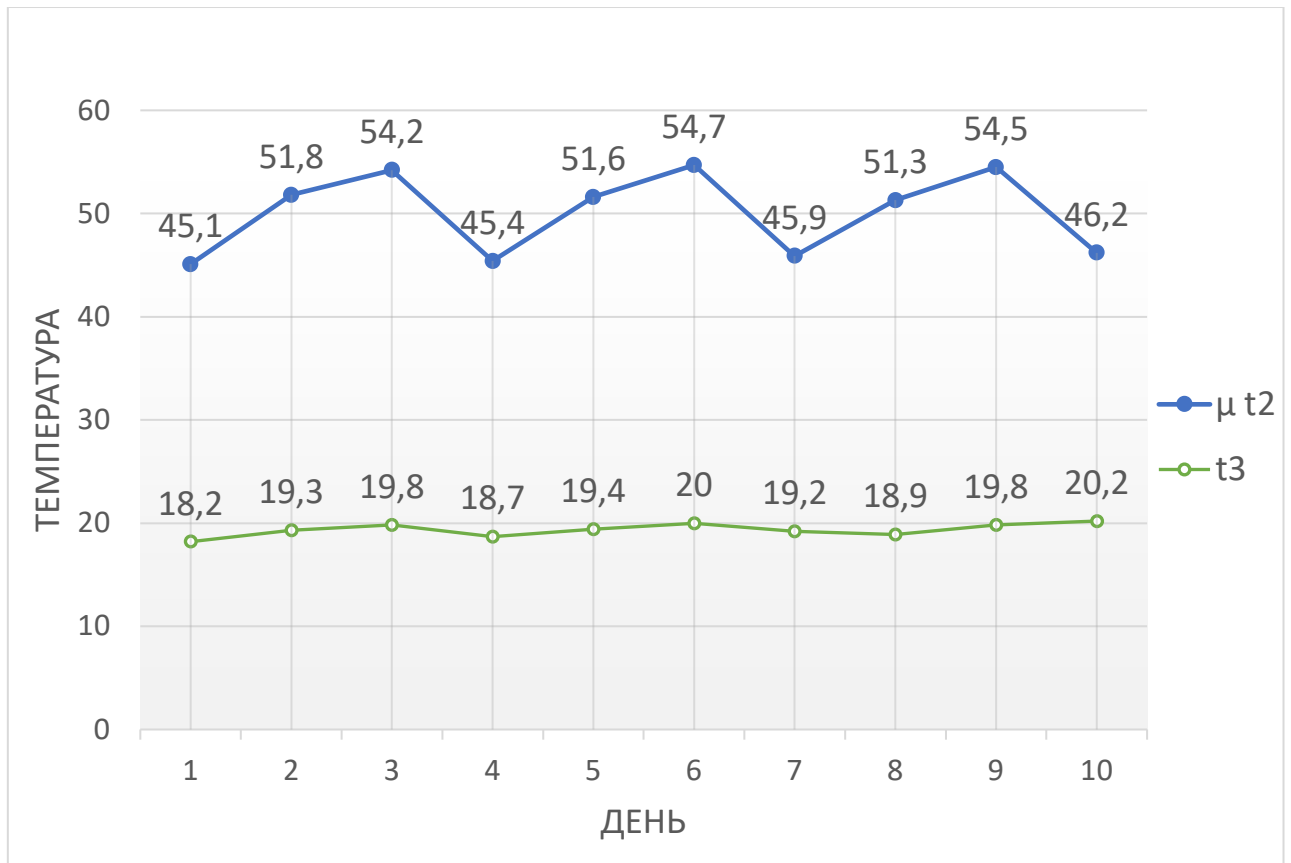


Рисунок 2.36 - Графік зміни показників середнє арифметичного значення температури за робочу зміну протягом 10 робочих днів ( $\mu t_2$ ) та середнє значення температури повітря в цеху ( $t_3$ )

На першому графіку ми бачимо збільшення температури в процесі роботи верстата на протязі всієї зміни, крім інтервалу з 12.00 до 13.00, коли проходить обідня перерва і верстати вимикають.

Температура на датчику встановленому в шафі підіймається вище 50 градусів, що класифікується як висока температура. За вимірами температури за допомогою інфрачервоного пірометра, пристрій показав, що температура електронних компонентів може бути на 60-70% більша, ніж середня температура в шафі, яку міряє датчик, це приблизно 90-95 градусів.



Рисунок 2.37 - Теплова карта в середині електронно-комутаційної шафи  
верстата Biesse Rover a 1632

На другому графіку ми бачимо зміну показників середнє арифметичного значення температури за робочу зміну протягом 10 робочих днів та середнє значення температури повітря в цеху. Кожні 3 дні середні показники температури повертаються до норми і становлять менше 50 градусів. Це пов'язано в тим, що в рамках обслуговування верстатів проводиться чищення системи охолодження обладнання. Але у зв'язку з високою пильністю біля верстатів, яка є нормою для меблевих виробництв, фільтри та канали системи охолодження швидко забруднюються.

Дослідження показало, що для ефективної роботи системи охолодження верстатів, потрібно робити її обслуговування кожен день після робочої зміни. Також можливо задіяти більш інноваційний підхід - це встановлення системи центральної аспірації типу «циклон», яка значно зменшить запиленість, але має дуже високу ціну реалізації та не виключає обслуговування системи охолодження, хоча і знизить частоту виконання цієї роботи.

## **3 УДОСКОНАЛЕННЯ СИСТЕМИ КОНТРОЛЮ ЗА СТАНОМ ОБЛАДНАННЯ**

### **3.1 Віддалений доступ до системи контролю із мережі інтернету**

Налаштування віддаленого доступу до Home Assistant за допомогою кастомного доповнення Dataplicity. За допомогою цього компонента можливо отримати віддалений доступ до сервера Home Assistant. З'являється адреса в інтернеті, на яку можливо зайти за допомогою браузера і побачити інтерфейс системи Home Assistant.

Це доповнення дозволяє налаштувати віддалений доступ навіть на сірому IP. Використання цього рішення – це один із найпростіших способів зробити віддалений доступ до сервера Home Assistant [17].

Для того, щоб з'явився віддалений доступ до сервера Н.А. потрібно зробити:

1. Встановити доповнення на сервер Home Assistant.
2. Підключити компонент.
3. Зареєструватись на сервісі <https://www.dataplicity.com>.
4. Налаштувати компонент та сервіс.

Встановлення доповнення на сервер, для цього потрібно зайти на сторінку кастомного доповнення на GitHub <https://github.com/AIT/Dataplicity>, зайти в розділ "Релізи" та завантажити zip файл з вихідним кодом компонента.

Zip файл потрібно розархівувати і всі файли з папки Dataplicity-1.0.0/custom\_components/dataplicity/ потрібно перенести в папку config/custom\_components/dataplicity/

Папка custom\_components повинна бути в тій же директорії, що і файл configuration.yaml.

Ось така структура файлів і папок повинна бути:

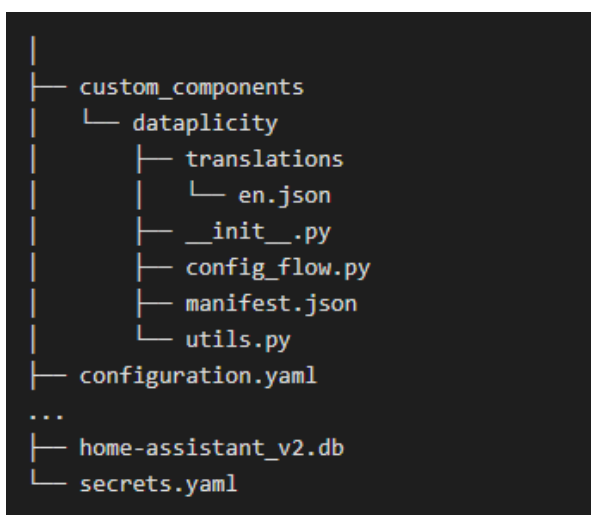


Рисунок 3.1 – Структура папок для роботи компонента

Після розміщення файлів потрібно перезавантажити систему Home Assistant.

Після того, як компонент встановлений його потрібно підключити. Для цього заходимо до розділу Configuration в Home Assistant, далі Integrations і натискаємо кнопку Add integration. З'явиться віконце в якому треба вибрати інтеграцію Dataplicity, після вибору цієї інтеграції з'являється віконце:

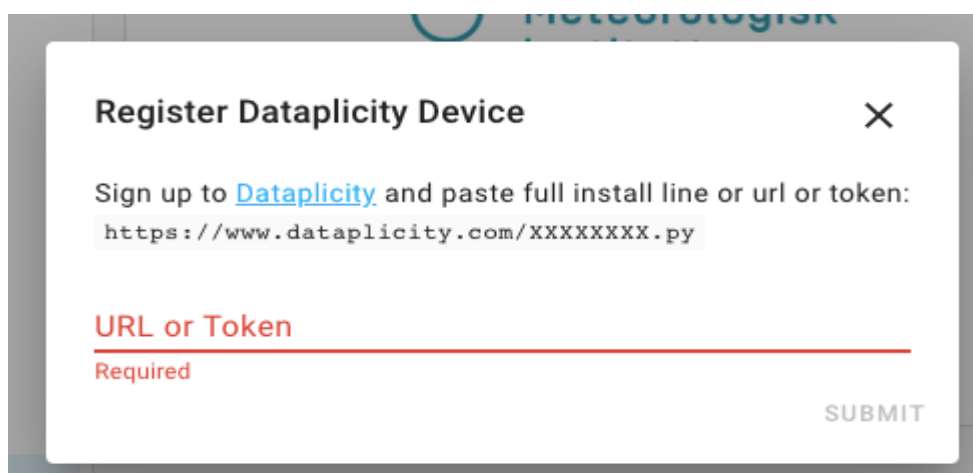


Рисунок 3.2 – Віконце налаштування роботи інтеграції

У це віконце необхідно ввести дані від сервісу Dataplicity. Слово "Dataplicity" у цьому віконці - це посилання на сайт <https://www.dataplicity.com/>.

Після реєстрації на сайті з'явиться рядок коду який потрібно ввести в віконці кастомного компонента в Home Assistant.

Приклад рядка, який може показати сайт:

```
curl -s https://www.dataplicity.com/qlqyu6nt.py | sudo python
```

З цього рядка потрібна тільки адреса:

```
https://www.dataplicity.com/qlqyu6nt.py
```



Рисунок 3.3 – Вводимо отриману адресу

Через деякий час з'явиться віконце яке оповістить про успішний результат налаштування. Залишилося виконати останню дію, увімкнення wormhole. Потрібно перейти за посиланням за допомогою кнопки Device settings у віконці. Вмикаємо перемикач Wormhole для роботи віддаленого доступу:

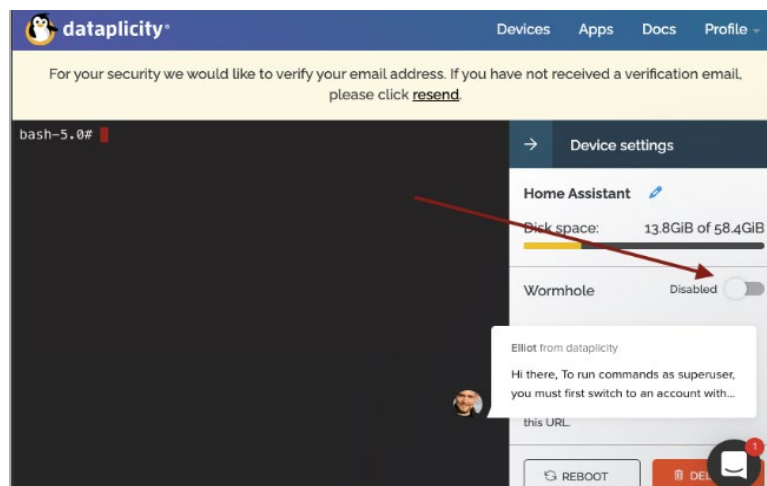


Рисунок 3.4 – Активація функція віддаленого доступу

З'явиться адреса, яка має такий вигляд: <https://consanguine-wasp-4419.dataplicity.io/>. Якщо зайти за цією адресою, то відкриється інтерфейс серверу Home Assistant.

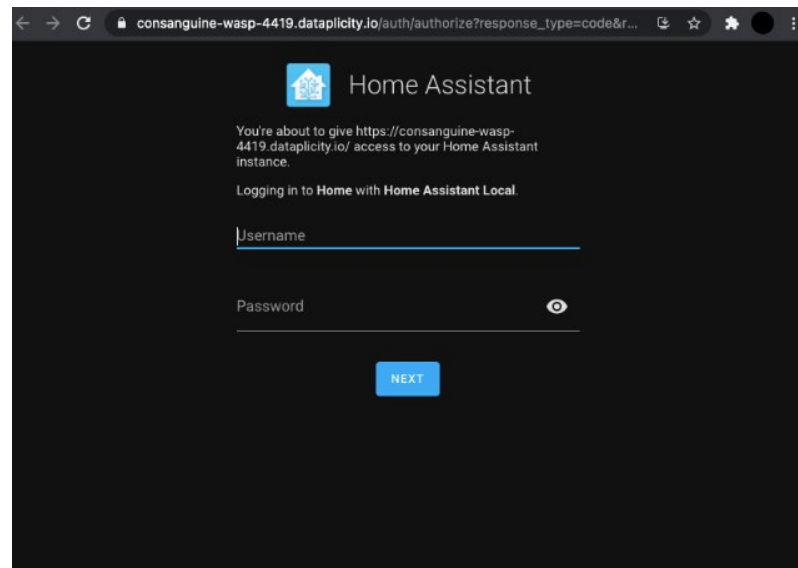


Рисунок 3.5 – Віддалений доступ до інтерфейсу системи

Ця адреса – це і є адреса, за якою можливо зайти на сервер Home Assistant з будь-якої точки світу. На цьому налаштування віддаленого доступу до системи контролю за станом обладнання закінчилося успішно.

### 3.2 Перспективи розвитку системи контролю за станом обладнання

Система вийшла універсальною і масштабованою, завдяки роботі системи на базі Home Assistant є можливість додавати безліч готових інтеграцій з іншими IoT пристроями. Також апаратна платформа Raspberry Pi 4 має достатній запас продуктивності і зможе без проблем обслуговувати сотні датчиків, комутаторів та яких завгодно інших розумних пристроїв.

У перспективі в роботу системи можливо додати прилади для контролю за станом мережевої напруги, які будуть збирати інформацію про стан

живлення, та робити повідомлення коли датчик зафіксує відхилення від норми.

Також важливою необхідністю є слідкування за станом системи автоматичного змазування вузлів верстата. Рівень змазуючих рідин може стати критично низьким, що в свою чергу погано впливає на роботу рухливих механізмів та може спричинити зайве навантаження на тягові вузли верстата.

Для вирішення цієї проблеми можливо застосування безконтактних датчиків рівня рідини, наприклад датчик ХКС-Y25.

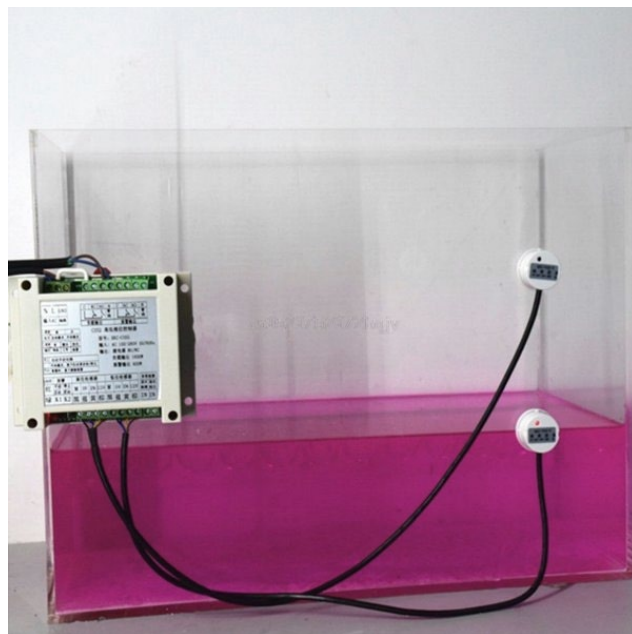


Рисунок 3.6 – Приклад встановлення датчиків ХКС-Y25 на ємність з рідиною

Датчик застосовується для безконтактного вимірювання рівня агресивних, стерильних, токсичних рідин і розчинників. Повинен бути встановлений на неметалевої стінці контейнера.

Це тільки невелика частка тих рішень, які можливо здійснити на базі впровадженої в виробництво системи контролю за станом обладнання.

## ВИСНОВКИ

Описано сучасні тенденції розвитку цифрових систем управління виробництвом. Провідним трендом цифровізації являється Індустрія 4.0 (Industry 4.0) «Четвертої промислової революції», описано характерні риси, та провідні технології, одна з яких являється Інтернет речей (Internet of Things, IoT). Основна задача інтернет речей: реагування на зміни, отримання інформації, відправка інформації в систему обробки даних, отримання зворотних команд та виконання дій (якщо це можливо).

Описано роботу ZegBee протоколу, його переваги в порівнянні з іншими рішеннями, принцип формування мережі, mesh-топологія, маршрутизація, пошук маршрутів та інше.

Встановлено та налаштовано роботу системи контролю за станом обладнання на базі Home Assistant – це універсальна безкоштовна операційна система для керування пристроями інтернету речей різних виробників, а також їхньої автоматизації.

Наведено характеристики датчику вимірювання температура та вологості CC2530 Efekta TNP\_LR\TNP, описано етапи збирання датчика (пайка елементів на друковану плату, прошивка радіомодуля, збирання в корпус). Ціна всіх компонентів датчика складає 14\$, що помітно дешевше ніж готові рішення.

Описано роботу датчику, та їх основне завдання – це забезпечити перетворення інформації про зовнішнє середовище в дані, придатні для передачі їх засобам обробки.

Налагоджена робота датчиків з системою контролю, налаштовано інтерфейс користувача. Додано автоматизацію для роботи оповіщень про високу температуру, автоматизація спрацює коли температура на датчику буде більше 50 градусів і система відправить відповідне повідомлення в заздалегідь створений Telegram бот.

Завдяки автоматизації відповідальна особа отримає повідомлення на свій телефон про високу температуру в електронно-комутаційній шафі на верстатах. Також повідомлення буде відображатися в системі Home Assistant і записуватиметься в журнал подій для подальшої інспекції роботи відповідального персоналу виробництва. Обслуговування обладнання буде проводитися своєчасно, що в перспективі зможе подовжити безперебійну роботу верстатів.

Проведено моніторинг стану температури після впровадження системи в виробництво для перевірки ефективності роботи системи контролю. Зроблено заміри температури в середині електронно-комутаційній шафі верстата Biesse Rover a 1632 протягом 10 робочих днів. Побудовано графіки зміни показників температури, мінімальне середнє арифметичне значення температури становить 45,1 градусів, максимальне 54,7 - що класифікується як висока температура. За вимірами температури за допомогою інфрачервоного пірометра, пристрій показав, що температура електронних компонентів може бути на 60-70% більша, ніж середня температура в шафі, яку міряє датчик, це приблизно 90-95 градусів. Дослідження показало, що для ефективної роботи системи охолодження верстатів, потрібно робити її обслуговування кожен день після робочої зміни.

Були запропоновані удосконалення системи контролю за станом обладнання, а саме віддалений доступ до системи контролю із мережі інтернету. Наведено перспективи розвитку системи контролю за станом обладнання.

Система вийшла універсальною і масштабованою, завдяки роботі системи на базі Home Assistant є можливість додавати безліч готових інтеграцій з іншими IoT пристроями. Також апаратна платформа Raspberry Pi 4 має достатній запас продуктивності і зможе без проблем обслуговувати сотні датчиків, комутаторів та яких завгодно інших розумних пристроїв.

## ПЕРЕЛІК ПОСИЛАНЬ

1. **Василенко О.В.** «М'яка» цифровізація підприємств в українських реаліях [Текст] / О.В. Василенко, Г.В. Сніжної, В.І. Рева, С.А. Івченко // VII Міжнар. наук.-техн. конф. Комп'ютерне моделювання та оптимізація складних систем. – Дніпро: УДХТУ, 2021. – С. 126-128.
2. **Доценко С. І.** Аналіз методологічних основ формування інформаційно-управляючих систем підприємств в Індустрії 3.0: рух до Індустрії 4.0 [Текст] / С.І. Доценко, О.О. Ілляшенко, С. Каменський, Д.Р. Купрейшвілі, В.С. Харченко // Радіoeлектронні і комп'ютерні системи. – 2019. – № 2. – С. 29-44. DOI.: 10.32620/reks.2019.2.03.
3. **Radhakisan B.** Cyber-physical systems [Текст] / B. Radhakisan, G. Helen // The impact of control technology. – 2011. – No. 12.1. – P. 161-166.
4. Щодо національної політики імплементації нормативних документів, які становлять основу індустрії 4.0. [Електронний ресурс]. – Режим доступу: <https://niss.gov.ua/doslidzhennya/ekonomika/schodo-nacionalnoi-politiki-implementacii-normativnikh-dokumentiv-yaki>.
5. Про затвердження Національної економічної стратегії на період до 2030 року. [Електронний ресурс] – Режим доступу: <https://www.kmu.gov.ua/npas/pro-zatverdzhennya-nacionalnoyi-eko-a179>.
6. Пропозиції АППАУ та руху 4.0 до Національної економічної стратегії 2030. [Електронний ресурс] – Режим доступу: <https://www.industry4ukraine.net/publications/propozycji-appau-ta-ruhu-4-0-do-naczionalnoyi-ekonomichnoyi-strategiyi-2030/>.
7. Мережі ZigBee [Електронний ресурс]. – Режим доступу: <http://www.rtlsnet.ru/technology/view/3>.
8. Internet of Things, IoT [Електронний ресурс]. – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/internet-veschej-internet-of-things-iot>.

9. Industry 4.0 [Електронний ресурс]. – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/industry-4>.
10. DIY Zigbee датчик температури [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/company/ruvds/blog/659263/>.
11. Install Home Assistant Operating System [Електронний ресурс]. – Режим доступу: <https://www.home-assistant.io/installation/raspberrypi>.
12. Встановлення Home Assistant [Електронний ресурс]. – Режим доступу: <https://1wire.com.ua/ru/umnij-dom-xiaomi-zigbee-na-raspberry-pi-home-assistant.html>.
13. Zigbee2mqtt: show the networkmap in home assistant [Електронний ресурс]. – Режим доступу: <https://community.home-assistant.io/t/zigbee2mqtt-show-the-networkmap-in-home-assistant/89116>.
14. Telegram [Електронний ресурс]. – Режим доступу: <https://www.home-assistant.io/integrations/telegram/>.
15. Повідомлення Home Assistant в Telegram [Електронний ресурс]. – Режим доступу: <https://dzen.ru/media/id/5f5bea452c75477b342dab/uvedomleniia-home-assistant-v-telegram-5f849f46cae5a83b553c4d78>.
16. Fourth Industrial Revolution [Електронний ресурс]. – Режим доступу: [https://en.wikipedia.org/wiki/Fourth\\_Industrial\\_Revolution](https://en.wikipedia.org/wiki/Fourth_Industrial_Revolution).
17. Налаштування віддаленого доступу до Home Assistant за допомогою кастомного компонента Dataplicity [Електронний ресурс]. – Режим доступу: <https://ivan.bessarabov.ru/wiki/home-assistant/remote-access/custom-component-dataplicity>.
18. Щодо національної політики імплементації нормативних документів, які становлять основу індустрії 4.0 [Електронний ресурс]. – Режим доступу: <https://niss.gov.ua/doslidzhennya/ekonomika/schodo-nacionalnoi-politiki-implementacii-normativnikh-dokumentiv-yaki>.
19. Про затвердження Національної економічної стратегії на період до 2030 року [Електронний ресурс]. – Режим доступу: <https://www.kmu.gov.ua/npas/pro-zatverdzhennya-nacionalnoyi-eko-a179>.

- 20.Пропозиції АППАУ та руху 4.0 до Національної економічної стратегії 2030. [Електронний ресурс] – Режим доступу: <https://www.industry4ukraine.net/publications/propozycji-i-appau-ta-ruhu-4-0-do-naczionalnoyi-ekonomichnoyi-strategiyi-2030/>.
- 21.Smart MAIC, Моніторинг ресурсів на всі випадки! [Електронний ресурс] – Режим доступу: <https://smart-maic.com/uk/>.
- 22.Український мобільний оператор запустив в комерційну експлуатацію мережу «Інтернет речей» NB-IoT – Freetrack [Електронний ресурс] – Режим доступу: <https://freetrack.com.ua/ukrajinskyj-mobilnyj-operator-zapustyv-v-komertsijnu-ekspluatatsiyu-merezhу-internet-rechej-nb-iot/>.
- 23.INTECH. Надійні рішення для промислових підприємств [Електронний ресурс] – Режим доступу: <https://www.intech-ukraine.com/ua/>.
24. What Are IoT Devices : Definition, Types, and 5 Most Popular Ones for 2023 [Електронний ресурс] – Режим доступу: <https://www.simplilearn.com/iot-devices-article>.
- 25.YAML [Електронний ресурс] – Режим доступу: <https://www.home-assistant.io/docs/configuration/yaml/>.
- 26.Configuration.yaml [Електронний ресурс] – Режим доступу: <https://www.home-assistant.io/docs/configuration/>.
27. Troubleshooting your configuration [Електронний ресурс] – Режим доступу: <https://www.home-assistant.io/docs/configuration/troubleshooting>.
- 28.Common Tasks – Core [Електронний ресурс] – Режим доступу: <https://www.home-assistant.io/common-tasks/core/#configuration-check>.
- 29.IIOT в концепции индустрия 4.0 [Електронний ресурс] – Режим доступу: <https://www.industrie4-0.com.ua/iiot-koncepcii-industrija-4-0.html>.
- 30.Чому так важлива четверта промислова революція? [Електронний ресурс] – Режим доступу: <https://idcard.com.ua/ua/blog/why-is-the-fourth-industrial-revolution-so-important/>.