

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень

Мережевий шлюз Інтернету речей на базі Raspberry Pi  
Назва теми

КВРКІ 200226.21.04.83 ПЗ  
Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Виконав: студент IV курсу, група KI2-21-4



Підпис

Мирослав АНДРУЩАК

Ініціали, прізвище

Керівник



Підпис, дата

Василь ЯЦКІВ

Ініціали, прізвище

Нормоконтролер



Підпис, дата

Тетяна КИСІЛЬ

Ініціали, прізвище

До захисту допускаю:  
зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем



Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

«16» червня 2025 р.

Хмельницький 2025

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 11 ” 02 2025 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Мирославу АНДРУЩАКУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Мережевий шлюз Інтернету речей на базі Raspberry Pi

Керівник проекту (роботи) Василь ЯЦКІВ, д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 09.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

Дослідження області Інтернету речей та практична постановка задачі

Проектування шлюзу Інтернет речей на базі Raspberry Pi

Програмно-апаратна реалізація шлюзу Інтернет речей на основі Raspberry Pi та його

тестування

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) \_\_\_\_\_

Загальна схема апаратного забезпечення шлюзу Інтернет речей на базі Raspberry Pi

Загальна схема ролі шлюзу Інтернет речей у програмній частині архітектури

Схема електрична структурна підключення пристроїв до Raspberry Pi 4B

6. Консультанти розділів дипломного проекту (роботи)

| Розділ        | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|---------------|---|----------------|------------------|
|               |   | завдання видав | завдання прийняв |
| Нормоконтроль | Тетяна КИСІЛЬ, доцент кафедри КІС         |                |                  |
| Антиплагіат   | Андрій НІЧЕПОРУК, доцент кафедри КІС      |                |                  |

7. Дата видачі завдання « 11 » 02 2025 р.

**КАЛЕНДАРНИЙ ПЛАН**

| №з/п | Назва етапів (розділів) дипломного проекту (роботи)   | Термін виконання етапів проекту (роботи) | Примітка |
|------|---|--|----------|
| 1    | Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником                                  | 12.02.2025                               | виконано |
| 2    | Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження | 20.02.2025                               | виконано |
| 3    | Робота над розділом 1 – дослідження області Інтернету речей та практична постановка задачі                            | 09.03.2025                               | виконано |
| 4    | Робота над розділом 2 – проектування шлюзу Інтернету речей на базі Raspberry Pi                                       | 15.04.2025                               | виконано |
| 5    | Робота над розділом 3 – програмно-апаратна реалізація шлюзу Інтернету речей на основі Raspberry Pi та його тестування | 02.05.2025                               | виконано |
| 6    | Оформлення пояснювальної записки згідно вимог   | 23.05.2025                               | виконано |
| 7    | Попередній захист ВКР   | 26.05.2025                               | виконано |
| 8    | Захист ВКР на засіданні ЕК  | Червень 2025 року                        |          |

Студент

Підпис

Мирослав АНДРУЦАК

Ініціали, прізвище

Керівник роботи

Підпис

Василь ЯЦКІВ

Ініціали, прізвище



## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Мережевий шлюз Інтернету речей на базі Raspberry Pi».

Автор роботи: Мирослав АНДРУЩАК.

Керівник роботи: Яцків Василь Васильович.

Пояснювальна записка: 68 с., 73 рис., 5 дод., 63 джерела.

Графічна частина: 3 креслення.

ІНТЕРНЕТ РЕЧЕЙ, ПАОП, RASPBERRY PI, ДАВАЧ, ІНТЕРФЕЙС, ПРОТОКОЛ, GPIO, МЕРЕЖА, ШЛЮЗ.

Метою дипломної роботи є дослідження та загальний опис області сфери інформаційних технологій, іменованої як Інтернет речей: її суті, сенсу, структури та ролі у сфері інформаційних технологій; виконання аналізу, проектування та програмно-апаратної реалізації структурного елемента у системі Інтернету речей, званого як шлюз; проведення апробації шлюзу Інтернету речей у системі та, за умови наявності недоліків, їх усунення при можливості.

Об'єктом дослідження є функціонування шлюзу Інтернет речей у екосистемі.

Предметом дослідження є аналіз методів взаємодії шлюзу Інтернет речей із пристроями та обчислювальними системами.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.







Підпис студента

30.05.2025

Дата

## ЗМІСТ

|  |    |
|--|----|
| <b>ВСТУП</b> .....   | 3  |
| <b>1 ДОСЛІДЖЕННЯ ОБЛАСТІ ІНТЕРНЕТУ РЕЧЕЙ ТА ПРАКТИЧНА ПОСТАНОВКА ЗАДАЧІ</b> .....                            | 4  |
| 1.1 Чим є шлюз Інтернету речей: його опис та функціональність.....   | 4  |
| 1.2 Як шлюз Інтернету речей працює .....   | 11 |
| 1.3 Програмно-апаратні рішення шлюзу Інтернету речей.....  | 11 |
| 1.4 Про Raspberry Pi .....   | 15 |
| 1.5 Аналіз й опис обраної платформи Raspberry Pi .....   | 17 |
| 1.6 Висновок до першого розділу.....   | 20 |
| <b>2 ПРОЕКТУВАННЯ ШЛЮЗУ ІНТЕРНЕТУ РЕЧЕЙ НА БАЗІ RASPBERRY PI</b> .....                                       | 21 |
| 2.1 Функціональні вимоги шлюзу Інтернету речей.....  | 21 |
| 2.2 Основні елементи шлюзу Інтернету речей.....  | 23 |
| 2.3 Висновок до другого розділу .....  | 29 |
| <b>3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ШЛЮЗУ ІНТЕРНЕТУ РЕЧЕЙ НА ОСНОВІ RASPBERRY PI ТА ЙОГО ТЕСТУВАННЯ</b> ..... | 30 |
| 3.1 Встановлення необхідного програмного середовища та конфігурація Raspberry Pi .....                       | 30 |
| 3.1.1 Вибір, встановлення та налаштування операційної системи .....  | 30 |
| 3.1.2 Підключення девайсів до плати Raspberry Pi.....  | 37 |
| 3.1.3 Встановлення програмних інструментів для взаємодії шлюзу з девайсами системи Інтернету речей.....      | 39 |
| 3.2 Реалізація програмного забезпечення для роботи шлюзу .....   | 46 |
| 3.2.1 Створення та тестування програмного забезпечення для комунікації з девайсами .....                     | 46 |
| 3.2.2 Налаштування віртуальної приватної мережі – VPN.....   | 52 |

|          |      |                   |   |          |   |               |       |         |
|----------|------|-------------------|---|----------|---|---------------|-------|---------|
|          |      |                   |   |          | КвРКІ 200226.21.04.83 ПЗ                            |               |       |         |
| Зм.      | Арк. | Нодокум.          | Підпис  | Дата     | Мережевий шлюз Інтернету речей на базі Raspberry Pi | Літера        | Арквш | Арквшів |
| Виконав  |      | Мирослав АНДРУЩАК |  | 16.06    |   | y             |       | 68      |
| Перевір. |      | ВасильЯЦКІВ       |  | 16.06    |   |               |       |         |
| Н.контр. |      | Тетяна КИСІЛЬ     |  | 16.06.21 | Пояснювальна записка                                | ХНУ, КІ2-21-4 |       |         |
| Затвер   |      | Ольга ПАВЛОВА     |  | 16.06.21 |   |               |       |         |



## ВСТУП

Інтернет речей (англ. *Internet of things; IoT*) – концепція взаємодії фізичних об'єктів, що з допомогою відповідних апаратних та програмних рішень об'єднані в мережу для взаємодії між собою в ній з певною ціллю. Заснована Кевіном Ештоном й ним же почата з розміщення RFID-мікросхем у продуктах для відстеження їх доставки концепція у нинішній час є досить перспективною та вартісною областю інформаційних технологій, що може дати досить значущий та практичний поштовх у розширенні сфер використання цієї самої області.

Актуальність дослідження обумовлена все більшим й більшим розповсюдженням кіберфізичних систем у житті людини. У сфері інформаційних технологій Інтернет речей прийнято рахувати досить перспективним підрозділом даної дисципліни, який закликаний підвищити рівень можливостей людини й надати змогу розкривати процес взаємодії у концепції «людина-об'єкт» більш широко та об'ємно. У даній роботі буде розглянуто та спроектовано елемент архітектури Інтернету речей, котрий являється зв'язуючою ланкою між пристроями, безпосередньо взаємодіючими із навколишнім середовищем, й обчислювальними системами, призначеними оперувати даними із навколишнього середовища за необхідними для користувача методами. Назва елемента – шлюз Інтернету речей.

Об'єктом дослідження є функціонування шлюзу Інтернет речей у екосистемі.

Предметом дослідження є аналіз методів взаємодії шлюзу Інтернету речей із пристроями та обчислювальними системами.

Метою дослідницької роботи є:

1. Загальний опис області сфери інформаційних технологій, іменованої як Інтернет речей: її суті, сенсу, структури та ролі у сфері інформаційних технологій.
2. Виконання аналізу, проектування та програмно-апаратної реалізації структурного елемента у системі Інтернету речей, званого як шлюз.
3. Проведення апробації шлюзу Інтернету речей у системі та, за умови наявності недоліків, їх усунення при можливості.

|      |      |         |        |      |  |                          |           |
|------|------|---------|--------|------|--|--------------------------|-----------|
|      |      |         |        |      |  | КВРКІ 200226.21.04.83 ПЗ | Арк.<br>3 |
| Зм.. | Арк. | №докум. | Підпис | Дата |  |                          |           |

# 1 ДОСЛІДЖЕННЯ ОБЛАСТІ ІНТЕРНЕТУ РЕЧЕЙ ТА ПРАКТИЧНА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Чим є шлюз Інтернету речей: його опис та функціональність

Інтернет речей представляє собою, кажучи злегка спрощено, певну низку пристроїв, об'єднаних у мережу, де є хмара для обміну даними, збираними датчиками. В загальному контексті комп'ютерних мереж, поняття «шлюз» означає вузол для зв'язування пристроїв з різними інтерфейсами та протоколами передачі даних; ним може виступати як апаратний маршрутизатор, так і програмне забезпечення. В контексті Інтернету речей, шлюз виконує ту ж саму функцію, об'єднує електронні пристрої (датчики та контролери, що разом формують сенсор), що мають можливість передачі даних за різними протоколами та направляє зібрані дані у хмарне середовище для їх аналізу, моніторингу та схожих дій (рисунок 1.1) [1-2].

### Як шлюз Інтернет речей працює?



Рисунок 1.1 – Ілюстрація шлюзу Інтернету речей у екосистемі [1-2]

В більшості, сучасні шлюзи Інтернету речей обладнані такими технічними характеристиками [11-15]:

1. Однокристална система.

|      |      |         |        |      |  |                          |           |
|------|------|---------|--------|------|--|--------------------------|-----------|
|      |      |         |        |      |  | КВРКІ 200226.21.04.83 ПЗ | Арк.<br>4 |
| Зм.. | Арк. | №докум. | Підпис | Дата |  |                          |           |

2. Оперативна та постійна пам'яті.
3. Інтерфейси для підключення пристроїв та інтерфейси для підключення до мережі (як безпроводних, так і, можливо, провідних).
4. Операційна система.
5. Програмне забезпечення для роботи шлюзу з пристроями та хмарним середовищем.
6. Безпекові модулі (наприклад, чіп TPM).
7. Належне основне й додаткове джерело живлення.

На фото нижче наведений приклад шлюзу Vivocom TG451 [16]. Його технічні характеристики:

1. Стільникові інтерфейси:
  - a. 2 слоти DSSM SIM.
  - b. Стандарт 4G LTE CAT 6.
  - c. EMEA&APAC: B1/B3/B5/B7/B8/B20/B28.
  - d. NA: B2/B4/B5/B7/B12/B13/B25.
  - e. 2 SMA роз'єми по 50 Ом типу мама.
  - f. Захист від електростатичного розряду 15 кВ.
2. Ethernet:
  - a. 6 портів RJ45 (1-WAN, 5-LAN).
  - b. Швидкість передачі 10/100 Мбіт/с (з автовизначенням).
  - c. Магнітна ізоляція 1,5 кВ.
3. Серійний порт:
  - a. 6-контактна клемна колодка 3,5 мм із замком.
  - b. 2 порти RS232 та 2 порти RS485.
  - c. Швидкість передачі від 300 до 230400 біт/с.
  - d. Захист від електростатичного розряду 8 кВ для RS232 та 15 кВ для RS485
4. Wi-Fi:
  - a. 1 SMA роз'єм 50 Ом типу мама.

|     |      |         |        |      |  |                          |           |
|-----|------|---------|--------|------|--|--------------------------|-----------|
|     |      |         |        |      |  | КВРКІ 200226.21.04.83 ПЗ | Арк.<br>5 |
| Зм. | Арк. | №докум. | Підпис | Дата |  |                          |           |

- b. Стандарт IEEE 802.11b/g/n, підтримка режимів точки доступу та клієнта
  - c. Швидкість передачі IEEE802.11b/g до 54 Мбіт/с, IEEE802.11n до 300 Мбіт/с.
  - d. Стандарт безпеки WPA, WPA2, WPA/WPA2 Enterprise, Radius.
  - e. Потужність Tx 18 дБм (11g), 21 дБм (11b).
  - f. Чутливість Rx <-72дБм@54Мбіт/с.
5. I/O:
- a. 6-контактна клемна колодка 3,5 мм із замком.
  - b. 2 цифрових входи (від 0 до 30 В).
  - c. 2 реле (до 5 А та напругою до 30 В постійного чи 250 В перемінного току).
6. Зовнішня пам'ять:
- a. 1 порт USB 2.0.
  - b. 1 слот microSD до 32 Гб.
  - c. Можливість зберігання даних й оновлення програмного забезпечення.
7. GPS:
- a. Вбудований модуль GPS.
  - b. 1 SMA роз'єм 50 Ом типу мама.
8. Живлення та споживання:
- a. 2-контактний роз'єм з клемною колодкою 3,5 мм.
  - b. Стандартне живлення 12 В та 1,5 А постійного току; номінальна напруга від 5 до 35 В.
  - c. Споживання 280-330mA@12 В постійного току.
  - d. Споживання в режимі очікування 190-240mA@112 В постійного току.
9. Програмні протоколи:

|      |      |         |        |      |                          |           |
|------|------|---------|--------|------|--------------------------|-----------|
|      |      |         |        |      | КВРКІ 200226.21.04.83 ПЗ | Арк.<br>6 |
| Зм.. | Арк. | №докум. | Підпис | Дата |                          |           |

a. Протоколи маршрутизації – PPP, PPPoE, SNMP v1/v2c/v3, TCP, UDP, DHCP, RIPv1/v2, OSPF, DNS, DDNS, HTTP, ARP, QoS, SNMP, Telnet, VLAN, SSH тощо.

b. Протоколи серійного порту – MQTT, Transparent (TCP клієнт/сервер, UDP клієнт/сервер), шлюз Modbus (з Modbus RTU на Modbus TCP).

c. Протоколи VPN – IPsec/PPTP/L2TP/GRE/OpenVPN.

d. Брандмауер – ACL, DMZ, переадресація портів, прив'язка до MAC-адреси.

e. Керування – Web, CLI, SMS, Cloud DMP (платформа керування пристроями).

f. Надійність – відмовостійкість WAN, резервне копіювання на дві SIM-карти, балансування навантаження, контроль апаратного та програмного забезпечення.

g. Розробка на основі Python, LUA та SDK.

h. Підтримка OpenWrt.

10. Фізичні характеристики:

a. Захист від проникнення стандарту IP30.

b. Металевий корпус.

c. Вага 630 г, без аксесуарів.

d. Розміри 145 × 114 × 45 мм.

e. Можливість настільного монтажу з допомогою DIN-рейки.

11. Робочі умови:

a. Робоча температура від -35 °C до +75 °C (від -31 °F до +167 °F).

b. Температура зберігання від -40 °C до +80 °C (від -40°F до +176°F).

c. Відносна робоча вологість оточення від 0% до 93% (без конденсації) при 25°C/77°F.

d. Ізоляція Ethernet 1,5 кВ RMS.

12. Інше:

a. Кнопка скидання RESET.

|     |      |         |        |      |                          |           |
|-----|------|---------|--------|------|--------------------------|-----------|
|     |      |         |        |      | КВРКІ 200226.21.04.83 ПЗ | Арк.<br>7 |
| Зм. | Арк. | №докум. | Підпис | Дата |                          |           |



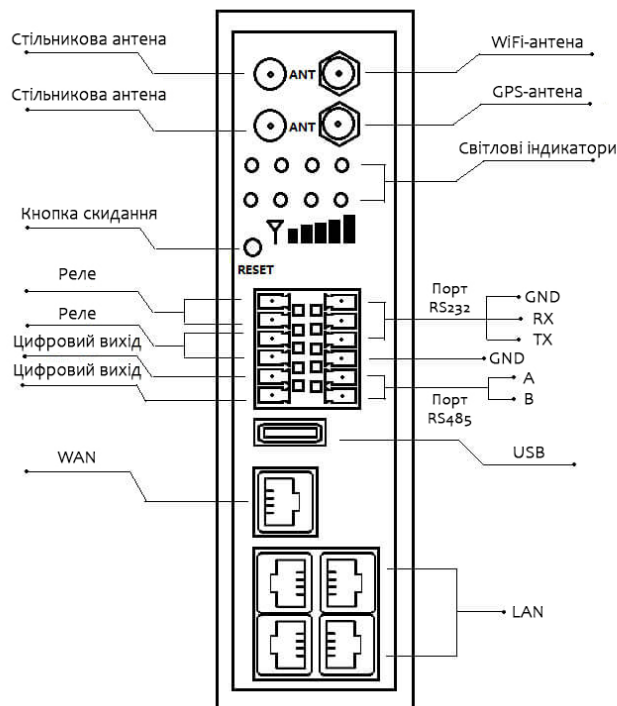


Рисунок 1.4 – Фізична специфікація шлюзу Інтернету речей Vivosom TG451. Вид – ззаду [16]

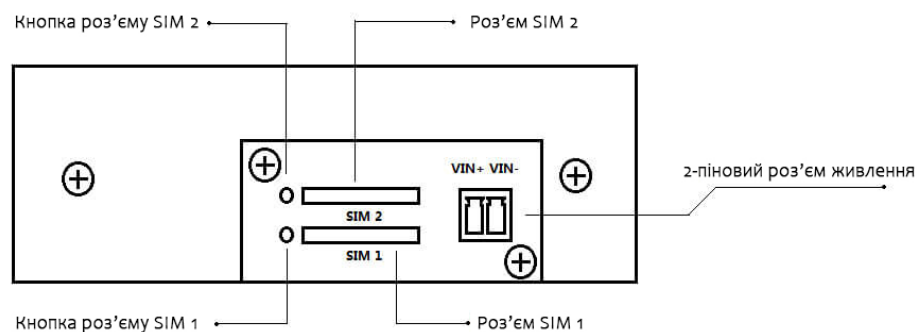


Рисунок 1.5 – Фізична специфікація шлюзу Інтернету речей Vivosom TG451. Вид – зверху [16]

Основними функціями шлюзу Інтернету речей є [1-2, 10-15]:

1. Спілкування з хмарним середовищем. Це – одна із фундаментальних функцій. Шлюз зв'язує кінцеві пристрої – Endpoints – та хмарне середовище, – Cloud – де необроблені дані, що зібрані із девайсів (сенсорів, камер, побутових приладів та схоже), пройшовши рівень туманних обчислень, можуть передаватись до хмарного середовища для їх кінцевої обробки й передаватись до клієнта для демонстрації.

2. Підтримка декількох протоколів передачі. Так як девайси (зокрема, давачі й сенсори), в силу своїх технічних особливостей, в більшості своїй використовують короткодальні та малопотужні протоколи для забезпечення апаратної компактності та як можна більш низького рівня енергоспоживання (наприклад, короткодальні та малопотужні Bluetooth Low Energy [25-26], ZigBee [29], Radio Frequency Identification (RFID) [33], LoRaWAN [30] або середньодальні та більш потужні Wi-Fi [21], DASH7 [22], LoRa [29] та NB-Fi [34]) [4, 6], шлюз тут виконує роль медіатора для збору даних через ці самі протоколи та їх передачі через більш енергоспоживчі протоколи, як LTE або Ethernet [31, 35].

3. Маршрутизація трафіку. У шлюзу Інтернету речей принцип роботи такий ж, як у звичайного мережевого шлюзу: він служить для маршрутизації трафіку, являючись однією, центральною точкою для цієї цілі, через що потоком даних (трафіком) управляти легше.

4. Брандмауер для сенсорних вузлів. Так як шлюз являється центральною точкою у екосистемі Інтернету речей з доступом до інтернету, трафік, зібраний із девайсів, можна фільтрувати з допомогою прописування відповідних правил, що легше, ніж, наприклад, регулювання доступу до інтернету для кожного девайсу окремо.

5. Безпека даних. Зв'язано із попереднім пунктом. Так як монтування різноманітним апаратних та/або програмних безпекових модулів є, як мінімум, з точки зору ціноформування та складності не доцільним, роль безпекового модуля виконує шлюз: він практично завжди поставляється з основними механізмами безпеки, через що є можливість «розгулятись» в плані дбання про безпеку даних з допомогою як і фільтрації трафіку, про що було сказано вище, так і, наприклад, розширеним шифруванням даних з допомогою криптографічного процесора TPM 2.0, протидії шкідливого програмного забезпечення з допомогою антивірусів та схоже.

6. Кешування. В силу того, що доступ до інтернету та, як наслідок, хмарного середовища є не завжди, шлюз має можливість притримати дані у локальному сховищі на протязі життя пакетів (TTL), що забезпечує їх безпеку;



Прикладами програмного забезпечення для роботи з шлюзом Інтернету речей можуть виступити:

1. Eclipse Kura [36]. Це кросплатформовий розширюваний фреймворк з відкритим кодом на основі Java/OSGi. Kura пропонує API-доступ до апаратних інтерфейсів шлюзів Інтернету речей (серійні інтерфейси, GPS, GPIO, I2C та схоже). Він містить готові до використання польові протоколи (включаючи Modbus, OPC-UA, S7), контейнер додатків і веб-програмування візуального потоку даних для отримання даних із сенсорів, їх обробки шлюзом та публікації в провідні хмарні платформи Інтернету речей через підключення MQTT.

2. Ubiworx [37]. Програмне забезпечення для ОС Linux. Воно працює на вбудованих системах для створення шлюзів Інтернету речей, які з'єднують давачі та виконавчі механізми з системами зберігання даних і звітності в хмарі, щоб сформувати повні наскрізні рішення з підтримкою Інтернету речей.

3. OpenHAB [38]. Це – одна з найбільших платформ домашньої автоматизації з відкритим кодом, яка підтримує тисячі пристроїв і може похвалитися спільнотою з понад 43 000 учасників. Підтримує масу стандартів підключення, включаючи напопулярніші, і дозволяє керувати пристроями за допомогою голосових команд, сценаріїв, тригерів і правил на основі подій.

4. Home Assistant [39]. Якщо потрібне щось з більш інтуїтивно ясным інтерфейсом – можна спробувати ОС Home Assistant. Як і OpenHAB, він підтримує величезну кількість пристроїв і операційних систем, дозволяючи комбінувати гаджети розумного дому, як заманеться.

5. Domoticz [40]. Ще одне досить хороше ПЗ для домашньої автоматизації з відкритим кодом, хоча трохи обмеженіша, ніж Home Assistant і OpenHAB. Проте не потребує деінсталяції основної системи, а встановлюється поверх неї, що для людей, не бажаючих деінсталювати основну систему, може бути єдиним варіантом.

6. Contiki-NG [41]. Кросплатформова операційна система з відкритим вихідним кодом для пристроїв Інтернету речей наступного покоління. Фокусується на безпечній та надійній комунікації пристроїв малої потужності та стандартних протоколах, таких як IPv6/6LoWPAN, 6TiSCH, RPL та CoAP. Contiki-NG





Рисунок 1.7 – Шлюз Advantech UTX-3117 [17]

3. Laird Connectivity, Inc. Популярний виробник девайсів Інтернету речей, до яких шлюзи входять теж. Проте варто бути уважним: шлюзи не універсальні в плані протоколів та інтерфейсів, проте, в більшості, зі захистом від води [23]. Приклад шлюзу Sentrius RG186 наведено на рисунку 1.8.



Рисунок 1.8 – Шлюз Laird Sentrius RG186 [23]

4. AAEON Technology, Inc. Виробляє шлюзи Інтернету речей для промислових проектів [18]. Приклад шлюзу SRG-IMX8P наведено на рисунку 1.9.



Рисунок 1.9 – Шлюз AAEON SRG-IMX8P [18]

Так як у нас розробка шлюзу Інтернету речей на базі Raspberry Pi, ми можемо проаналізувати апаратні компоненти шлюзів вищенаведених або ще яких-небудь спеціалізованих компаній, вивести з них середнє арифметичне моделі нашого шлюзу й сформувавши список необхідних йому апаратних компонентів, для його успішної роботи.

#### 1.4 Про Raspberry Pi

Raspberry Pi – мультифункціональний одноплатний ПАОП, часто використовуваний для роботи та досліджень в області інформаційних технологій; випускається компанією Raspberry Pi Foundation. Говорячи про Raspberry Pi серії Pi 1-5, не дивлячись на розмір площі плати приблизно з кредитну картку, продукція цієї серії являються найпопулярнішою у компанії. Згідно статистики Леса Паундера, викладача курсу «Picademy» у Raspberry Pi Foundation, на сайті tomshardware.com ним у статті від 29.02.2025 було сказано, що за весь час існування компанії загальна кількість продажів Raspberry Pi становить 61 мільйон, із яких 21 мільйон припадає на одноплатні комп'ютери Raspberry Pi 3B/3A+/3B+ (рисунок 1.10) [42]. Не дарма: завдяки апаратним характеристикам, ентузіасти зі всього світу використовують їх для створення найрізноманітніших домашніх проектів – від створення розважальних ігрових консолей та медіа-серверів до створення дослідницьких станцій для машинного навчання або програмування, потенціал проектів – колосальний.



Рисунок 1.10 – Raspberry Pi 3A+, 3B та 3B+ (зліва направо) [50]

Кількість продукції компанії – теж не є скудною: крім одноплатних комп’ютерів (й не тільки серії Pi 1-5), на офіційному сайті можна знайти мікроконтролери, сенсори, дисплеї, розширювальні модулі, плати для вбудованих систем, корпуси, кабелі й масу інших потрібних для цих самих проектів аксесуарів, не говорячи про те, що сумісна з продукцією компанії Raspberry Pi Foundation випускається й іншими компаніями. Приклад – компанія Waveshare: її продукція (рисунки 1.11-1.12), крім досить точної сумісності, містить і ясну технічну документацію для більшої своєї частини [43].



Рисунок 1.11 – Продукція компанії Waveshare [43]

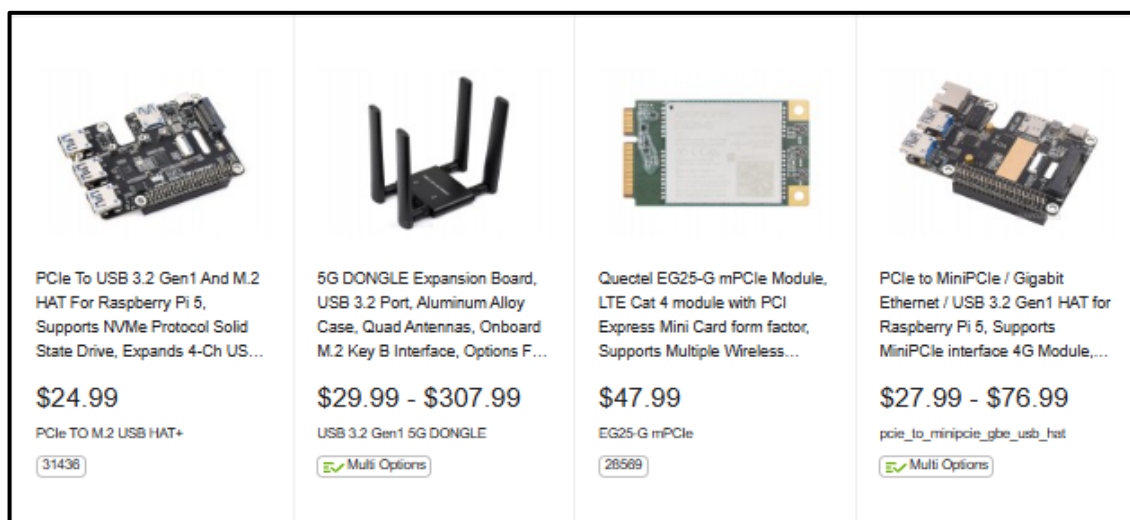


Рисунок 1.12 – розширювальні плати для проектів IoT на базі Raspberry Pi [43]

## 1.5 Аналіз й опис обраної платформи Raspberry Pi

Для створення шлюзу Інтернету речей обрано варіант Raspberry Pi 4B (рисунки 1.12 та 1.13).



Рисунок 1.12 – Зовнішній вигляд Raspberry Pi 4B. Вид – зверху [44]

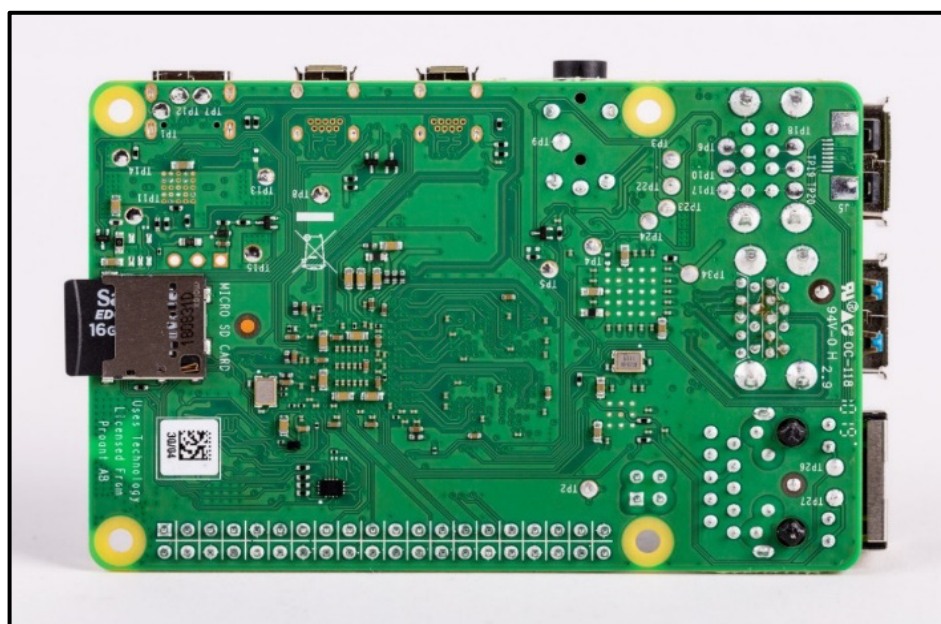


Рисунок 1.13 – Зовнішній вигляд Raspberry Pi 4B. Вид – знизу [44]

|      |      |         |        |      |
|------|------|---------|--------|------|
|      |      |         |        |      |
| Зм.. | Арк. | №докум. | Підпис | Дата |

Дана модель обладнана достатньо потужною та мультифункціональною специфікацією й досить точно підходить у якості платформи для розробки шлюзу. У ній є [44]:

1. Апаратне обладнання:

а. Однокристална система Broadcom BCM2711 з:

- 1) центральним процесором ARM Cortex A-72;
- 2) апаратним декодером для кодеків H.264 та H.265;
- 3) мультимедійним процесором VideoCore VI 3D Graphics;

б. 1 Гб LPDDR4-2400 SDRAM (існують варіанти моделі з 2-ма, 4-ма та 8-ма Гб на борту).

2. Інтерфейси:

а. Безпроводні:

- 1) WLAN стандарту 802.11b/g/n/ac;
- 2) Bluetooth 5.0 з підтримкою BLE;

б. Провідні:

1) 1 порт microSD (з підтримкою SDHC, SDXC);  
2) 2 порти Micro HDMI (з підтримкою моніторів із розширенням до 4K);

3) 2 порти USB 2.0 та 2 порти USB 3.0;

4) 1 порт 10/100/1000 Ethernet (з підтримкою PoE з додатковим PoE NAT);

5) 1 порт MIPI CSI (для підключення камери) та 1 порт MIPI DSI (для підключення дисплею);

6) 1 порт 3.5 мм mini Jack (стандарту TRRS – з підтримкою аудіо у форматі стерео та відео у форматі CVBS);

7) 40-контактний GPIO, 28 з яких з підтримкою інтерфейсів:

а) UART (до 6-ти портів);

б) I2C (до 6-ти портів);

в) SPI (до 5-ти портів);

г) PWM (до 2-ох каналів);

|     |      |         |        |      |                          |            |
|-----|------|---------|--------|------|--------------------------|------------|
|     |      |         |        |      | КВРКІ 200226.21.04.83 ПЗ | Арк.<br>18 |
| Зм. | Арк. | №докум. | Підпис | Дата |                          |            |

- e) GPCLK (до 3-ох виходів);
- f) SDIO (1 порт);
- g) DPI (1 порт);
- h) PCM (1 порт);

3. Програмне обладнання:
  - a. сет інструкцій для ARMv8.
  - b. наявність функцій GPU за допомогою стандартних API.
  - c. стек ПЗ для роботи ядра Linux.
  - d. API для роботи з графікою – OpenGL ES 3.1 та Vulkan 1.0.

Живлення Raspberry Pi 4B може відбуватись трьома методами:

1. Через порт USB-C (5V, 3A постійного току мінімум).
2. Через відповідний порт GPIO (5V, 3A постійного току мінімум).
3. Через порт PoE (за умови наявності плати розширення PoE HAT).

Вибір – варіант 1: це найнадійніший варіант живлення, т.я., крім самої плати, живитимуться й додатково під'єднані до плати комплектуючі. Нормальна температура для роботи з Raspberry Pi – 0-50 °C. У проекті для забезпечення благонадійності його роботи необхідно передбачити:

1. Надійний блок живлення.
2. Надійний носій пам'яті, який працюватиме через порт microSD або USB (виступатиме у якості ПЗУ, тому для забезпечення цілісності інформації скупитись не варто; якщо носій microSD – його важливо обрати класу, в ідеалі, 10-го).
3. Корпус, перекриваючий доступ до елементів плати.
4. Охолодження для плати (як мінімум, пасивне; може йти у комплекті з корпусом).

Розмір плати без аксесуарів – 85 × 56 × 16 мм. Фізична специфікація плати зображена на рисунках 1.14 та 1.15. Більш детальну інформацію про модель даного одноплатного комп'ютера можна знайти у описі продукту та, краще, його технічному паспорті на офіційному сайті [44].

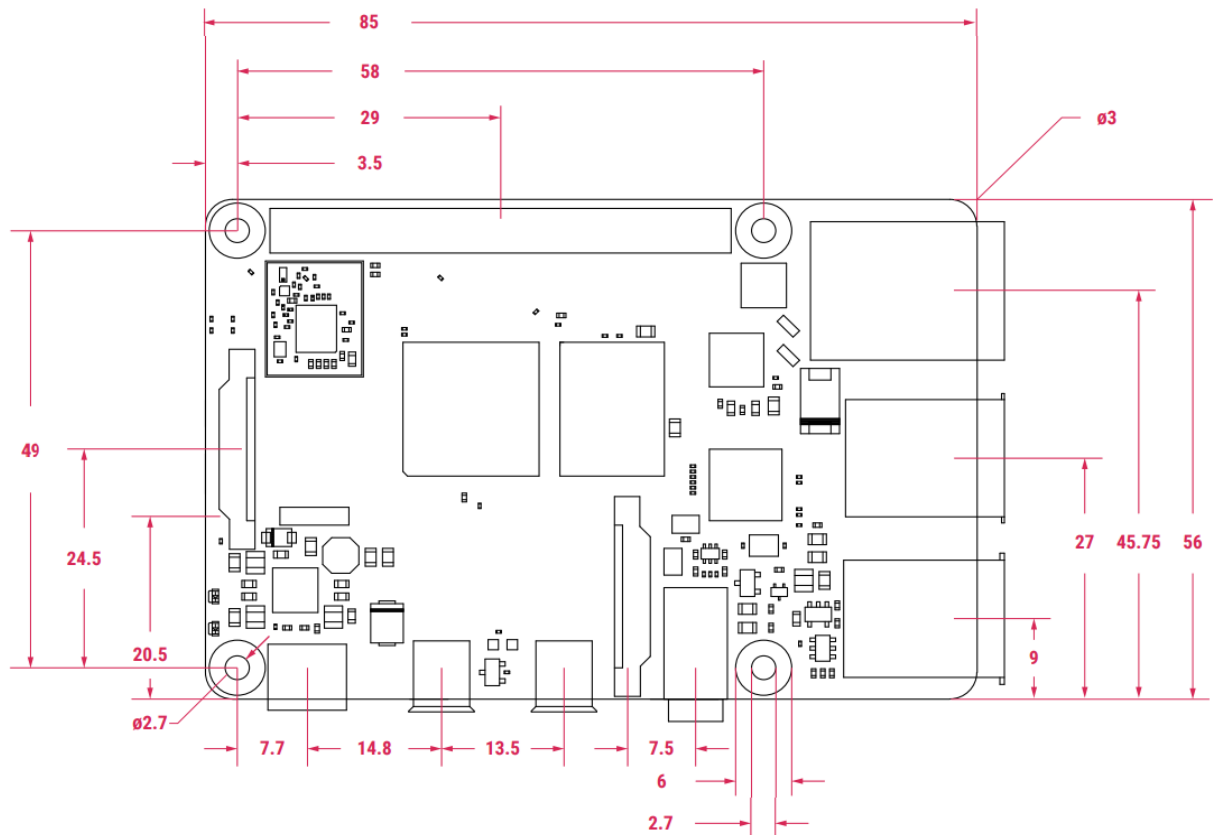


Рисунок 1.15 – Фізична специфікація плати Raspberry Pi 4B. Числові значення – у міліметрах [44]

## 1.6 Висновок до першого розділу

Під час роботи над першим розділом дипломної роботи я більш поглиблено ознайомився з поняттям Інтернет речей, його архітектурою та детально познайомився з роллю в цій самій архітектурі такого елемента, як шлюз Інтернету речей. Отримані знання, базовані на аналізі апаратних та програмних рішень для створення шлюзу Інтернету речей, дали можливість конвертувати їх у моделювання власного проекту шлюзу Інтернету речей з використанням продукції від відомої світової компанії під назвою Raspberry Pi Foundation, а саме – Raspberry Pi, й визначитись над додатковою апаратною й програмною частиною для його проектування та подальшої реалізації.

## 2 ПРОЕКТУВАННЯ ШЛЮЗУ ІНТЕРНЕТУ РЕЧЕЙ НА БАЗІ RASPBERRY PI

### 2.1 Функціональні вимоги шлюзу Інтернету речей

В якості зв'язуючого елементу у нас виступає комп'ютер, до якого, з допомогою відповідних інтерфейсів та протоколів, приєднуються IoT-елементи: сенсори, сервери для обчислення даних, пристрої для оперування даними та схоже. На рисунку 2.1 приведений детальний приклад основних шарів архітектури Інтернету речей. За ним можна зрозуміти, що відбувається на кожному з них [45]:

1. На першому, «сприйнятливому» рівні – давачі, сенсори або інші схожого сенсу пристрої збирають дані про оточуюче середовище. Шлюз, з допомогою відповідних й, в більшості, низькоспоживчих та короткодальніх провідних/безпроводних інтерфейсів та протоколів, отримує дані з кожного девайсу.

2. На другому, «транспортному» рівні – зібрані дані з допомогою шлюзу, на якому наявні та налаштовані відповідні інтерфейси та протоколи, передаються у відповідне апаратне середовище для їх подальшої обробки. Шлюз виступає як маршрутизатор, до того ж зв'язуючий пакети даних з різними протоколами їх передачі та з допомогою технологій (стандартів, інтерфейсів, протоколів) передачі даних на далекі відстані передають їх до відповідного серверного середовища.

3. На третьому, «обробному» рівні – отримані дані, відповідно, самим середовищем оброблюються: частіше за все, агрегуються, аналізуються, сортуються, передаються до баз даних й, можливо, на їх основі проводиться машинне навчання та/або ще які-небудь залежні від специфіки конкретної задачі дії. На даному рівні, шлюз участі практично не бере, може лиш виконати невелику первинну обробку даних в силу своїх апаратних можливостей, які, частіше за все, не можуть порівнюватись із потужністю масштабних серверів.

4. На четвертому, «користувацькому» рівні – вже оброблені й отримані з серверів дані демонструються у програмному середовищі користувачу. Шлюз

Інтернету речей ролі не грає абсолютно ніякої, все цілком залежить від програмно-апаратного забезпечення сервера та клієнта.

5. На п'ятому, «бізнес» рівні – на основі отриманих даних, користувач приймає рішення – наприклад, що додати до архітектури, що відняти, які дані надсилати, які – фільтрувати, та схожі необхідні для себе дії. Тут шлюз ролі не грає теж, скоріше користувач задає правила для його ролі у архітектурі.

## П'ЯТИРІВНЕВА АРХІТЕКТУРА ІНТЕРНЕТ РЕЧЕЙ



Рисунок 2.1 – Шари архітектури Інтернету речей [45]

## 2.2 Основні елементи шлюзу Інтернету речей

В залежності від специфіки та масштабу архітектури, шари можуть мінятись, додаватись або деконструюватись на більш тонкі, для поставленої задачі вищеприведена версія моделі архітектури Інтернету речей підходить. Загальна схема апаратного забезпечення шлюзу буде зображена у додатку А, основні шари та роль шлюзу у системі визначено. Відповідно, визначаємо подальший план дій по його створенню, а саме визначаємо основні нашого шлюзу.

Для першого рівня необхідно визначитись, з якими девайсами шлюз працюватиме. У нашому випадку, це чотири девайси:

1. Давач BMP280 (рисунок 2.2). Комбінований давач тиску, вологості та температури; додає у архітектуру можливість вимірювати тиск, рівень вологості й температури у навколишньому середовищі; вимірюватиме – відповідно, у Паскалях (Pa), відсотках (%), та градусах Цельсія (°C). Технічні характеристики [46]:

- a. Діапазон вимірювання тиску від 300 до 1100 гПа (еквівалентно +9000...-500 м над/під рівнем моря).
- b. Діапазон вимірювання температури від -40 до +85 °C.
- c. Корпус 8-выводний металевий LGA з кришкою.
- d. Розмір: площа  $2,0 \times 2,5$  мм<sup>2</sup>, висота 0,95 мм.
- e. Вага – 1,2 гр.
- f. Відносна точність ~ 0,12 гПа, що еквівалентно ±1 м.
- g. Абсолютна точність ~ 1 гПа (при температурі від 0 до +40 °C).
- h. Коефіцієнт температурного зміщення 1,5 гПа/К, що еквівалентно 12,6 см/К (при температурному діапазоні від +25 до +40 °C та тиску навколишнього середовища 900 гПа).
- i. Цифровий інтерфейс I<sup>2</sup>C (до 3,4 МГц);
- j. Споживання струму 2,7 мкА при частоті вибірки 1 Гц.
- k. Відповідає стандарту RoHS, не містить галогенів.
- l. Рівень вологості відповідає рейтингу MSL 1.

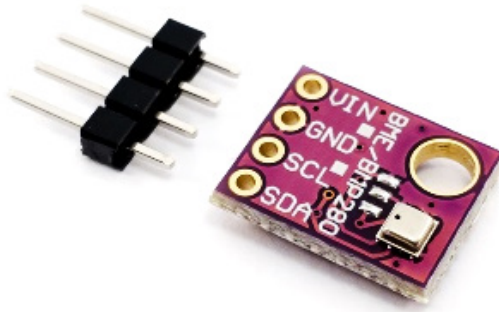


Рисунок 2.2 – Давач VMP280 [46]

2. Давач TSL2561 (рисунок 2.3). Давач освітленості; додає у архітектуру можливість вимірювати у навколишньому середовищі інтенсивність освітлення; вимірюватиме – у люксах (лк). Технічні характеристики [47]:

- a. Діапазон вимірювань освітленості – 0.1 лк до 40,000 лк.
- b. Динамічний діапазон – 0.1 лк до 88,000 лк.
- c. Діапазон довжини хвилі – 300 нм до 1000 нм.
- d. Точність –  $\sim 0.1$  лк.
- e. Інтерфейс – I2C.
- f. Живлення – напруга від 3.3 до 5 В та сила току до 0.6 мА.
- g. 2 вбудованих інтегруючих модуля АЦП.
- h. Розмір – 4 мм  $\times$  4 мм  $\times$  0.9 мм (SMD); площа плати – 17 мм  $\times$  13 мм.
- i. Робоча температура – від -40 °С до +85 °С.
- j. Вбудований фільтр для блокування інфрачервоного світла.

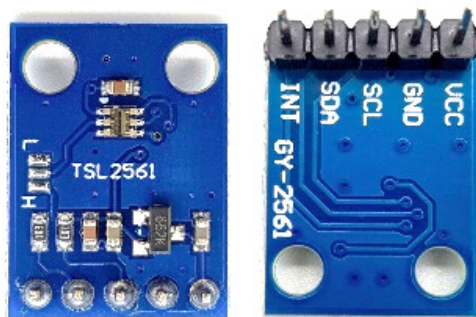


Рисунок 2.3 – Давач TSL2561 [47]

3. Давач MQ-135 (рисунок 2.4). Комбінований давач якості повітря; додає у архітектуру можливість вимірювати у навколишньому середовищі рівень аміаку, бензену, спирту, чадного та вуглекислого газів й схожих домішок у повітрі; вимірюватиме – у відсотках (%), частинках на мільйон (ppm) та опорі навантаження (LR) (одиниця вимірювання залежить від речовини). Технічні характеристики [48]:

a. Діапазон вимірювань – залежно від газу, включає гази, як аміак (NH<sub>3</sub>), оксид вуглецю (CO), бензол (C<sub>6</sub>H<sub>6</sub>), дим, летючі органічні сполуки (VOCs) та інші.

b. Чутливість – варіюється в залежності від газу, для якого вимірюється, і встановленої концентрації.

c. Інформаційний вихід – цифровий (DO) та аналоговий (AO), який змінюється залежно від концентрації газу.

d. Робочий струм – близько 150 мА.

e. Робоча напруга – 5 В.

f. Робоча температура – від -10°C до +50 °C.

g. Розміри – 35 мм × 20 мм × 21 мм.

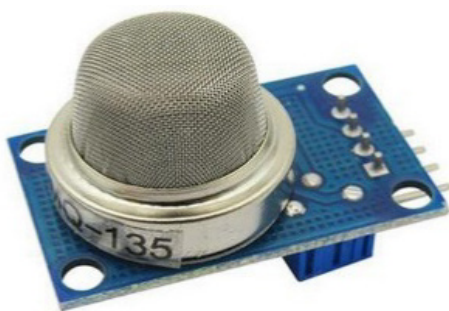


Рисунок 2.4 – Давач MQ-135 [48]

4. Модуль SIM800C GSM/GPRS HAT (рисунок 2.5). Плата розширення, додаючи до шлюзу можливість комунікування з іншими пристроями за технологією мобільного зв'язку GSM на частотах 850, 900 (E-GSM), 1800 (DCS) та 1900 (PCS) МГц (що дає можливість, наприклад, обміну SMS та/або дуплексної телефонії) та технологією пакетної передачі даних GPRS (що дає можливість,

наприклад, обміну даними з пристроями у мережі GSM та мережі Інтернет (до 85 кб/с)). Технічні характеристики [49]:

- a. Інтерфейс – UART.
- b. Підтримувані мережі – GSM/GPRS.
- c. Робочі частоти – 850/900/1800/1900 МГц.
- d. Підтримка текстових повідомлень (SMS) та даних GPRS.
- e. Інтерфейс SIM-карти – microSIM.
- f. Підтримка голосового зв'язку.
- g. Підтримка HTTP, FTP, SMTP, POP3 протоколів.
- h. Підтримка TCP/IP і UDP.
- i. Робоча потужність – напруга 5В, сила току до 2А (через GPIO або microUSB).
- j. Вихідна потужність – до 2 Вт (900 МГц), до 1 Вт (1800 МГц).
- k. Робоча температура – від -40 до +85 °С.

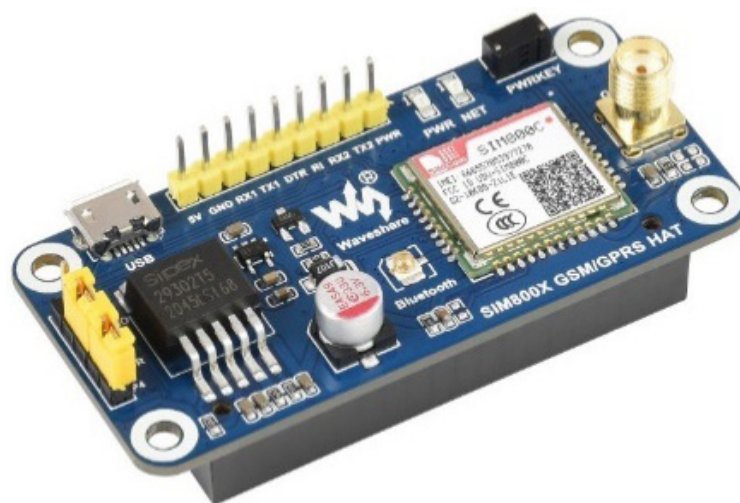


Рисунок 2.5 – Модуль SIM800C GSM/GPRS HAT [49]

Давач MQ-135 з допомогою цифрового виходу (DO) комунікуватиме через інтерфейс GPIO, давачі BMP280 та TSL2561 комунікуватимуть через інтерфейс I2C й модуль SIM800C GSM/GPRS HAT комунікуватиме через інтерфейс UART.

Для другого рівня необхідно, в залежності від девайсів на першому рівні, забезпечити плату Raspberry Pi додатковими адаптерами/конвертерами, які будуть

додавати до неї змогу комунікувати з іншими пристроями з допомогою інших інтерфейсів та протоколів. В даному випадку, інтерфейси, за якими Raspberry Pi може комунікувати з девайсами першого рівня та пристроями маршрутизації даних другого рівня, вже забезпечені виробником плати й тому необхідності додавати які-небудь адаптери в даному випадку немає.

Додатково платі додано можливість працювати на технологіях мобільного стільникового зв'язку GSM та GPRS, що може бути корисним за умови відсутності доступу до інтернету через Wi-Fi та/або Ethernet.

Опціонально, т.я. у давачу MQ-135, крім цифрового, передбачено аналоговий вихід для виведення результату вимірювань у точних значеннях, для Raspberry Pi можна забезпечити АЦП для їх зчитування, наприклад, ADS1015 (рисунок 2.6).

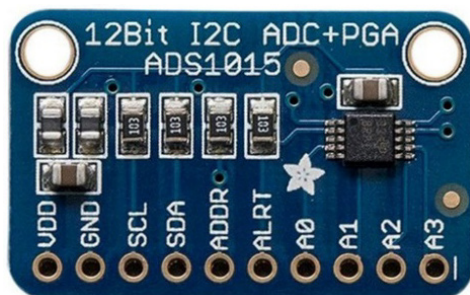


Рисунок 2.6 – АЦП ADS1015

Також варто відмітити, що з безпроводних технологій передачі даних у стандартній специфікації у нас доступні лиш дві – Wi-Fi та Bluetooth – що для шлюзу Інтернету речей є досить скудним набором. Крім вищеназваних, більшість сучасних шлюзів Інтернету речей оснащена додатковими безпроводними технологіями для комунікації з девайсами за іншими такими ж інтерфейсами/протоколами (наприклад, Zigbee, Z-Ware, NFC, RFID та схожі). Якщо у системі були б передбачені девайси, працюючі, наприклад, на технології Zigbee (рисунок 2.8) – для шлюзу необхідно було б забезпечити додатковий адаптер, як, наприклад, Sonoff Zigbee Dongle Plus (рисунок 2.7), працюючий за вже наявним у Raspberry Pi 4B інтерфейсом USB й додаючий до нього здатність комунікації за цією технологією. У нашому випадку шлюз працюватиме з

провідними девайсами та у стаціонарному режимі. Проте, за необхідності його, наприклад, банального перенесення до іншого місця в будинку – через провідні девайси зробити це буде куди складніше, що не скажеш про ситуацію, якби шлюз працював лиш з сучасними безпроводними девайсами за відповідними інтерфейсами.



Рисунок 2.7 – Адаптер Sonoff Zigbee 3.0 USB Dongle Plus



Рисунок 2.8 – Давачі й сенсори, працюючі за технологією Zigbee

Для третього та четвертого рівнів необхідно реалізувати програмну частину шлюзу, котра відповідатиме за програмно-апаратну взаємодію між усіма елементами IoT-системи. В силу того, що апаратна конфігурація вже визначена та обрана, на даному рівні залишається необхідним забезпечити:

1. Вибір операційної системи. З її участю проводитимуться процедури із встановлення програмного забезпечення, конфігурування інтерфейсів, налаштування апаратної частини Raspberry Pi та схожі загальні, забезпечуючі роботоздатність та можливість конфігурування ПЗ.

2. Встановлення програмних інструментів для роботи шлюзу. З його допомогою проводяться мережеві налаштування, встановлення програмного інструментарію (бібліотек, середовищ розробки й схожого) для підтримки інтерфейсів та протоколів девайсів через наявні інтерфейси та протоколи, встановлення системи баз даних для їх зберігання, веб-серверу для створення веб-інтерфейсів для комфортної користувацької взаємодії з даними, забезпечення цілісності даних з допомогою ПЗ для шифрування й схожі важливі дії.

3. Створення спеціалізованого програмного забезпечення. За допомогою середовищ та технологій розробки ПЗ, на даному етапі проводиться, власне, створення спеціалізованого ПЗ, з допомогою якого будуть реалізуватись зв'язки з пристроями, API, обробка даних та подій й схожі конкретні дії, реалізовані з допомогою програмного коду для «зв'язування» кожного елементу у системі.

П'ятий рівень – залежить від користувача, шлюз ролі не грає. Певною мірою він може бути спряжений із четвертим рівнем, на якому досить важливо створити комфортний та інтуїтивно ясний користувацький досвід та програмний інтерфейс.

### 2.3 Висновок до другого розділу

Під час роботи над другим розділом дипломної роботи було спроектовано варіант програмно-апаратної реалізації шлюзу Інтернету речей для його подальшого практичного втілення у системі Інтернету речей. Загальна схема ролі шлюзу Інтернет речей програмній частині архітектури системи буде вказано у додатку Б.

### 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ШЛЮЗУ ІНТЕРНЕТУ РЕЧЕЙ НА ОСНОВІ RASPBERRY PI ТА ЙОГО ТЕСТУВАННЯ

#### 3.1 Встановлення необхідного програмного середовища та конфігурація Raspberry Pi

##### 3.1.1 Вибір, встановлення та налаштування операційної системи

Робота проводитиметься на офіційній операційній системі, що випускається та підтримується компанією – Raspberry Pi OS на базі Debian GNU/Linux. Для встановлення необхідні [50]:

1. Носій пам'яті до 512 Гб (потрібна інформація з якого буде перенесена (!): в ході встановлення ОС вся інформація з нього буде стерта):

а. Якщо формату microSD – додатково потрібний перехідник microSD – SD/USB.

б. Якщо формату USB – додатково не потрібно нічого.

2. ПЗ для запису образів на обраний носій.

3. Образ ОС, підтримуваний обраним ПЗ.

4. Кабель або перехідник з Micro HDMI на Mini HDMI/HDMI для виведення зображення з Raspberry Pi (для належного вибору необхідно ознайомитись з версією порту HDMI на своєму моніторі; формат кабеля – тато-тато/тато-мама, де тато – сторона з Micro HDMI).

Обраний носій – формату microSD. Проводитиметься запис ОС з допомогою офіційного ПЗ компанії для запису образів під назвою Raspberry Pi Imager, його особливість – можливість вибрати необхідну ОС для потрібної платформи, певної версії та конфігурувати її в самому програмному інтерфейсі, що комфортно, проте в ході запису ОС необхідний якісний доступ до інтернету, т.я. вона буде завантажуватись й записуватись одночасно й поступово. Інтерфейс програми – на рисунку 3.1.

|      |      |         |        |      |                          |            |
|------|------|---------|--------|------|--------------------------|------------|
|      |      |         |        |      | КВРКІ 200226.21.04.83 ПЗ | Арк.<br>30 |
| Зм.. | Арк. | №докум. | Підпис | Дата |                          |            |









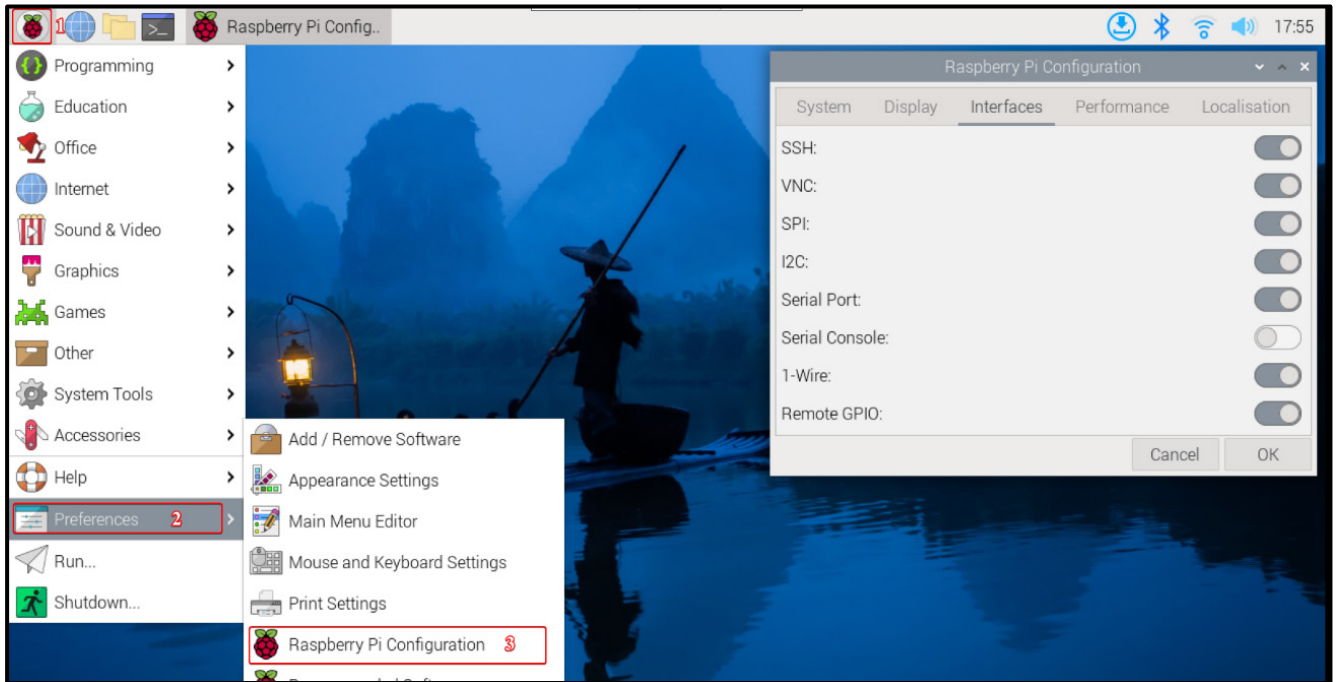


Рисунок 3.7 – Графічний метод налаштування Raspberry Pi OS

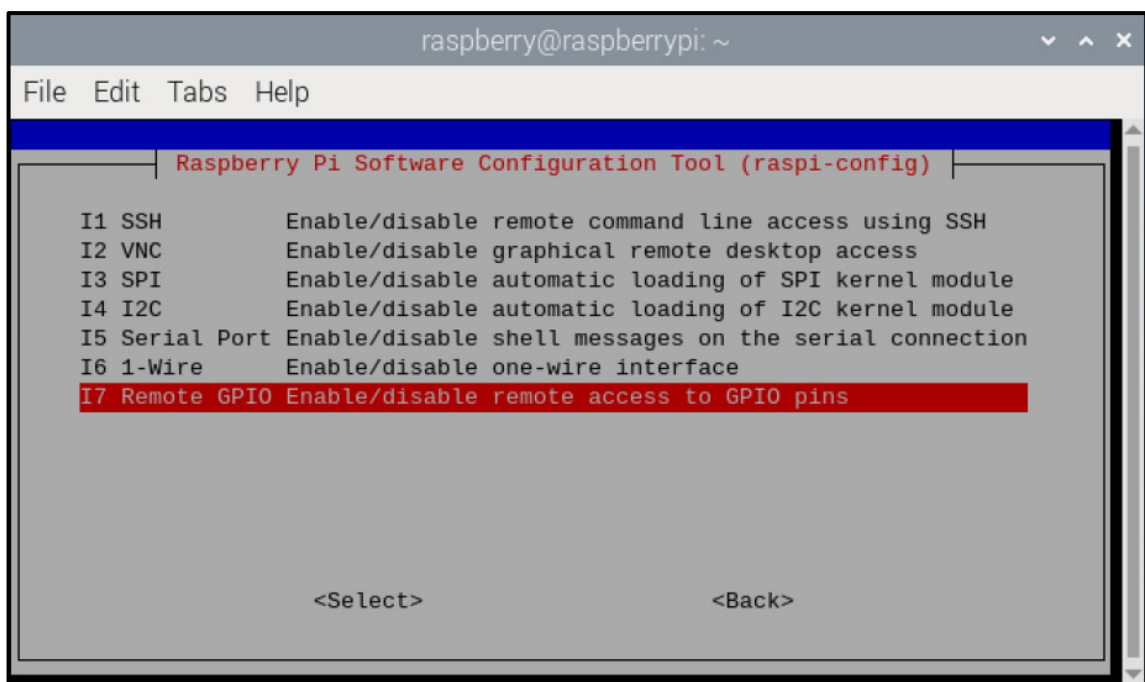


Рисунок 3.8 – Консольний метод налаштування Raspberry Pi OS

Основне що налаштуємо – інтерфейси (рисунок 3.8): вони необхідні для комунікації з іншими пристроями. Спочатку, для віддаленого консольного доступу до Raspberry Pi через SSH-клієнт – ввімкнемо SSH; для локального графічного доступу до Raspberry Pi через LAN/WLAN та VNC-клієнт – ввімкнемо VNC-сервер.

На ПК завантажуюємо VNC-клієнт для потрібної ОС та встановлюємо його. Після – в адресному просторі вводимо раніше знайдену й запам’ятовану IP-адресу/назву хосту нашого Raspberry Pi, тиснемо «Enter» та вводимо дані для авторизації, якщо вони були раніше задані (рисунок 3.9). Після цього ми отримуємо графічний доступ до Raspberry Pi (як на рисунку 3.4) й матимемо змогу комфортно комунікувати з ним через наш основний ПК, без підключення монітору та периферії до Raspberry Pi безпосередньо.

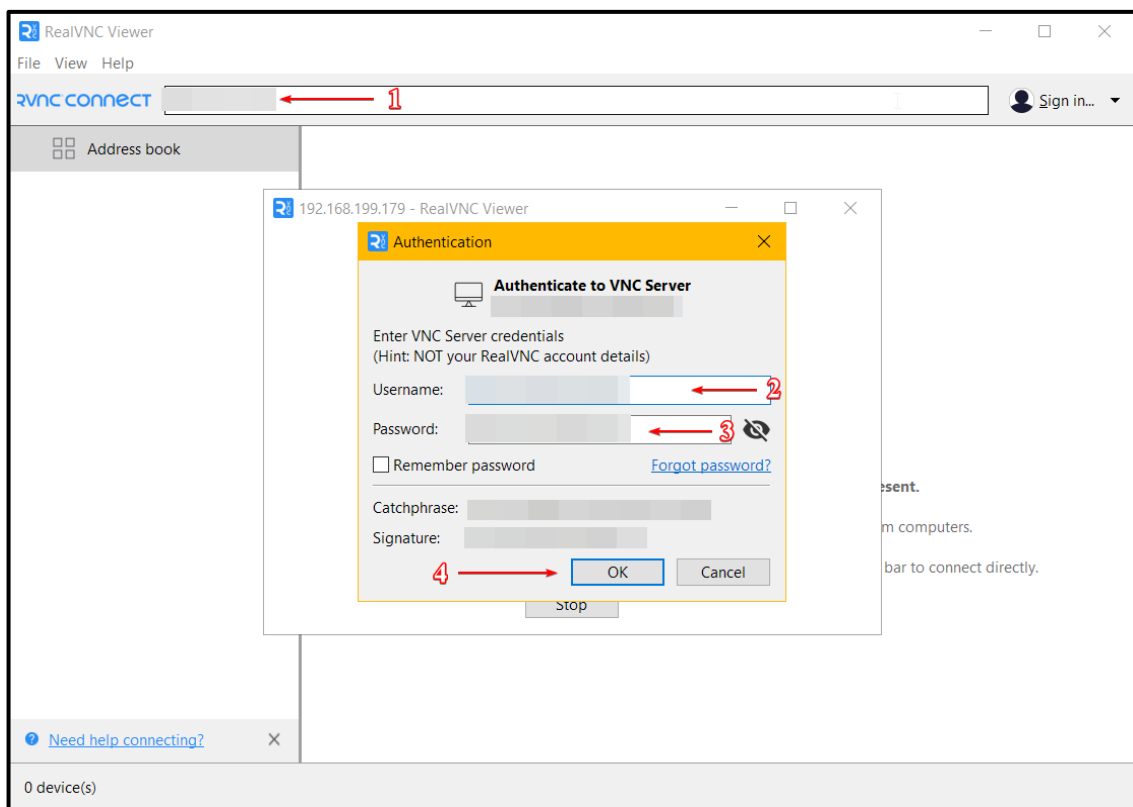


Рисунок 3.9 – Програмний інтерфейс та вікно для авторизації VNC Viewer

Тепер – ввімкнемо підтримку провідних інтерфейсів, які будуть працювати через GPIO та через які ми будемо комунікувати з під’єднуваними девайсами. Так як у системі є девайси, що приєднуюватимуться з допомогою інтерфейсів I2C й UART, їх підтримку необхідно ввімкнути. У тій же вкладці «Інтерфейси» (рисунок 3.10) необхідно ввімкнути підтримку I2C та Serial Port (так іменується порт UART). Опісля – перезавантажити Raspberry Pi.



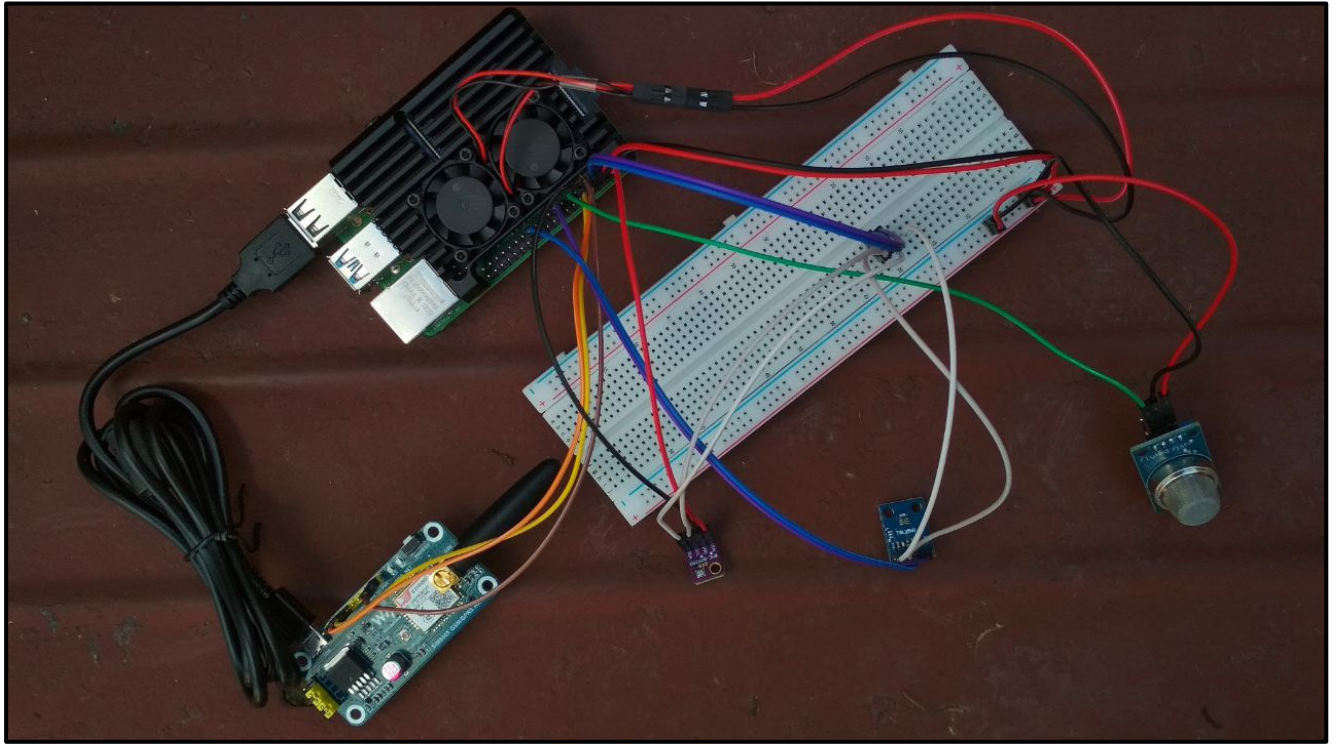


Рисунок 3.11 – Зображення підключення девайсів до плати у вигляді фото

```

raspberrypi@raspberrypi: ~
File Edit Tabs Help
raspberrypi@raspberrypi:~$ i2cdetect -y 1
 00: -- -- -- -- -- -- -- -- -- -- -- -- -- --
 10: -- -- -- -- -- -- -- -- -- -- -- -- -- --
 20: -- -- -- -- -- -- -- -- -- -- -- -- -- --
 30: -- -- -- -- -- -- -- -- 39 -- -- -- -- --
 40: -- -- -- -- -- -- -- -- -- -- -- -- -- --
 50: -- -- -- -- -- -- -- -- -- -- -- -- -- --
 60: -- -- -- -- -- -- -- -- -- -- -- -- -- --
 70: -- -- -- -- -- -- -- -- 76 -- -- -- -- --
raspberrypi@raspberrypi:~$

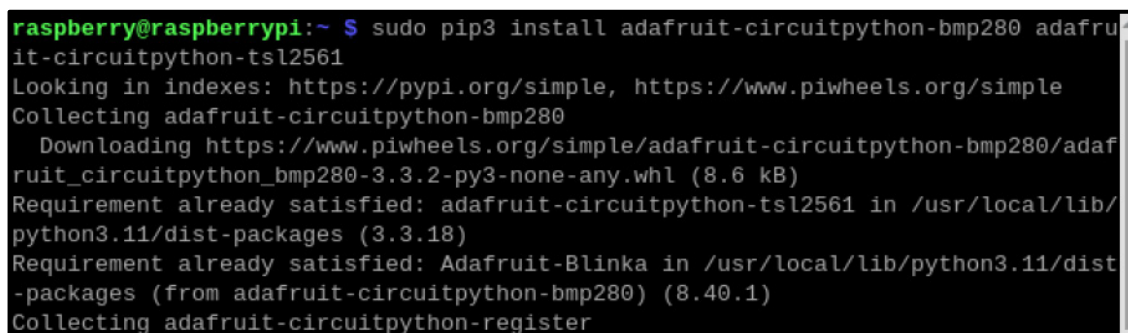
```

Рисунок 3.12 – I2C-адреси кожного девайсу, працюючого за даним протоколом

Перевірку підключення решти девайсів забезпечують світлові індикатори, розташовані на платі самих девайсів. Так як давач моніторингу якості повітря – MQ-135 – працює у цифровому режимі (за допомогою цифрового виходу), сповіщувати користувача про наявність шкідливих для нього домішок у повітрі від може лише термінами на кшталт «порогова концентрація газу/газів досягнута» (1; на платі зелений індикатор горить) й «порогова концентрація газу/газів не досягнута» (0; на платі зелений індикатор не горить), для значного розширення його можливостей в плані, наприклад, демонстрації точних значень присутності конкретної речовини без АЦП не обійтись.

### 3.1.3 Встановлення програмних інструментів для взаємодії шлюзу з девайсами системи Інтернету речей

ПЗ для інтерфейсу I2C на версії Raspberry Pi OS Full забезпечено «з коробки»: для роботи з девайсами на мові програмування Python забезпечено модуль «python-smbus» [52], для роботи з інтерфейсом I2C консольним методом забезпечено утиліту «i2c-tools» [53]. Якщо його немає – встановити дане ПЗ можна командою `sudo apt install python-smbus && sudo apt install i2c-tools`. Для давачів BMP280 та TSL2561, встановимо відповідні модулі для комунікації з ними на мові програмування CircuitPython [54] – схожій похідній від мови Python від компанії Adafruit. Зробимо це посередництвом пакетного менеджера pip3, з допомогою команди `sudo pip3 install adafruit-circuitpython-bmp280 adafruit-circuitpython-tsl2561`.



```
raspberrypi@raspberrypi:~$ sudo pip3 install adafruit-circuitpython-bmp280 adafruit-circuitpython-tsl2561
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting adafruit-circuitpython-bmp280
  Downloading https://www.piwheels.org/simple/adafruit-circuitpython-bmp280/adafruit_circuitpython_bmp280-3.3.2-py3-none-any.whl (8.6 kB)
Requirement already satisfied: adafruit-circuitpython-tsl2561 in /usr/local/lib/python3.11/dist-packages (3.3.18)
Requirement already satisfied: Adafruit-Blinka in /usr/local/lib/python3.11/dist-packages (from adafruit-circuitpython-bmp280) (8.40.1)
Collecting adafruit-circuitpython-register
```

Рисунок 3.13 – встановлення модулів adafruit-circuitpython-bmp280 та adafruit-circuitpython-tsl2561



```
Welcome to minicom 2.8

OPTIONS: I18n
Port /dev/ttyS0

Press CTRL-A Z for help on special keys

AT
OK
AT+CSCS?
+CSCS: "IRA"

OK
```

Рисунок 3.15 – Консоль minicom

Створюватиметься шлюз Інтернету речей на базі Domoticz – програмного забезпечення для домашньої автоматизації, що певною мірою є комбінацією готового серверу та програмного веб-інтерфейсу [40]. Для початку встановимо його: у цьому допоможе команда – `sudo bash -c "$(curl -sSfL https://install.domoticz.com)"` (рисунки 3.16 та 3.17).

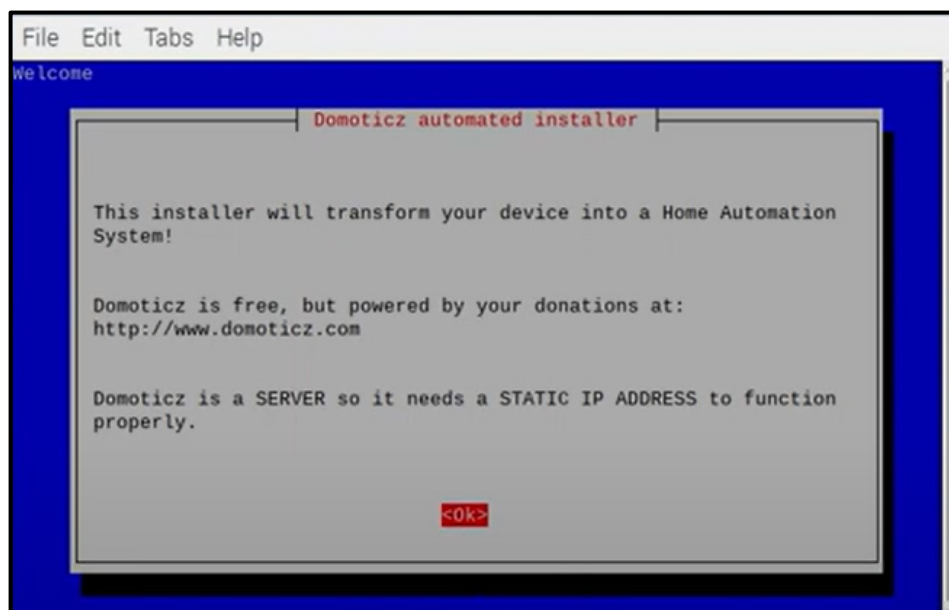


Рисунок 3.16 – Інсталятор Domoticz

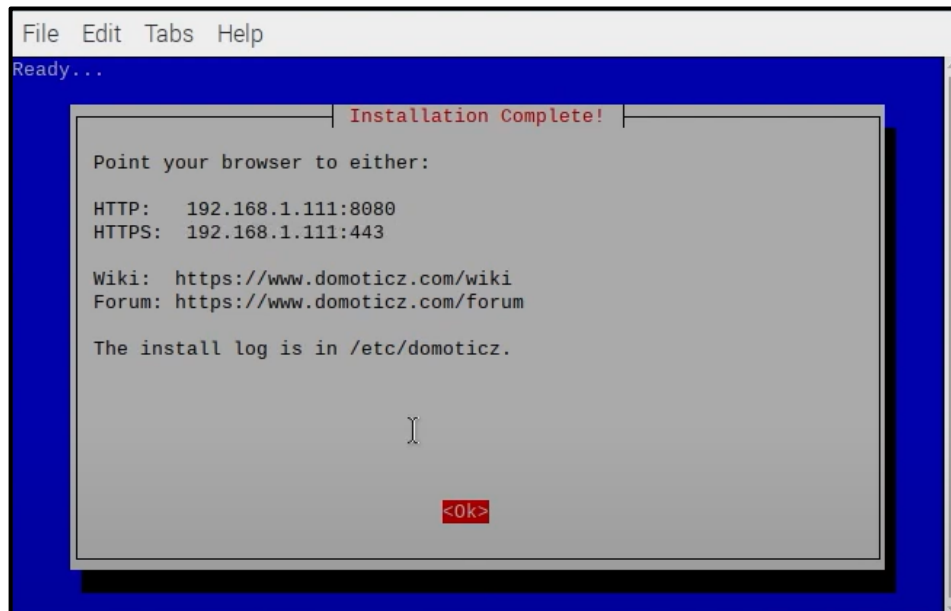


Рисунок 3.17 – Зображення результату встановлення Domoticz

Як видно, після встановлення виводиться приклад підключення до сервера, веб-сайт ПЗ з його форумом та директорія, куди Domoticz було встановлено. Згадавши IP-адресу Raspberry Pi у LAN, під'єднуємось за портом. Після цього нас зустріне форма авторизації (рисунок 3.18). За замовчуванням: логін – admin, пароль – domoticz.

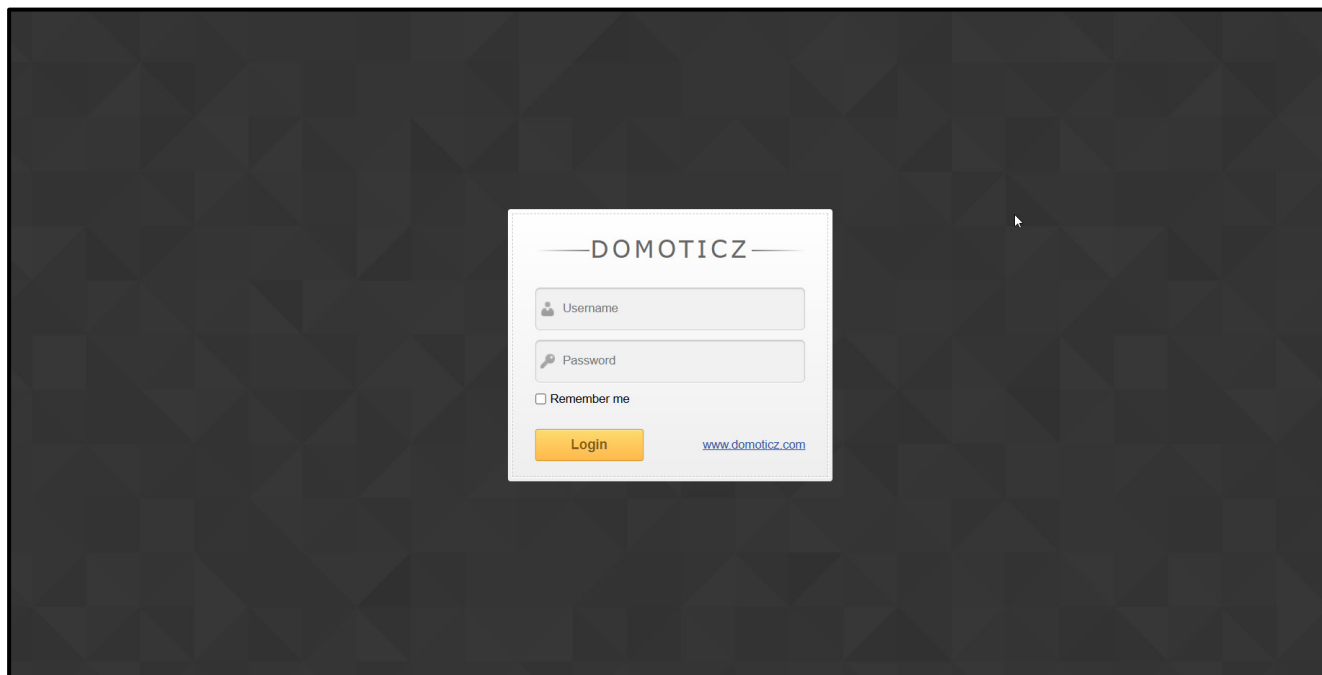


Рисунок 3.18 – Вікно авторизації Domoticz







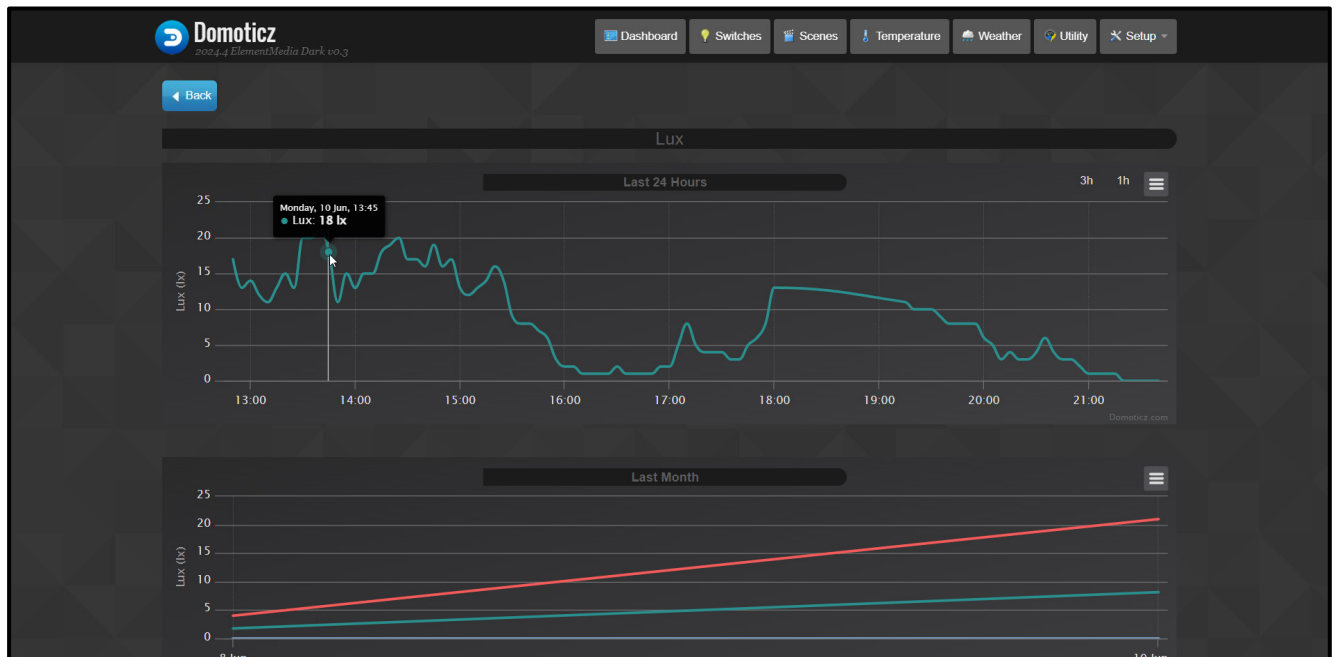


Рисунок 3.24 – Графік освітлення Давача TSL2561

## 3.2 Реалізація програмного забезпечення для роботи шлюзу

### 3.2.1 Створення та тестування програмного забезпечення для комунікації з девайсами

Напишемо програмний код для роботи кожного з датчиків. Результати для кожного – на відповідних рисунках: 3.25, 3.26 та 3.27.

```

bash
File Edit Tabs Help
Temperature - 24.3 degrees of Celsius
Pressure - 980.5 hPa
Altitude - 0.0 meters

Temperature - 24.3 degrees of Celsius
Pressure - 980.5 hPa
Altitude - -0.2 meters

Temperature - 24.3 degrees of Celsius
Pressure - 980.5 hPa
Altitude - -0.4 meters

Temperature - 24.3 degrees of Celsius
Pressure - 980.5 hPa
Altitude - -0.5 meters
  
```

Рисунок 3.25 – Результат роботи коду для BMP280

```
raspberrypi@raspberrypi: ~/SCRIPTS/TSL2561
File Edit Tabs Help
Chip ID = (5, 0)
Enabled = True
Gain = 0
Integration time = 1
Configuring TSL2561...
Getting readings...
Enabled = True
Gain = 0
Integration time = 1
Broadband = 200
Infrared = 89
Lux = 132.61
raspberrypi@raspberrypi:~/SCRIPTS/TSL2561 $
```

Рисунок 3.26 – Результат роботи коду для TSL2561

```
bash
File Edit Tabs Help
Gas is not detected
Gas is not detected
Gas is not detected
Gas is detected!
Gas is detected!
Gas is detected!
Gas is detected!
Gas is detected!
Gas is not detected
Gas is not detected
Gas is not detected
Gas is detected!
Gas is detected!
```

Рисунок 3.27 – Результат роботи коду для MQ-135

Тепер, напишемо програмний приклад для модуля SIM800C GSM/GPRS НАТ. Від тієї ж компанії Adafruit є прекрасний мануал з переліком усіх АТ-команд – спеціальних команд для GSM-модемів, розроблених компанією Hayes – та їх синтаксисом, які ми можемо інтегрувати у програмний код. Напишемо код з використанням АТ-команд на мові програмування Python (рисунок 3.29): давач MQ-135 буде кожні 5 хвилин перевіряти наявність шкідливого газу у приміщенні, де він встановлений, та, при його наявності, модуль SIM800C на заданий номер телефону буде відправляти SMS-повідомлення з сигналом про це з періодичністю раз в 30 секунд. До того ж, у програмі буде реалізована додаткова функція: при отриманні SMS-команди зі сторони адресанта, він буде ділитись температурою

процесора Raspberry Pi, через ту ж технологію SMS. Його робота – асинхронна, проте, для яснішої демонстрації на зображеннях, блоки програмного коду для кожної функції були запущені роздільно. Результат роботи коду на рисунках 3.28 – 3.31:

```

File Edit Tabs Help
Serial port opened successfully. Initializing...
ATZ
OK

AT+CMGF=1
OK

---CMGLs number: 4

46,"REC READ","Kyivstar","", "24/06/08, 23:21:38+12"
04350440044304320430044204380020043D043E043C04350440043E043C002C0020043
3E0441043B044304330438002C0020043F043504400435043204560440044F044204380
47,"REC READ","Kyivstar","", "24/06/08, 23:21:38+12"
043F043E043F043E0432043D044E043204300442043800200440043004450443043D043
5604570020043C043E0436043D04300020044300200434043E043404300442043A04430
48,"REC READ","Kyivstar","", "24/06/08, 23:21:38+12"
0442043004400020006D0079006B0079006900760073007400610072002E00700061006

---CMGLs number: 4

46,"REC READ","Kyivstar","", "24/06/08, 23:21:38+12"
04350440044304320430044204380020043D043E043C04350440043E043C002C0020043
3E0441043B044304330438002C0020043F043504400435043204560440044F044204380
47,"REC READ","Kyivstar","", "24/06/08, 23:21:38+12"
043F043E043F043E0432043D044E043204300442043800200440043004450443043D043
5604570020043C043E0436043D04300020044300200434043E043404300442043A04430
48,"REC READ","Kyivstar","", "24/06/08, 23:21:38+12"
0442043004400020006D0079006B0079006900760073007400610072002E00700061006

---CMGLs number: 4

46,"REC READ","Kyivstar","", "24/06/08, 23:21:38+12"
04350440044304320430044204380020043D043E043C04350440043E043C002C0020043
3E0441043B044304330438002C0020043F043504400435043204560440044F044204380
47,"REC READ","Kyivstar","", "24/06/08, 23:21:38+12"
043F043E043F043E0432043D044E043204300442043800200440043004450443043D043
5604570020043C043E0436043D04300020044300200434043E043404300442043A04430
48,"REC READ","Kyivstar","", "24/06/08, 23:21:38+12"
0442043004400020006D0079006B0079006900760073007400610072002E00700061006

---CMGLs number: 5
+CMTI: "SM",1
1,"REC UNREAD","+380971", "", "24/06/11, 19:31:19+12"
get_temperature
Received 'get_temperature' command at CMGL - 1
temp=34.5'C

Message with temperature of Raspberry Pi sent

AT+CMGD=1
46,"REC READ","Kyivstar","", "24/06/08, 23:21:38+12"
04350440044304320430044204380020043D043E043C04350440043E043C002C0020043
3E0441043B044304330438002C0020043F043504400435043204560440044F044204380
47,"REC READ","Kyivstar","", "24/06/08, 23:21:38+12"
043F043E043F043E0432043D044E043204300442043800200440043004450443043D043
5604570020043C043E0436043D04300020044300200434043E043404300442043A04430
48,"REC READ","Kyivstar","", "24/06/08, 23:21:38+12"
0442043004400020006D0079006B0079006900760073007400610072002E00700061006

```

Рисунок 3.28 – Результат виконання програмного коду для функції надсилання температури за SMS-запитом

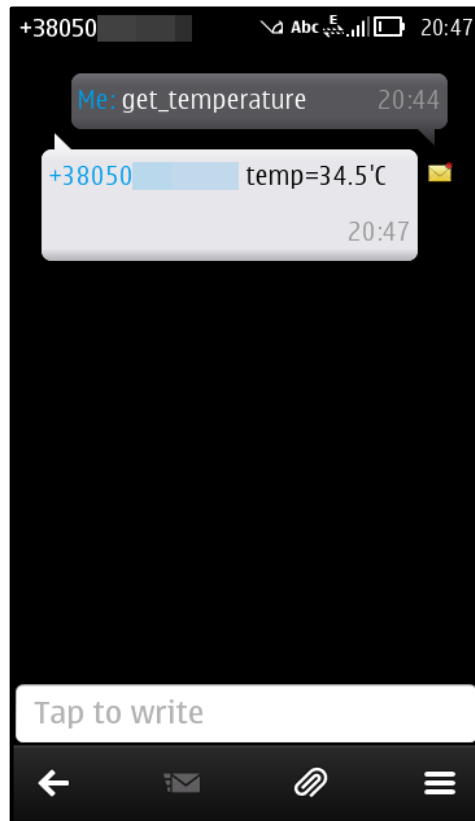


Рисунок 3.29 – Отримання відповіді на SMS-запит

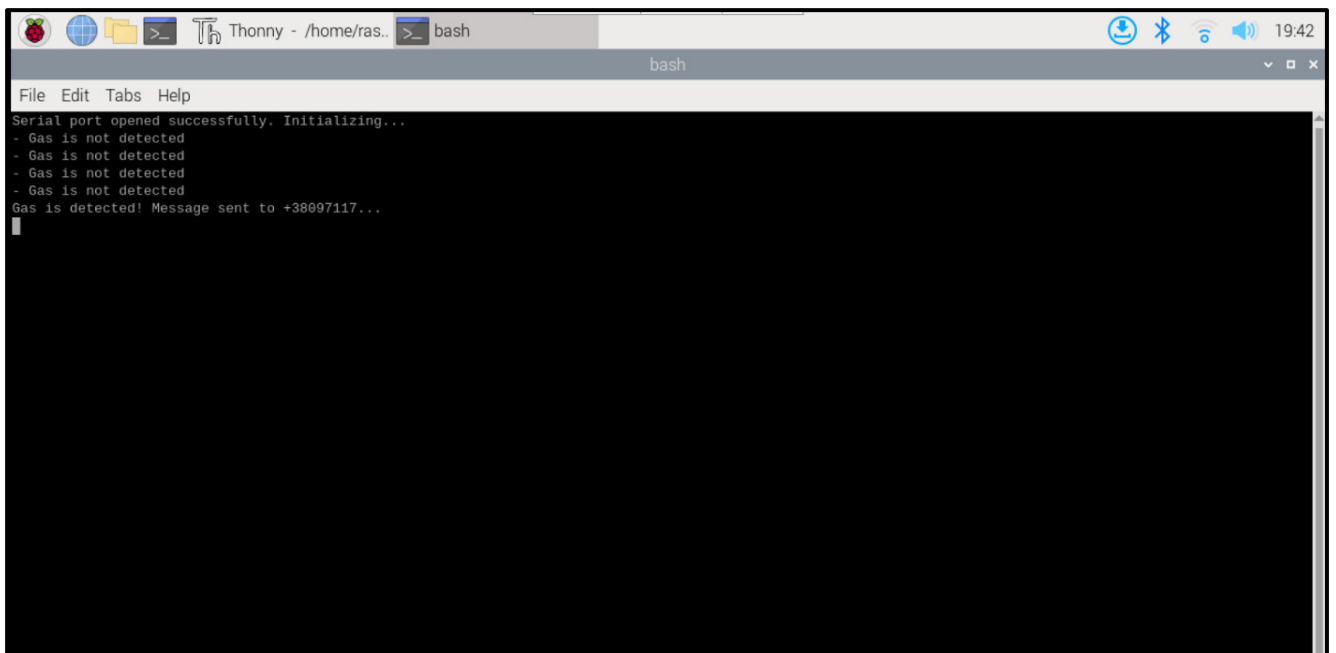


Рисунок 3.30 – Результат виконання програмного коду для сигналізування про наявність шкідливого газу по SMS

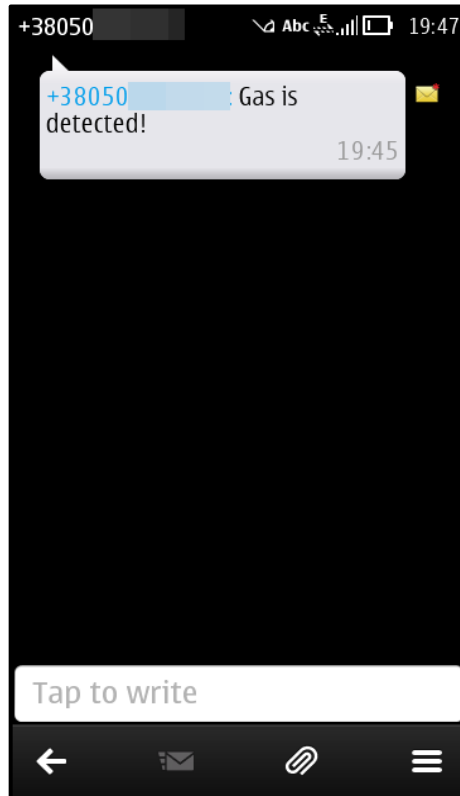


Рисунок 3.31 – Отримання повідомлення з сигналом про наявність газу

Як видно, пристрої працюють успішно. Лістинг програмних кодів для кожного з них буде зображено у додатку Г.

Програмне забезпечення Domoticz має розроблений прикладний програмний інтерфейс Domoticz API, розширюючий його функціональні можливості [58]. Працює він за текстовим форматом обміну даними – JSON. Формат запити виклику API для версії Domoticz Stable 2024.4 – <http://<username:password@>domoticz-ip<:port>/json.htm?api-call> – де:

1. <username:password@> – ім'я користувача й пароль для доступу до Domoticz.
2. domoticz-ip – IP-адреса пристрою.
3. <:port> – номер порту для підключення.

Так як в системі присутні давач освітлення та комбінований давач температури, тиску та вологості, відповідні API-запити для кожного з них виглядатимуть так:

1. Давач

освітлення

–

/json.htm?type=command&param=udevice&idx=IDX&nvalue=0&svalue=VOLTAGE

– де:

a. IDX – ідентифікатор девайсу (цей номер може бути знайдений у таблиці девайсів, у розділі «IDX»).

b. VALUE – значення напруги Давача, у Вольтах.

2. Комбінований давач температури, тиску та вологості

–

/json.htm?type=command&param=udevice&idx=IDX&nvalue=0&svalue=TEMP;HU

M;HUM\_STAT;BAR;BAR\_FOR – де:

a. IDX – ідентифікатор девайсу (цей номер може бути знайдений у таблиці девайсів, у розділі «IDX»).

b. TEMP – температура.

c. HUM – вологість.

d. HUM\_STAT – статус вологості:

1) 0 – нормальний рівень;

2) 1 – комфортний рівень;

3) 2 – низький рівень;

4) 3 – високий рівень;

e. BAR – барометричний тиск.

f. BAR\_FOR – прогноз барометра:

1) 0 – немає інформації;

2) 1 – сонячно;

3) 2 – частково хмарно;

4) 3 – хмарно;

5) 4 – дощ.

Дане API можна використати для забезпечення передачі даних до фізичного або хмарного сервера для подальшої їх обробки, сортування, зберігання у БД з відповідною СКБД та демонстрації їх у клієнтському додатку з допомогою графічних елементів – діаграм, графіків та схожого.

### 3.2.2 Налаштування віртуальної приватної мережі – VPN

У мережі Інтернет кожен пристрій має свою глобальну індивідуальну адресу, – IP – котра є публічною для ресурсу, на якому він знаходиться. Публічна, справжня IP-адреса пристрою створює певні безпекові ризики, на кшталт несанкціонованого доступу до пристрою за допомогою відкритих на ньому програмних портів, перехват трафіку з користувацькими даними та схожі небагатодійні аспекти. В даній ситуації, так як для шлюзу досить важливо забезпечувати належний рівень безпеки передачі даних до хмарного середовища у цій же мережі Інтернет, було забезпечено віртуальну приватну мережу, до якої пристрій, перед доступом до того чи іншого ресурсу (в даному випадку, хмарного серверу) в інтернеті, буде під'єднуватись й лиш після цього передавати вже пропущений через VPN-сервер трафік, котрий, може бути як просто пропущений через сервер в цілях зміни IP-адреси (так зване проксування), так і, додатково, може бути обфускований, зашифрований та/або яким-небудь іншим чином видозмінений в цілях безпеки. Принцип роботи технології VPN [59] зображено на рисунку 3.32:

## СХЕМА РОБОТИ VPN

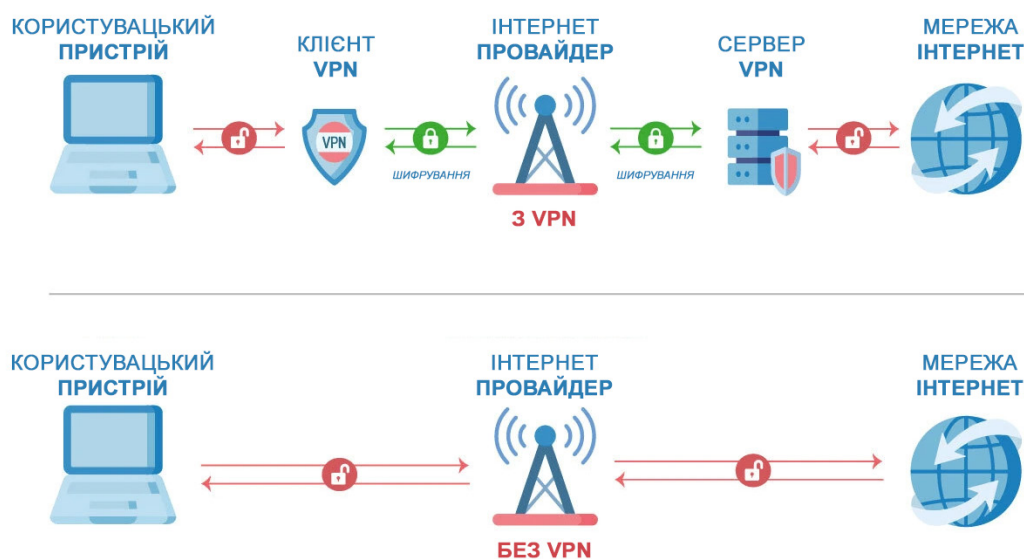


Рисунок 3.32 – Принцип роботи VPN [59]

На стороні користувача встановлюється клієнт, котрий, перед доступом до інтернету, приєднується до VPN-сервера за наданими виробником конфігураційними даними, зазвичай – IP-адресою, портом для підключення, відкритим ключем сервера та закритим ключем клієнта. На ринку існує широка кількість виробників програмного забезпечення даного роду, як платного, так і безоплатного. Приклади – GhostVPN, Norton, ExpressVPN, Mullvad и схожих. У нашому випадку буде організовуватись власний VPN-сервер на базі іншого комп'ютера, з доступом до нього через клієнт на нашому шлюзі.

Для даної задачі існує немало рішень теж, найширше використовувані у світі – WireGuard [60] та OpenVPN [61]. Вибір пав на WireGuard – вільне програмне забезпечення для створення VPN-тунелів. Його перевагою, в порівнянні з OpenVPN, є менші енерговитрати, більша швидкість та простіша конфігурація [62]. Проводитиметься робота на базі PiVPN – скрипта, спрощуючого створення VPN на базі вищеназваного ПЗ [63]. Для початку – встановимо його, вводимо в консолі команду `curl -L https://install.pivpn.io | bash` (рисунок 3.33).

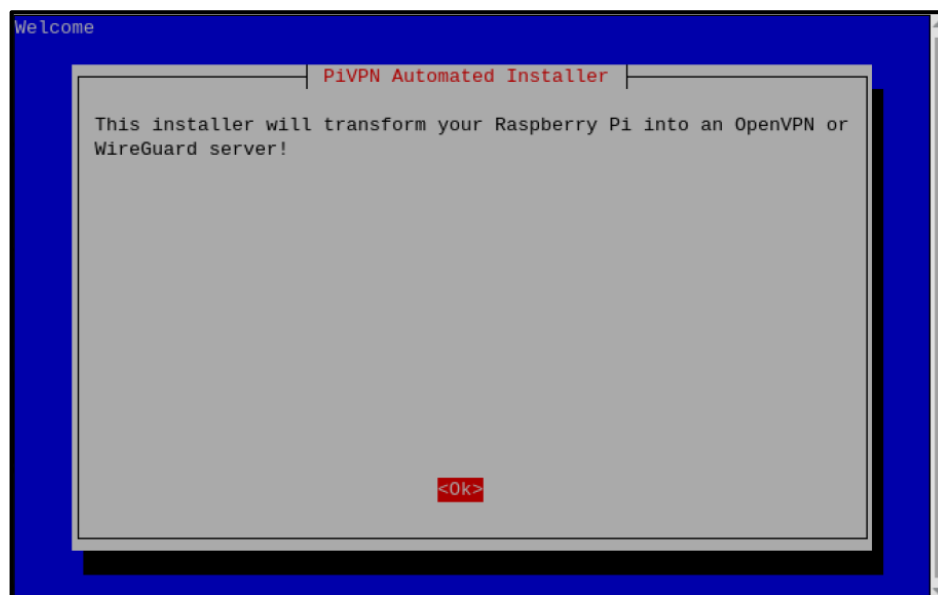


Рисунок 3.33 – Вікно встановлення PiVPN

За умови ввімкненої на маршрутизаторі функції DHCP, – технології для автоматичного задання IP-адрес пристроїв у мережі – для запобігання конфліктів

IP-адрес у мережі, в ході встановлення PiVPN необхідно ввімкнути DHCP-резервацію (рисунок 3.34). Так як це сервер, його IP-адреса, для коректного доступу до нього, повинна бути статичною, на етапі запиту її вибору, відповідно, обираємо «Так». Після вибору, для перевірки правильності даних, користувачу будуть продемонстровані IP-адреса та шлюз для підключення (рисунок 3.35).

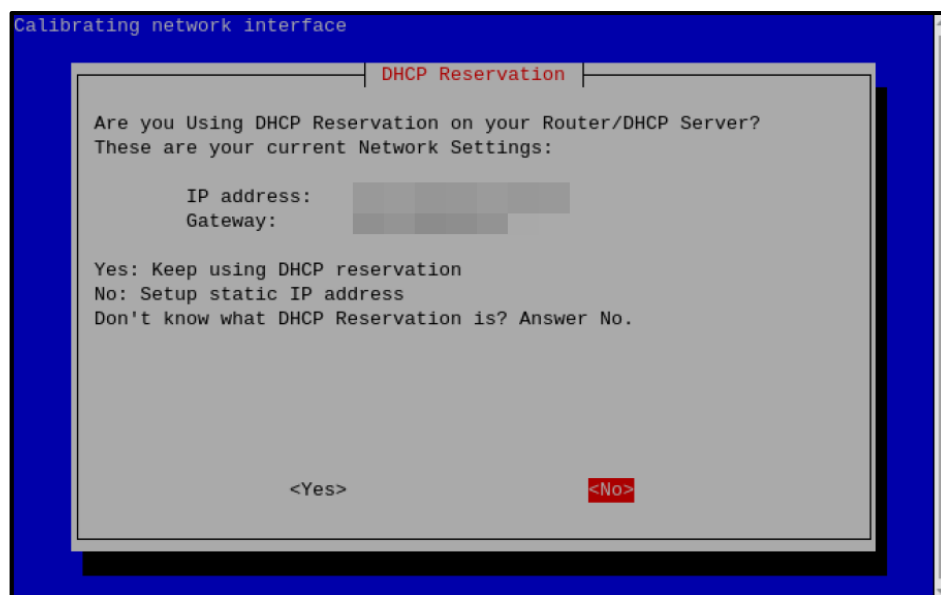


Рисунок 3.34 – Налаштування DHCP для VPN-сервера



Рисунок 3.35 – Демонстрація IP-адреси майбутнього VPN-сервера





Рисунок 3.38 – Показ програмного порта для роботи WireGuard

Тепер необхідно обрати DNS-провайдера. Одні з найвідоміших – Google, Norton, CloudFlare. Обираємо (рисунок 3.39), у випадку незадоволення вибором його можна легко переобрати пізніше:

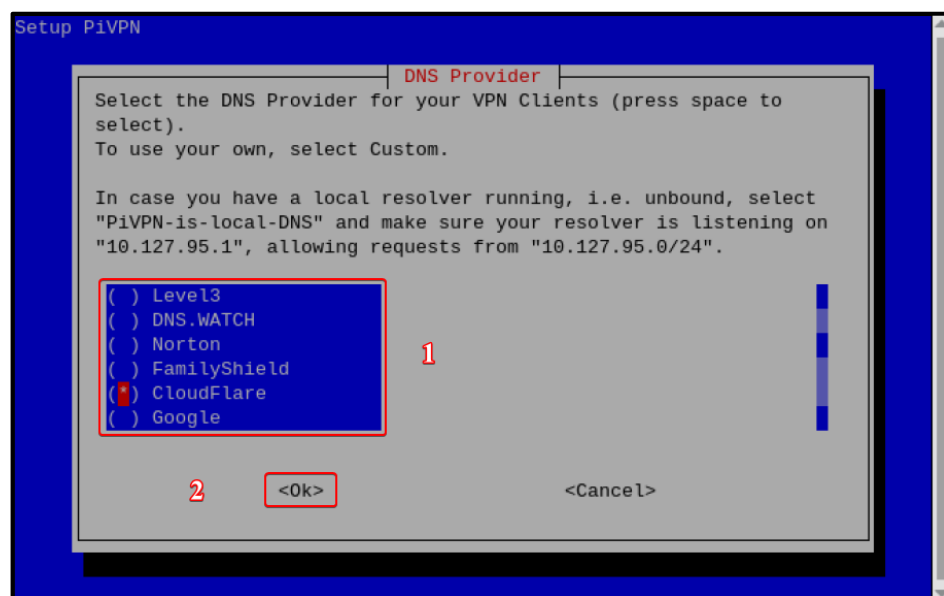


Рисунок 3.39 – Список DNS-провайдерів для роботи WireGuard у PiVPN

Після вибору DNS-провайдера вибираємо метод підключення до VPN, DNS-чи IP-адреса (рисунок 3.40):

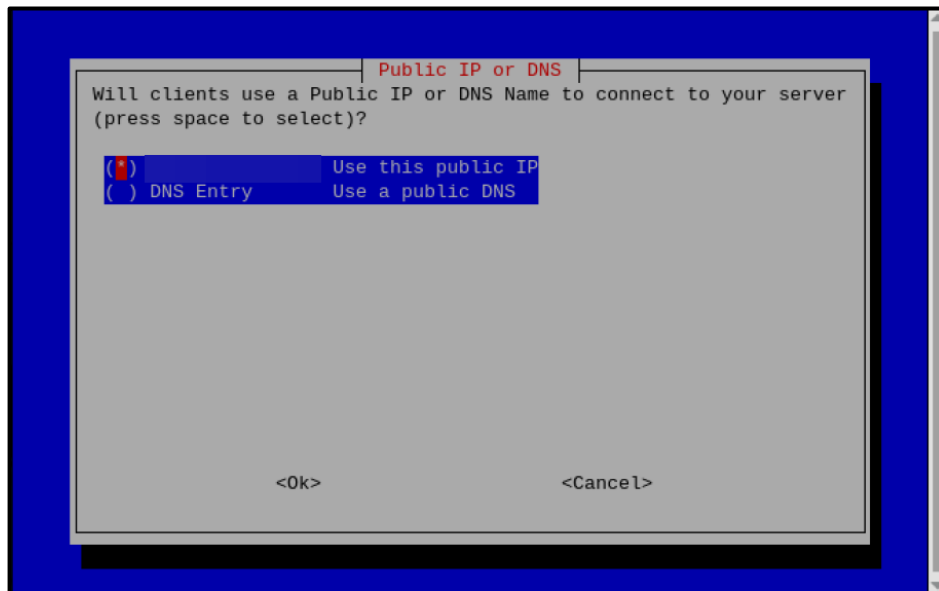


Рисунок 3.40 – Методи підключення до VPN

Далі, для забезпечення стабільнішої та безпечнішої роботи WireGuard, даємо дозвіл на автоматичне завантаження оновлень/патчів для WireGuard (рисунок 3.41):

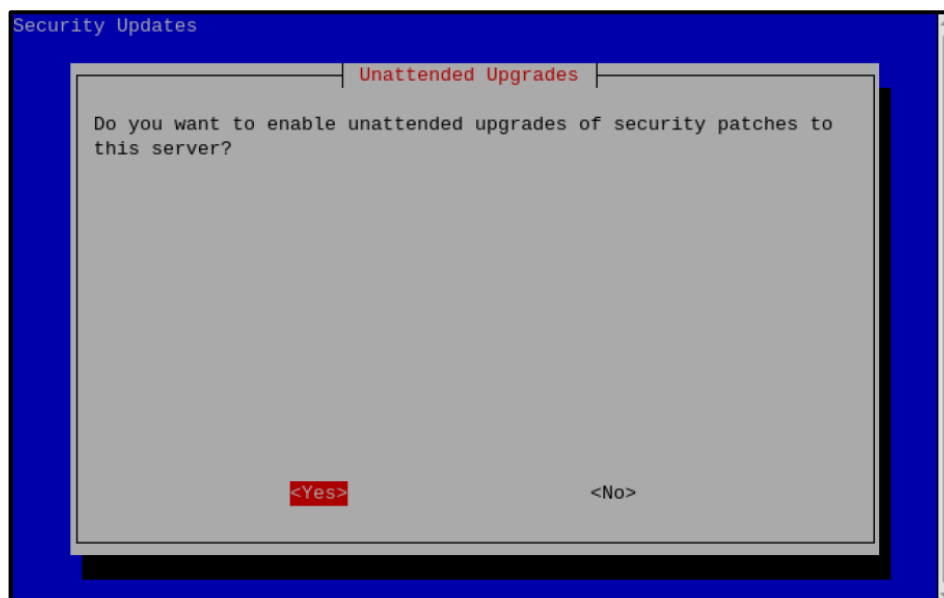


Рисунок 3.41 – Запит дозволу на автоматичне завантаження оновлень для WireGuard

Програмне забезпечення PiVPN на базі WireGuard – встановлено (рисунок 3.41). Далі – необхідне перезавантаження Raspberry Pi (рисунок 3.42).

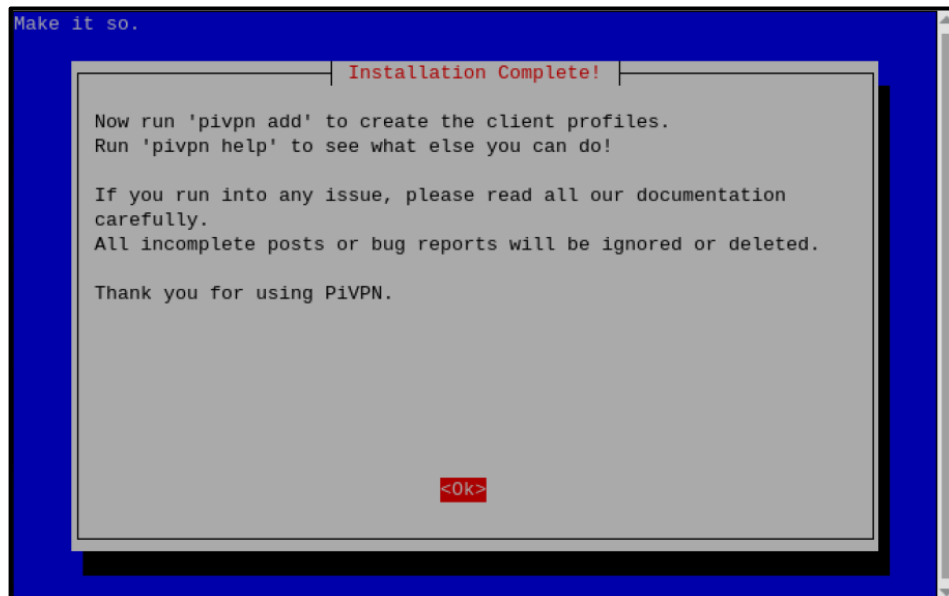


Рисунок 3.41 – Сповіднення про встановлення PiVPN

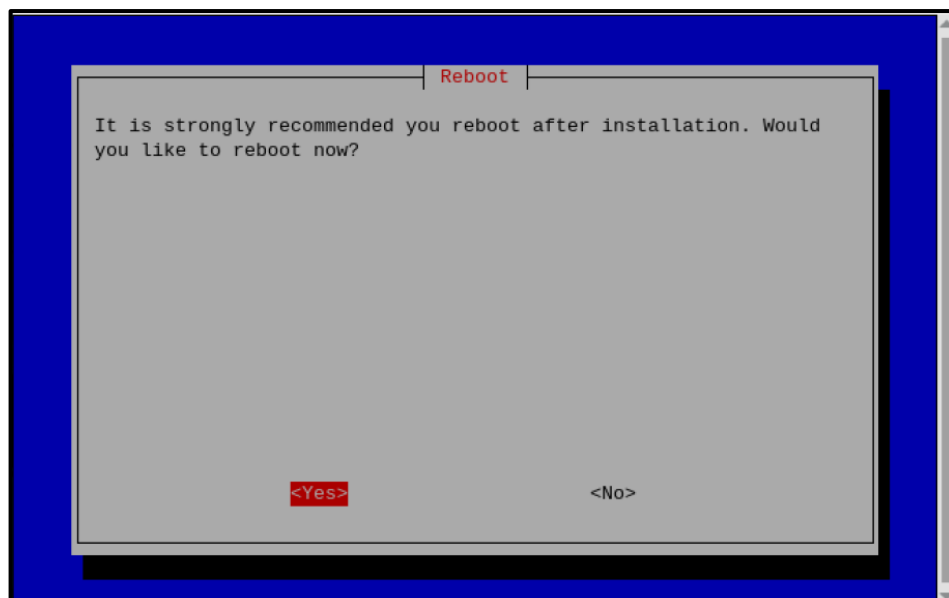
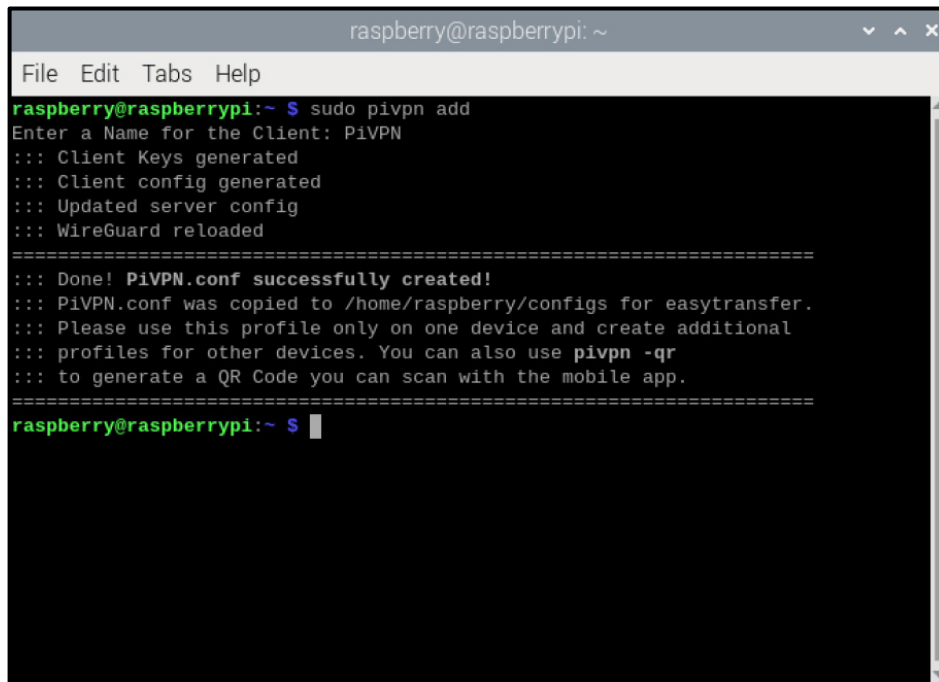


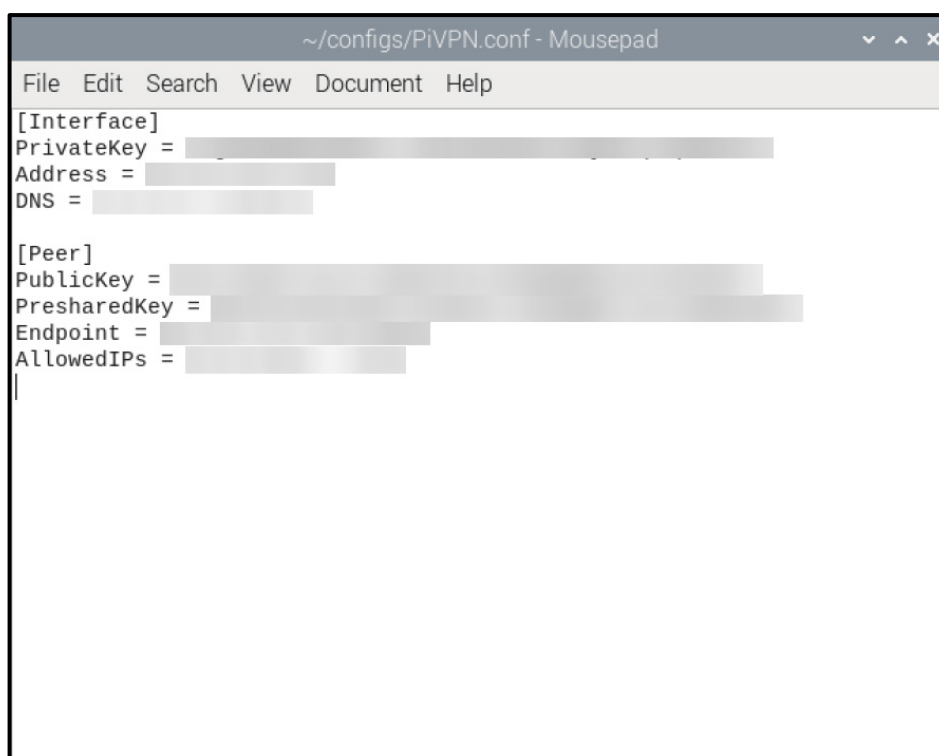
Рисунок 3.42 – Запит на перезавантаження Raspberry Pi у PiVPN

Далі, після перезапуску Raspberry Pi, вводимо в тій же консолі команду для створення нового профілю для VPN-клієнта – `sudo pivpn add` (рисунок 3.43). Задаємо ім'я профілю й, заодно, конфігураційного файлу та зберігаємо їх. За замовчуванням, файл зберігається у директорії `/home/«ім'я_користувача»/configs`, де ми можемо переглянути його зміст й ознайомитись із базовим синтаксисом для створення профілю у WireGuard (рисунок 3.44).



```
raspberrypi@raspberrypi: ~  
File Edit Tabs Help  
raspberrypi@raspberrypi:~$ sudo pivpn add  
Enter a Name for the Client: PiVPN  
::: Client Keys generated  
::: Client config generated  
::: Updated server config  
::: WireGuard reloaded  
=====  
::: Done! PiVPN.conf successfully created!  
::: PiVPN.conf was copied to /home/raspberrypi/configs for easytransfer.  
::: Please use this profile only on one device and create additional  
::: profiles for other devices. You can also use pivpn -qr  
::: to generate a QR Code you can scan with the mobile app.  
=====  
raspberrypi@raspberrypi:~$
```

Рисунок 3.43 – Створення профілю у PiVPN



```
~/configs/PiVPN.conf - Mousepad  
File Edit Search View Document Help  
[Interface]  
PrivateKey = [REDACTED]  
Address = [REDACTED]  
DNS = [REDACTED]  
  
[Peer]  
PublicKey = [REDACTED]  
PresharedKey = [REDACTED]  
Endpoint = [REDACTED]  
AllowedIPs = [REDACTED]  
|
```

Рисунок 3.44 – Файл конфігурації, створений за допомогою PiVPN

В даному файлі, в блокові «[Interface]» (інтерфейс, з допомогою якого клієнт взаємодіє із сервером), задані IP-адреса й маска підмережі клієнта, його приватний ключ й DNS-провайдер; в блокові «[Peer]» (вузол сервера, до якого під'єднується

інтерфейс клієнта), задані IP-адреса й порт сервера, його публічний ключ й дозволені IP-адреси клієнтів. У випадку VPN, ключ – гарант безпечного з'єднання з сервером, який запобігає несанкціонованому доступу з боку інших користувачів; WireGuard, для безпечної своєї роботи, використовує метод асиметричного шифрування, де є закриті й відкриті пари ключів для кожного профіля клієнта й сервера.

Крім того, можна створити QR-код для комфортнішого підключення до нового профіля VPN-сервера, з допомогою команди `pivpn -qr «PROFILENAME»` (рисунок 3.45):



Рисунок 3.45 – Створення QR-коду для заданого профілю у PiVPN

Нюанс PiVPN в тому, що файл конфігурації вона задає у вільній директорії, до якої доступ є у всіх користувачів. Для запобігання цього та, додатково, загального розуміння принципу налаштування VPN-сервера WireGuard з допомогою профіля, розберемо поетапно процес його налаштування у самому WireGuard без участі PiVPN:





```
raspberrypi@raspberrypi: ~  
File Edit Tabs Help  
GNU nano 7.2 /etc/sysctl.conf *  
  
# Uncomment the next line to enable packet forwarding for IPv4  
net.ipv4.ip_forward=1  
  
# Uncomment the next line to enable packet forwarding for IPv6  
# Enabling this option disables Stateless Address Autoconfiguration  
# based on Router Advertisements for this host  
net.ipv6.conf.all.forwarding=1  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

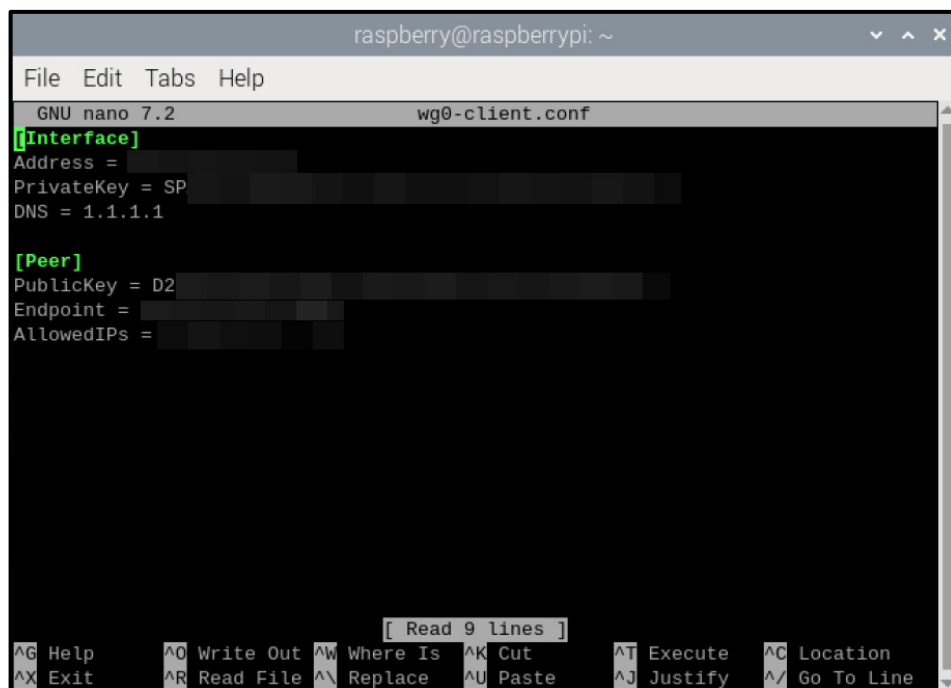
Рис 3.48 – Налаштування дозволу на пересилання трафіку

```
raspberrypi@raspberrypi: ~  
File Edit Tabs Help  
  
root@raspberrypi:/etc/wireguard# systemctl enable wg-quick@wg0  
root@raspberrypi:/etc/wireguard# chown -R root:root /etc/wireguard/  
root@raspberrypi:/etc/wireguard# chmod -R og-rwx /etc/wireguard/*  
root@raspberrypi:/etc/wireguard# nano wg0.conf  
root@raspberrypi:/etc/wireguard# exit  
exit  
raspberrypi@raspberrypi:~$ sudo reboot now
```

Рисунок 3.49 – Налаштування автозапуску WireGuard

Після перезапуску, переходимо до цієї ж директорії, як у пунктах 1-2 налаштування конфігурації сервера. Тепер, поетапно й без участі PiVPN розберемо процес налаштування конфігурації клієнта:

1. Генеруємо файл конфігурації клієнта й заодно редагуємо його у текстовому редакторі – nano wg0-client.conf.
2. Відповідно до того ж синтаксису, зображеному на рисунку 3.43, створимо новий конфігураційний файл для VPN-клієнта, із заданими власною IP-адресою, DNS-провайдером, згенерованими ключами й схожими, необхідними нам атрибутами (рисунок 3.48) та збережемо його. Код для файлу конфігурації VPN-клієнта буде продемонстровано у додатку Г.



```
raspberrypi@raspberrypi: ~
File Edit Tabs Help
GNU nano 7.2 wg0-client.conf
[Interface]
Address =
PrivateKey = SP
DNS = 1.1.1.1

[Peer]
PublicKey = D2
Endpoint =
AllowedIPs =

[ Read 9 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

Рисунок 3.50 – Файл конфігурації VPN-клієнта WireGuard

Налаштування VPN-тунелю – готово, лістинг програмних кодів для файлів конфігурації буде вказано в додатку Г. Для наглядності, запусимо файл на клієнті WireGuard на ПК (рисунок 3.51). Як видно, біля інтерфейсу з’єднання з мережею з’явилась назва файлу конфігурації клієнта, конфігурація сервера й клієнта задана вірно. Проте, якщо ПК підключений, наприклад, до VNC-сервера на Raspberry Pi чи інтернету, можна замітити, що на ньому з’єднання із ними – зникло.





## ВИСНОВКИ

У даній кваліфікаційній роботі проведено аналіз підрозділу концепції кіберфізичних систем, званого як Інтернет речей – концепції передачі даних з допомогою комп'ютерів між фізичними об'єктами у мережі Інтернет. Розкрито його основні поняття, сенс, структурні елементи та роль у сфері інформаційних технологій. Було проведено аналіз технологій передавання інформації в мережі Інтернету речей, що дало змогу виявити необхідні якості кожної з них та обрати оптимальний технологічний інструментарій для виконання практичного завдання.

Завданням даної кваліфікаційної роботи було створення структурного елементу системи Інтернету речей, званого шлюзом – пристроєм, пов'язуючим крайові елементи системи Інтернету речей із середовищем обробки даних, зібраних із них. Для його функціонування було забезпечено належний вибір платформи Raspberry Pi з необхідним апаратно-програмним інструментарієм, котрий дозволив об'єднати наявні в моделі даччі та, з допомогою програмного забезпечення для автоматизації «Domoticz», систематизувати отримані дані у комфортному для адміністратора графічному інтерфейсі, демонструючому стан системи наглядно. Додатково, для шлюзу Інтернету речей, окрім вже наявних технологій передачі даних, було забезпечено можливість комунікації з клієнтами за допомогою мережі мобільного стільникового зв'язку, функціонально розширюючим можливості даного шлюзу на базі Raspberry Pi. Задля забезпечення стабільності роботи шлюзу та збереження цілісності передаваних до середовища обчислення даних, було забезпечено VPN-з'єднання, роблячим комунікацію шлюзу у інтернеті безпечнішим та надійнішим.

|      |      |         |        |      |                          |            |
|------|------|---------|--------|------|--------------------------|------------|
|      |      |         |        |      | КВРКІ 200226.21.04.83 ПЗ | Арк.<br>67 |
| Зм.. | Арк. | №докум. | Підпис | Дата |                          |            |

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Sumatosoft. What is an Internet of Things (IoT) Gateway: Functions, Types, Examples. URL: <https://sumatosoft.com/blog/what-is-an-internet-of-things-iot-gateway-functions-types-examples> (дата звернення: 6.03.2025).

2. Lanner Inc. What is an IoT Gateway – 2022 Update. URL: <https://www.lannerinc.com/news-and-events/eagle-lanner-tech-blog/what-is-an-iot-gateway-2022-update> (дата звернення: 6.03.2025).

3. Statista. Internet of Things (IoT) total annual revenue worldwide from 2020 to 2030. URL: <https://www.statista.com/statistics/1194709/iot-revenue-worldwide/> (дата звернення: 6.03.2025).

4. Global Market Insights. IoT Gateway Devices Market – By Connectivity Technology, By Component, By Application & Forecast, 2021-2027. URL: <https://www.gminsights.com/industry-analysis/iot-gateway-devices-market> (дата звернення: 7.03.2025).

5. Dusun. Demystifying IoT Gateways: The Engine Behind Seamless Communication in IoT. URL: <https://www.dusuniot.com/blog/demystifying-iot-gateway-devices/> (дата звернення: 7.03.2025).

6. Microsoft Azure. Технологии и протоколы Интернета вещей. URL: <https://azure.microsoft.com/ru-ru/solutions/iot/iot-technology-protocols/> (дата звернення: 8.03.2025).

7. Rehack. The Best Open-Source Home Automation Platforms for 2025. URL: <https://rehack.com/iot/smart-home/open-source-home-automation/> (дата звернення: 8.03.2025).

8. AlternativeTo. Domoticz Alternatives. URL: <https://alternativeto.net/software/domoticz/?p=4> (дата звернення: 8.03.2025).

9. Dusun. Best IoT Gateways for 2024: Secure and Streamlined. URL: <https://www.dusuniot.com/blog/best-iot-gateways-secure-and-streamlined/> (дата звернення: 9.03.2025).

|      |      |         |        |      |  |                          |            |
|------|------|---------|--------|------|--|--------------------------|------------|
|      |      |         |        |      |  | КВРКІ 200226.21.04.83 ПЗ | Арк.<br>68 |
| Зм.. | Арк. | №докум. | Підпис | Дата |  |                          |            |

10. TechTarget. IoT gateway. URL: <https://www.techtarget.com/iotagenda/definition/IoT-gateway> (дата звернення: 12.03.2025).

11. WebbyLab. IoT Gateways Explained: A Detailed Guide for 2025. URL: <https://webbylab.com/blog/iot-gateways-explained-a-detailed-guide/> (дата звернення: 16.03.2025).

12. Coursera. What Is the Internet of Things (IoT)? With Examples. URL: <https://www.coursera.org/articles/internet-of-things> (дата звернення: 16.03.2025).

13. Device Authority. The Ultimate Guide to IoT Gateways: Everything You Need to Know. URL: <https://www.deviceauthority.com/blog/the-ultimate-guide-to-iot-gateways-everything-you-need-to-know/> (дата звернення: 18.03.2025).

14. Thales Group. Bridge the Gap with IoT Gateways. URL: <https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/inspired/iot-gateway> (дата звернення: 20.03.2025).

15. Particle. A Guide to IoT Gateways. URL: <https://www.particle.io/iot-guides-and-resources/iot-gateway/> (дата звернення: 20.03.2025).

16. Bivocom. Bivocom TG451 Industrial IoT Gateway. URL: <https://www.bivocom.com/products/iot-gateways/industrial-gateway-tg451> (дата звернення: 26.03.2025).

17. Advantech. Advantech UTX-3117. URL: [https://www.advantech.com/en/products/bda911fe-28bc-4171-aed3-67f76f6a12c8/utx-3117/mod\\_9a201cd3-1416-4282-9fc4-de8ca1e6bcbc](https://www.advantech.com/en/products/bda911fe-28bc-4171-aed3-67f76f6a12c8/utx-3117/mod_9a201cd3-1416-4282-9fc4-de8ca1e6bcbc) (дата звернення: 26.03.2025).

18. AAEON. AAEON Industrial IoT Gateway Solutions. URL: <https://www.aaeon.com/en/c/iot-gateway-solutions> (дата звернення: 26.03.2025).

19. Dell. Dell Edge Gateway 5200. URL: [https://www.dell.com/en-us/shop/cty/pdp/spd/dell-edge-gateway-5200/dn\\_edge5200\\_15513\\_vi](https://www.dell.com/en-us/shop/cty/pdp/spd/dell-edge-gateway-5200/dn_edge5200_15513_vi) (дата звернення: 29.03.2025).

20. Amazon Web Services. Что такое MQTT?. URL: <https://aws.amazon.com/ru/what-is/mqtt/> (дата звернення: 29.03.2025).

|      |      |         |        |      |                          |            |
|------|------|---------|--------|------|--------------------------|------------|
|      |      |         |        |      | КВРКІ 200226.21.04.83 ПЗ | Арк.<br>69 |
| Зм.. | Арк. | №докум. | Підпис | Дата |                          |            |

21. Wikipedia. Wi-Fi. URL: <https://en.wikipedia.org/wiki/Wi-Fi> (дата звернення: 29.03.2025).
22. Wikipedia. DASH7. URL: <https://en.wikipedia.org/wiki/DASH7> (дата звернення: 29.03.2025).
23. Laird Connect. Laird 450-0190. URL: <https://www.lairdconnect.com/part/450-0190> (дата звернення: 1.04.2025).
24. Z-Wave Alliance. How Z-Wave works. URL: <https://www.z-wave.com/learn> (дата звернення: 1.04.2025).
25. Bluetooth. Bluetooth. URL: <https://www.bluetooth.com/> (дата звернення: 1.04.2025).
26. Wikipedia. Bluetooth Low Energy. URL: [https://en.wikipedia.org/wiki/Bluetooth\\_Low\\_Energy](https://en.wikipedia.org/wiki/Bluetooth_Low_Energy) (дата звернення: 06.04.2025).
27. GeeksforGeeks. Near Field Communication (NFC). URL: <https://www.geeksforgeeks.org/near-field-communication-nfc/> (дата звернення: 6.04.2025).
28. Dusun. What is ZigBee in IoT? The Definitive Guide in 2024. – URL: <https://www.dusuniot.com/blog/the-definitive-guide-for-zigbee-gateway-for-solution-vendors/> (дата звернення: 13.03.2025).
29. Wikipedia. LoRa. URL: <https://en.wikipedia.org/wiki/LoRa> (дата звернення: 7.04.2025).
30. Amazon Web Services. What is LoRaWAN?. URL: <https://docs.aws.amazon.com/iot/latest/developerguide/connect-iot-lorawan-what-is-lorawan.html> (дата звернення: 8.04.2025).
31. Wikipedia. LTE. URL: <https://ru.wikipedia.org/wiki/LTE> (дата звернення: 10.04.2025).
32. Wikipedia. LTE-M. URL: <https://en.wikipedia.org/wiki/LTE-M> (дата звернення: 10.04.2025).
33. Wikipedia. Radio-frequency identification (RFID). URL: [https://en.wikipedia.org/wiki/Radio-frequency\\_identification](https://en.wikipedia.org/wiki/Radio-frequency_identification) (дата звернення: 10.04.2025).

|     |      |         |        |      |                          |            |
|-----|------|---------|--------|------|--------------------------|------------|
|     |      |         |        |      | КВРКІ 200226.21.04.83 ПЗ | Арк.<br>70 |
| Зм. | Арк. | №докум. | Підпис | Дата |                          |            |

34. NB-Fi Alliance. NB-Fi – the IoT Standard. URL: <https://nb-fi.org/resources/nb-fi-specification-overview/> (дата звернення: 10.04.2025).

35. Ethernet. URL: <https://en.wikipedia.org/wiki/Ethernet> (дата звернення: 15.04.2025).

36. Eclipse Kura. URL: <https://eclipse.dev/kura/> (дата звернення: 15.04.2025).

37. Ubiworx. URL: <https://ubiorx.com/> (дата звернення: 19.04.2025).

38. openHAB. URL: <https://www.openhab.org/> (дата звернення: 22.04.2025).

39. Home Assistant. URL: <https://www.home-assistant.io/> (дата звернення: 23.04.2025).

40. Domoticz. URL: <https://www.domoticz.com/> (дата звернення: 23.04.2025).

41. Contiki-NG. URL: <https://www.contiki-ng.org/> (дата звернення: 25.04.2025).

42. Les Pounder: Raspberry Pi celebrates 12 years as sales break 61 million units. URL: <https://www.tomshardware.com/raspberry-pi/raspberry-pi-celebrates-12-years-as-sales-break-61-million-units> (дата звернення: 30.04.2025).

43. Waveshare Wiki. URL: [https://www.waveshare.com/wiki/Main\\_Page](https://www.waveshare.com/wiki/Main_Page). (дата звернення: 1.05.2025).

44. DATASHEET Raspberry Pi 4B. URL: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf> (дата звернення: 2.05.2025).

45. Introduction to IOT – IoT Master Class. URL: <https://www.pantechsolutions.net/introduction-to-iot-iot-master-class-day-1> (дата звернення: 4.05.2025).

46. Pressure sensor BMP280. URL: <https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp280/> (дата звернення: 5.05.2025).

47. TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER. URL: <https://cdn-shop.adafruit.com/datasheets/TSL2561.pdf> (дата звернення: 5.05.2025).

48. MQ-135 – Gas Sensor for Air Quality. URL: <https://components101.com/sensors/mq135-gas-sensor-for-air-quality> (дата звернення: 5.05.2025).

49. SIM800C GSM/GPRS HAT. URL:  
[https://www.waveshare.com/wiki/SIM800C\\_GSM/GPRS\\_HAT](https://www.waveshare.com/wiki/SIM800C_GSM/GPRS_HAT) (дата звернення:  
5.05.2025).

50. Raspberry Pi Documentation. URL:  
<https://www.raspberrypi.com/documentation/computers/getting-started.html> (дата  
звернення: 6.05.2025).

51. How to use dual boot on Raspberry Pi?. URL:  
<https://raspberrytips.com/raspberry-pi-dual-boot/> (дата звернення: 6.05.2025).

52. Smbus. URL: <https://smbus2.readthedocs.io/en/latest/index.html> (дата  
звернення: 8.05.2025).

53. I2C Tools. URL:  
[https://archive.kernel.org/oldwiki/i2c.wiki.kernel.org/index.php/I2C\\_Tools.html](https://archive.kernel.org/oldwiki/i2c.wiki.kernel.org/index.php/I2C_Tools.html) (дата  
звернення: 8.05.2025).

54. CircuitPython. URL: <https://en.wikipedia.org/wiki/CircuitPython> (дата  
звернення: 8.05.2025).

55. Minicom. URL: <https://en.wikipedia.org/wiki/Minicom> (дата звернення:  
8.05.2025).

56. How to start using minicom. URL:  
<https://forums.raspberrypi.com/viewtopic.php?t=309854> (дата звернення: 9.05.2025).

57. Hayes AT command set. URL:  
[https://en.wikipedia.org/wiki/Hayes\\_AT\\_command\\_set](https://en.wikipedia.org/wiki/Hayes_AT_command_set) (дата звернення: 9.05.2025).

58. Domoticz API/JSON URL's. URL:  
[https://www.domoticz.com/wiki/Domoticz\\_API/JSON\\_URL%27s](https://www.domoticz.com/wiki/Domoticz_API/JSON_URL%27s) (дата звернення:  
12.05.2025).

59. The Ultimate Guide to VPNs: What Does a VPN Protect You From?. URL:  
<https://management.org/what-does-a-vpn-protect-you-from> (дата звернення:  
14.05.2025).

60. WireGuard. Conceptual Overview. URL:  
<https://www.wireguard.com/#conceptual-overview> (дата звернення: 14.05.2025).

61. OpenVPN. Access Server — take full control by installing OpenVPN on your server. URL: <https://openvpn.net/as-docs/general.html#access-server---take-full-control-by-installing-openvpn-on-your-server> (дата звернення: 14.05.2025).

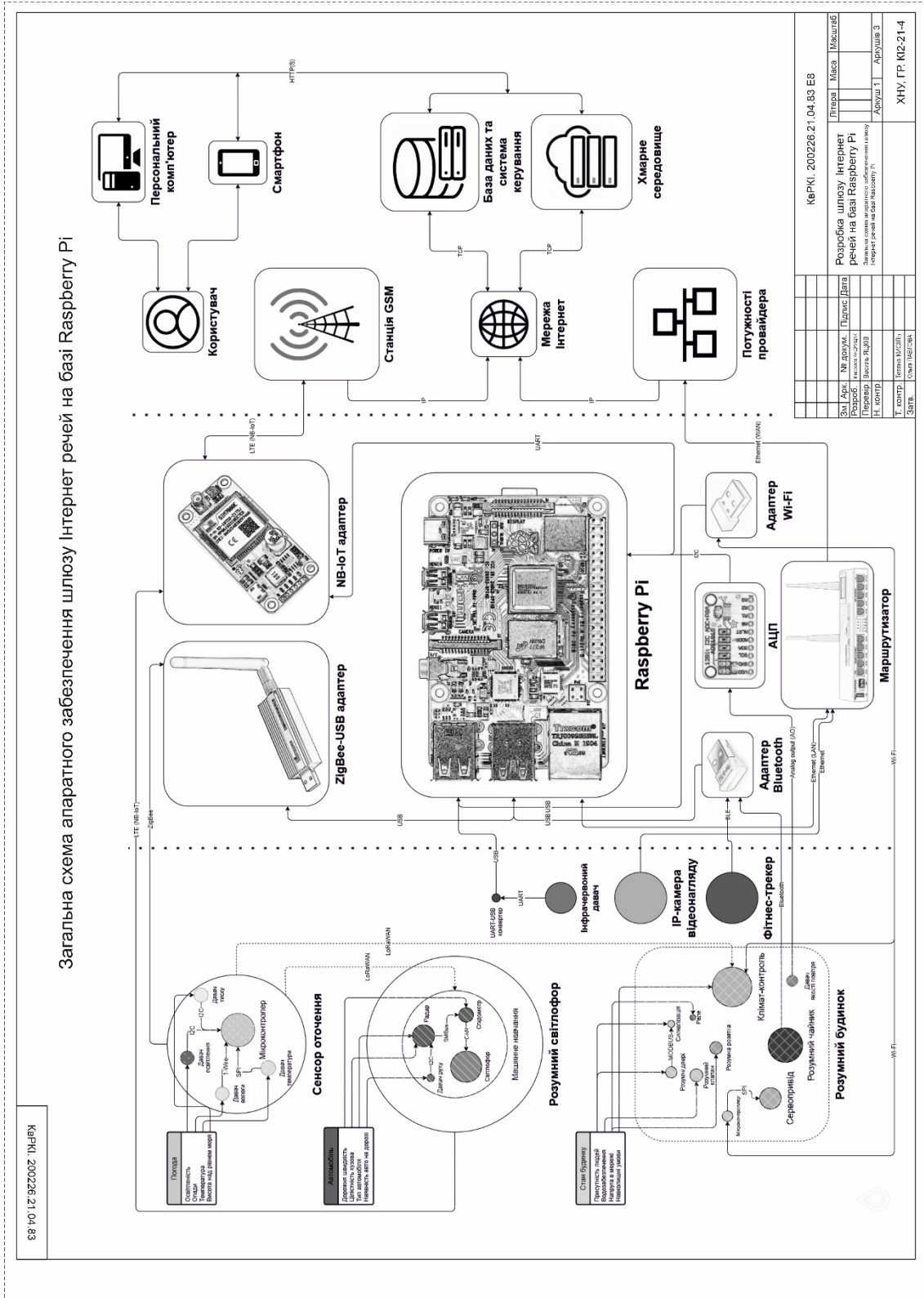
62. Raspberry Pi: OpenVPN vs WireGuard, which one is the best?. URL: <https://raspberrytips.com/openvpn-vs-wireguard/> (дата звернення: 14.05.2025).

63. Welcome to PiVPN Docs. URL: <https://docs.pivpn.io/> (дата звернення: 16.05.2025).

|      |      |         |        |      |                          |            |
|------|------|---------|--------|------|--------------------------|------------|
|      |      |         |        |      | КВРКІ 200226.21.04.83 ПЗ | Арк.<br>73 |
| Зм.. | Арк. | №докум. | Підпис | Дата |                          |            |

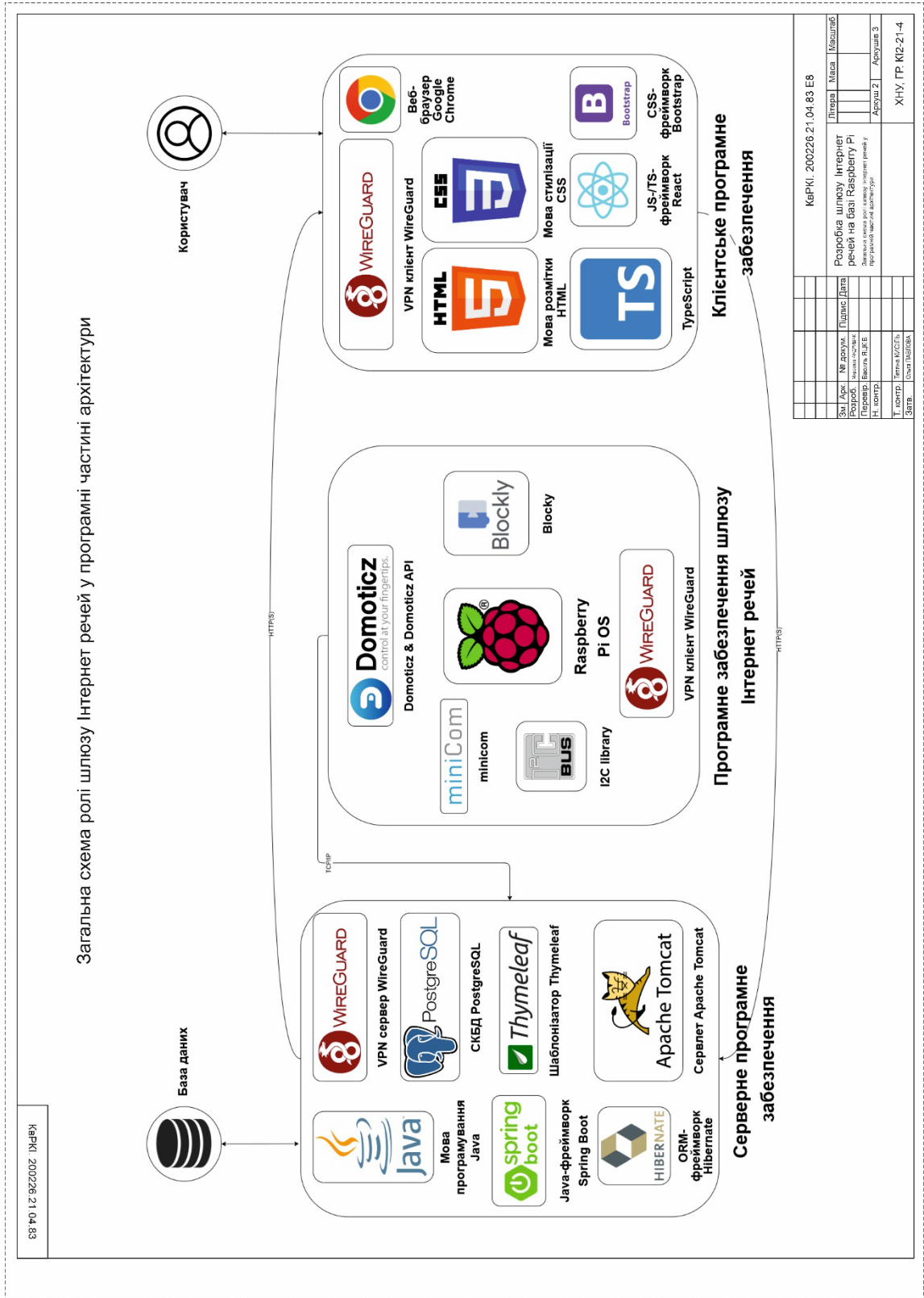
## Додаток А (обов'язковий)

# КОПІЯ КРЕСЛЕННЯ «ЗАГАЛЬНА СХЕМА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ ШЛЮЗУ ІНТЕРНЕТУ РЕЧЕЙ НА БАЗІ RASPBERRY PI»



**Додаток Б**  
(обов'язковий)

**КОПІЯ КРЕСЛЕННЯ «ЗАГАЛЬНА СХЕМА РОЛІ ШЛЮЗУ ІНТЕРНЕТУ  
РЕЧЕЙ У ПРОГРАМНІЙ ЧАСТИНІ АРХІТЕКТУРИ»**





## Додаток Г

### ЛІСТИНГ ПРОГРАМНИХ КОДІВ ДЛЯ ПРИСТРОЇВ

#### BMP280:

```
import board
import busio
import adafruit_bmp280
from time import sleep
# Create the BMP280 instance, passing in the I2C bus
i2c = busio.I2C(board.SCL, board.SDA)
# Create the BMP280 instance, passing in the I2C bus
bmp280 = adafruit_bmp280.Adafruit_BMP280_I2C(i2c, address=0x76)
# Calibration of height above sea level according to ambient pressure
bmp280.sea_level_pressure = 980.5
while True:
    print ("\nTemperature - %0.1f degrees of Celsius" % bmp280.temperature)
    print ("Pressure - %0.1f hPa" % bmp280.pressure)
    print ("Altitude - %0.1f meters" % bmp280.altitude)
    sleep(1)
```

#### TSL2561:

```
import board
import busio
import adafruit_tsl2561
import time
# Create the I2C bus
i2c = busio.I2C(board.SCL, board.SDA)
# Create the TSL2561 instance, passing in the I2C bus
tsl = adafruit_tsl2561.TSL2561(i2c)
# Print chip info
print("Chip ID = {}".format(tsl.chip_id))
print("Enabled = {}".format(tsl.enabled))
print("Gain = {}".format(tsl.gain))
print("Integration time = {}".format(tsl.integration_time))
print("Configuring TSL2561...")
# Enable the light sensor
# tsl.enabled = True
# time.sleep(1)
# Set gain 0=1x, 1=16x
tsl.gain = 0
# Set integration time (0=13.7ms, 1=101ms, 2=402ms, or 3>manual)
```

```

tsl.integration_time = 1
print("Getting readings...")
# Get raw (luminosity) readings individually
broadband = tsl.broadband
infrared = tsl.infrared
# Get raw (luminosity) readings using tuple unpacking
# broadband, infrared = tsl.luminosity
# Get computed lux value (tsl.lux can return None or a float)
lux = round(tsl.lux, 2)
# Print results
print("Enabled = {}".format(tsl.enabled))
print("Gain = {}".format(tsl.gain))
print("Integration time = {}".format(tsl.integration_time))
print("Broadband = {}".format(broadband))
print("Infrared = {}".format(infrared))

if lux is not None:
    print("Lux = {}".format(lux))
else:
    print("Lux value is None. Possible sensor underrange or overrange.")
# Disable the light sensor (to save power)
# tsl.enabled = False

```

## MQ-135:

```

import RPi.GPIO as GPIO
from time import sleep

GPIO.setmode(GPIO.BCM)
GPIO.setup(27, GPIO.IN)

try:
    while True:
        if GPIO.input(27):
            print("Gas is not detected")
        else:
            print("Gas is detected!")
        sleep(1)
except:
    print("Exiting...")
    GPIO.cleanup()

```

## SIM800C GSM/GPRS HAT:

```
import time
import subprocess
import serial
import asyncio
import RPi.GPIO as GPIO

# GPIO setup
GPIO.setmode(GPIO.BCM)
GPIO.setup(27, GPIO.IN)

# UART loading
serial_port = serial.Serial("/dev/ttyS0", 115200, timeout=1)
print("Serial port opened successfully. Initializing...")
time.sleep(5)

# Recipient`s phone number
phone_number = "+YOUR_PHONE_NUMBER"

# Set to base user profile
serial_port.write(b"ATZ\r")
response = serial_port.read(100)
time.sleep(0.5)
print(response.decode())

# Set messages to text format
serial_port.write(b"AT+CMGF=1\r")
response = serial_port.read(100)
time.sleep(0.5)
print(response.decode())

# Sending Raspberry Pi temperature at a SMS
async def sending_temperature_state():

    try:
        while True:
            await asyncio.sleep(0.5)

            # Read all stored messages
            serial_port.write(b'AT+CMGL="ALL"\r')
            data = serial_port.readall()
            time.sleep(1)
            data = data.decode("latin-1")
            CMGLs = data.split("+CMGL: ")
            print("\n- CMGLs number: " + str(len(CMGLs)))
            time.sleep(1)
            for cmgl in CMGLs:
                num = ""
```

```

cmglID = -1
for line in cmgl.splitlines():
    if len(line) > 0 and line != 'OK' and line != 'AT+CMGL="ALL"':
        print("\t" + line)
        parts = line.split(",")
        if len(parts) > 2:
            cmglID = parts[0].strip()
            parts[2] = parts[2].replace("'", "")
            num = parts[2]
            # Validate the requestors mobile number
        elif len(parts) == 1 and num == "+YOUR_PHONE_NUMBER":
            if parts[0] == "get_temperature":
                print("Received 'get_temperature' commandat CMGL -
" + cmglID)

                # Read the RPi temperature
                cmd = '/bin/vcgencmd measure_temp'
                process =
subprocess.Popen(cmd, stdout=subprocess.PIPE, shell=True)
                out, error = process.communicate()
                print(out.decode("latin-1"))
                # Sending the SMS
                serial_port.write(b'AT+CMGS="' + num.encode() +
b'"\'r')

                time.sleep(0.5)
                serial_port.write(out + b"\'r")
                time.sleep(0.5)
                serial_port.write(bytes([26])) # Ctrl + Z to send
SMS

                time.sleep(4)
                print("Message with temperature of Raspberry Pi
sent")

                response = serial_port.read(100)
                print(response.decode())

                # Delete the message so that it doesn't get
reprocessed

                serial_port.write(b"AT+CMGD=" +
cmglID.encode("utf-8"))

                response = serial_port.read(100)
                print(response.decode())

        finally:
            serial_port.close()

```

```

# Code for sensor MQ-135
async def mq135_gas_detection():

    try:
        while True:
            # If gas not detected
            if GPIO.input(27):
                print("\n--- Gas is not detected\n")
                await asyncio.sleep(15)
            # If gas detected
            else:
                gas_detection = "Gas is detected!"
                serial_port.write(b'AT+CMGS="' + phone_number.encode() + b'"' + b'\r')
                time.sleep(0.5)
                serial_port.write(gas_detection.encode() + b"\r")
                time.sleep(0.5)
                serial_port.write(bytes([26])) # Ctrl + Z to send SMS
                time.sleep(4)
                # Hiding the phone number in console
                def hiding_phone_number(phone_number):
                    if len(phone_number) > 9:
                        return phone_number[:9]
                    else:
                        return phone_number
                print("\n--- Gas is detected! Message sent to: " +
hiding_phone_number(phone_number) + "...")
                response = serial_port.read(100)
                print(response.decode())
                await asyncio.sleep(25)

        finally:
            GPIO.cleanup()
            print("Exiting...")
# General function
async def main():
    task_1 = asyncio.create_task(sending_temperature_state())
    task_2 = asyncio.create_task(mq135_gas_detection())
    await asyncio.gather(task_1, task_2)
asyncio.run(main())

```

## Додаток Д

### ЛІСТИНГ ПРОГРАМНИХ КОДІВ ДЛЯ ФАЙЛІВ КОНФІГУРАЦІЇ VPN

#### VPN-сервер:

```
[Interface]
Address = YOUR_SERVER_IP_ADDRESS/YOUR_SERVER_NET_SUBMASK
SaveConfig = true
PrivateKey = YOUR_SERVER_PRIVATE_KEY
ListenPort = YOUR_SERVER_PORT
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -A FORWARD -o %i -j ACCEPT;
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -D FORWARD -o %i -j ACCEPT;
iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE
[Peer]
PublicKey = YOUR_CLIENT_PUBLIC_KEY
AllowedIPs = YOUR_CLIENT_IP_ADDRESS/YOUR_CLIENT_NET_SUBMASK
```

#### VPN-клієнт:

```
[Interface]
Address = YOUR_CLIENT_IP_ADDRESS/YOUR_CLIENT_NET_SUBMASK
PrivateKey = YOUR_CLIENT_PRIVATE_KEY
DNS = 1.1.1.1

PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -A FORWARD -o %i -j ACCEPT;
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -D FORWARD -o %i -j ACCEPT;
iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

[Peer]
PublicKey = YOUR_SERVER_PUBLIC_KEY
Endpoint = YOUR_SERVER_IP_ADDRESS/YOUR_SERVER_PORT
```

## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** М.А. Андрущак

**Співавтор:**

**Назва:** Андрущак\_Мережевий шлюз Інтернету речей на базі Raspberry Pi

**Експерт:**

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 2.1%

**Коефіцієнт подібності 2:** 0.5%

**Мікропробіли:** 0

**Заміна букв:** 0

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2025-06-05 17:20:11.0

**Після аналізу Звіту подібності констатую наступне:**

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

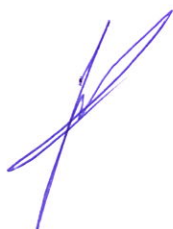
Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

**Обґрунтування:**

2025-06-05

Дата



Доцент Андрій Нічепорук

експерт

# Anti-Plagiarism v-15.257

**Максимальне співпадіння з одним документом 0.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. **Помилоч в документах: 17%**

|   |          |         |                                |         |
|---|----------|---------|--------------------------------|---------|
| ID: 131547<br>Назва: БКР Мережевий шлюз Інтернет<br>речей на базі Raspberry Pi<br>Додано в БД: 2024-06-19<br>Автора: М.А. Андрущак<br>Керівники: В.В. Яцків<br>Консультанти:<br>Опоненти: | Документ |         | Сумарний збіг по Базі<br>Даних |         |
|   | Символи  | Лексеми | Символи                        | Лексеми |
|   | 61542    | 522     | 439 (1%)                       | 9 (2%)  |

## Джерело плагіату

| ID | Опис | Наявність плагіату в документі |         |
|----|------|--------------------------------|---------|
|    |      | Символи                        | Лексеми |
|    |      |                                |         |

## РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Андрущак Мирослав Анатолійович

Тема: Мережевий шлюз Інтернету речей на базі Raspberry Pi

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   68  

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є реалізація мережевого шлюзу Інтернет речей на базі Raspberry Pi
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проаналізовано теоретичну інформацію про сферу Інтернет речей, роль та принципи роботи шлюзу Інтернет речей у архітектурі, виконано аналіз програмно-апаратних рішень шлюзу та аналіз платформи Raspberry Pi 4B у якості платформи для створення шлюзу Інтернет речей. В другому розділі кваліфікаційної роботи проведено аналіз функціональних вимог й основних програмно-апаратних елементів шлюзу та на їх основі проектування шлюзу Інтернет речей на базі Raspberry Pi. В третьому розділі кваліфікаційної роботи виконано програмно-апаратну реалізацію шлюзу Інтернет речей, а саме: забезпечено належне підключення апаратних компонентів до шлюзу за відповідними інтерфейсами і комунікація з ними за відповідними інтерфейсами й протоколами, забезпечено програмні компоненти для роботи девайсів у системі, забезпечено сконфігурованої платформи для збору даних і їх передачі до обчислювального середовища Domoticz та забезпечено VPN-клієнт для безпечного з'єднання із сервером для передачі даних.
4. Позитивні сторони роботи: детальне теоретичне поглиблення у сферу Інтернет речей та висока практична цінність роботи.

5. Негативні сторони роботи: забезпечення низької кількості інтерфейсів для роботи шлюзу.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректним чином, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.


8. Інші зауваження: немає

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)

Корсунська Л.О., к.т.н., доцент каф. АКІТ та Р, ХАУ

“ 16 ” червня 2025 р.

 (підпис)

Завідувачу кафедри КПС  
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Мирослава АНДРУЩАКА

---

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-4

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

16.06 2025 року

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ**  
**КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ**  
**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Мережевий шлюз Інтернет речей на базі Raspberry Pi

Автор: Андрушак Мирослав Анатолійович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Василь ЯЦКІВ, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок  | Позначка про відповідність |
|---|---|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.  | відповідає                 |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи |                            |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.        |                            |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.  |                            |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, тому що:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження й не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги являються широкоживаними фразами або виразами, про що свідчить посилання системи на збіг з 5-25 джерелами на один фрагмент речення;
- 4) в якості запозичень в окремих місцях системою зафіксовано програмний код з бібліотеки, котра являється збірником підпрограм для вирішення великої кількості задач та, відповідно, використовується широкою кількістю розробників і не може розглядатися як об'єкт авторських прав та їх порушення;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 2.01% згідно коефіцієнту подібності 1 та 0.5% згідно коефіцієнту подібності 2 і адресується до 8 першоджерел; та системою Anti-Plagiarism складає 0.0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



Василь ЯЦКІВ

Андрій Нічепорук

Ольга Павлова