МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних
технологій, обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Йосипчук Олексій Григорович

УДК 004.05:004.4

КВАЛІФІКАЦІЙНА РОБОТА

«Інформаційна система охорони розумного будинку на базі технологій IoT»

126 «Інформаційні системи та технології»

Подається на здобуття освітнього ступеня бакалавр

кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ О.Г. Йосипчук

Керівник роботи
Лапін Андрій Валерійович
кандидат економічних наук

Житомир – 2023

Висновок кафедри _____

за результатами попереднього захисту: _____

Протокол засідання кафедри

№ ____ від «____»_____ 2023 р.

Завідувач кафедри
доктор економічних наук,
професор _____ Николюк Ольга Миколаївна

«____» _____ 2023 р.

Результат захисту кваліфікаційної роботи

Здобувач вищої освіти Йосипчук Олексій Григорович захистив кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

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

АНОТАЦІЯ

Йосипчук О.Г. Інформаційна система охорони розумного будинку на базі технологій IoT. – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра за спеціальністю 126 – Інформаційні системи та технології. – Поліський національний університет, Житомир, 2023.

Кваліфікаційна робота на тему "Інформаційна система охорони розумного будинку на базі технологій IoT" присвячена створенню автоматизованої системи охорони розумного будинку. У роботі було проведено аналіз сучасних розумних систем та систем охорони будинків, визначено вимоги до системи охорони розумного будинку та розроблено її архітектуру. У результаті проведених досліджень було розроблено систему охорони, яка включає в себе як стандартні компоненти систем охорони, так і удосконалене ДБЖ, що значно розшить можливості для автоматизації різних систем під управлінням платформи Home Assistant, яка має відкритий код та створена для керування та автоматизації пристроїв IoT різних виробників.

Ключові слова: ДАТЧИК, ІНТЕРНЕТ РЕЧЕЙ, ІНФОРМАЦІЙНА СИСТЕМА, ОХОРОНА БУДИНКУ, РОЗУМНИЙ БУДИНОК, СИГНАЛІЗАЦІЯ.

SUMMARY

Yosypchuk O.H. Smart home security information system based on IoT technologies. - Qualification work as a manuscript.

Qualification work for obtaining a Bachelor's degree in specialty 126 - Information Systems and Technologies. - Polissia National University, Zhytomyr, 2023.

The qualification work on the topic "Information security system of a smart home based on IoT technologies" is dedicated to the development of an automated security system for a smart home. The work includes an analysis of modern smart systems and home security systems, determination of the requirements for the security system of a smart home, and the development of its architecture. As a result of the research conducted, a security system was developed, which includes both standard security system

components and an improved UPS that significantly expands the possibilities for automation of various systems under the control of the Home Assistant platform, which has an open-source code and is designed to manage and automate IoT devices from different manufacturers.

Keywords: ALARM SYSTEM, HOME SECURITY, INFORMATION SYSTEM INTERNET OF THINGS, SENSOR, SMART HOME

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IoT – Internet of Things (Інтернет речей)

ДБЖ – джерело безперебійного живлення

АЦП – аналого-цифровий перетворювач

ПК – персональний комп’ютер

ОС – операційна система

ВСТУП

В останні роки розвиток технологій Інтернету речей (IoT) виявився однією з ключових тенденцій в галузі інформаційних технологій. Застосування IoT стало незамінним для розвитку систем «розумних» будинків, що дає змогу забезпечити комфортні і безпечні умови проживання.

Система охорони розумного будинку на базі технологій IoT має на меті забезпечити безпеку та контроль за всіма процесами в будинку. Це можливо завдяки використанню датчиків руху, відеокамер, датчиків диму та води, а також виконавчих пристроїв, які підключені до мережі Інтернет. Інформація, отримана від датчиків, аналізується спеціальним програмним забезпеченням, яке визначає, які дії потрібно здійснити у випадку аварії чи небезпеки.

Одним з головних переваг системи охорони розумного будинку є можливість віддаленого керування всіма її функціями з використанням мобільних пристроїв чи комп’ютерів. Крім того, система може підключатися до інших систем «розумних» будинків, що дозволяє об’єднувати їх і створювати єдину мережу, що підтримується за допомогою IoT.

Метою кваліфікаційної роботи є розробка автоматизованої системи охорони, яка забезпечуватиме безпеку майна та буде спроможна масштабуватися, а також знизить ризики виникнення аварійних ситуацій.

Для досягнення мети проаналізовано потреби та вимоги користувачів, ринок та конкурентів в галузі розумних будинків, існуючі розумні систем та системи охорони для визначення найбільш ефективних рішень, розроблена архітектура системи охорони розумного будинку на базі технологій IoT, визначені необхідні компоненти для реалізації цього проекту, та, для забезпечення ефективної взаємодії з користувачем, створено інтерфейс системи, за допомогою якого, реалізована можливість віддаленого керування всіма її функціями з використанням мобільних пристроїв та комп’ютерів.

Предметом дослідження є інформаційна система охорони розумного будинку на базі технологій IoT, яке спрямоване на розробку та впровадження системи, яка

сприятиме безпеці та комфорту життя мешканців розумного будинку. Об'єкт дослідження – технології IoT, які використовуються для створення цієї системи, а також розроблені програмні та апаратні засоби системи.

Для реалізації досліджень використані метод аналізу літературних джерел, завдяки якому можна використовувати наукову та технічну літературу для збору інформації про технології IoT, метод експериментальних досліджень, що включає в себе тестування обладнання, програмного забезпечення та аналіз його роботи в різних умовах, метод моделювання та імітації за допомогою комп'ютерних програм з імітації роботи різних автоматизованих систем для тестування різних сценаріїв та ситуацій.

Під час роботи над розробкою системи охорони було опубліковано дві статті на теми «Побудова математичної моделі системи охорони розумного будинку на базі технологій IoT», в якій описується математична модель, що може визначити імовірність безвідмовної роботи системи безпеки та повідомити про це відповідних фахівців для запобігання можливим проблемам, та «Розумне джерело безперебійного живлення як гарант надійності системи розумного будинку», в якій описується розробка, за допомогою якої, можна отримувати важливу інформацію з ДБЖ, наприклад рівень заряду батареї, що значно розширить можливості автоматизації розумного будинку.

Розробка та впровадження системи охорони розумного будинку на базі технологій IoT дозволить забезпечити безпеку в будинку та на його території, об’єднати багато пристроїв в єдину систему для автоматизації багатьох процесів та виключення людини з цього ланцюга, здійснювати віддалений моніторинг та управління будинком.

Кваліфікаційна робота «Інформаційна система охорони розумного будинку на базі технологій IoT» має 42 сторінку, 17 рисунків, 4 лістинга коду, 7 додатків, 28 джерел.

Розділ 1. СИСТЕМНИЙ АНАЛІЗ ОСОБЛИВОСТЕЙ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОХОРОНИ БУДИНКУ НА БАЗІ ТЕХНОЛОГІЙ IOT

1.1 Аналіз інформаційних потреб і визначення моделі інформаційної системи

Аналіз інформаційних потреб є важливим етапом в розробці та вдосконаленні системи охорони будинку. Знання потреб користувачів в інформації допомагає розробити систему, яка буде ефективною, зручною та надійною, допоможе створити систему, яка задовольнить потреби користувачів та буде ефективною в захисті їхньої безпеки.

Головна задача системи охорони полягає в забезпеченні безпеки та захисту людей та майна від ризику вторгнення, крадіжки, пожежі, а також інших небезпечних ситуацій. Для досягнення цієї мети системи охорони мають бути оснащені різноманітними технічними засобами, які дозволяють відслідковувати події, сповіщати про їх виникнення та приймати ефективні заходи для їх запобігання або усунення.

Сьогодні сучасний ринок систем безпеки пропонує різні типи пристроїв. Це може бути обладнання з широким функціоналом або вузькою спеціалізацією. Системи, що реагують на різні фактори ризику або подають різні сигнали лиха. Створені для відлякування правопорушників або для відправлення повідомлення до центру управління.

Відповідно до того, як системою розцінюється небезпека, їх можна поділити на:

- охоронна сигналізація від злому (що реагує на перетинання периметру, рух, удари, розбиття вікон);
- протипожежна система (що повідомляє про задимлення, різкий підйом температури, захист від чадного газу);
- захист від затоплення (що фіксує виникнення протікань у системах опалення та водопостачання);
- універсальна (має на увазі реагування на всі небезпечні ситуації) [17].

Охоронна сигналізація – інформаційна система, яка дозволить завжди бути упевненим у безпеці свого будинку, офісу, квартири, складу, виробничого приміщення тощо. Вона розрахована на попередження несанкціонованого доступу до приміщення.

Основні складові сучасної охоронної сигналізації будинку:

Датчик - це пристрій, що сприймає зовнішні впливи і реагує на них зміною електричних сигналів та виробляє вихідний сигнал, зручний для дистанційного передавання, зберігання та використання у системах керування і має нормовані характеристики [16]. Датчики, які використовують в сучасній охоронній сигналізації:

Датчик руху – механічний або електронний прилад, призначений для виявлення фізичних рухів у приміщенні або на певній території.

Датчик вібрації – прилад, який завдяки вбудованому високочутливому акселерометру, здатен визначати порушення цілісності конструкцій, що охороняється, використовується для відстеження зовнішніх вібрацій та руху як додатковий елемент безпеки розумного будинку.

Датчиків відчинення – невід’ємний елемент системи охоронної сигналізації, який використовується з метою виявлення спроби несанкціонованого відкриття дверей або вікон під час спроби проникнення в квартиру або домоволодіння. Пристрій складається з двох частин: магніт і герконовий датчик. Перший елемент встановлюється на рухомій частині конструкції, а другий — монтується на дверній або віконній рами.

Виконуючі пристрої – це пристрої, здатні перетворювати цифрові електричні сигнали, що надходять від приладів обробки даних, в дії. Наприклад, для того, щоб при спрацюванні датчика відкриття вхідних дверей та вимкнутої охорони вмикалося світло.

Шлюз – ключовий елемент мережевого обладнання IoT, який забезпечує зв'язок між усіма елементами, що входять до системи. Численні датчики безпеки, контролю доступу, що працюють від батарейок, а також виконуючі пристрої, які використовують Zigbee, Z-Wave або Bluetooth LE, вимагатимуть наявність шлюзу

з підтримкою свого протоколу зв'язку для інтеграції в розумний будинок та управління ними.

Мережеве середовище (середовище зв'язку) – це модель, яка описує як повинні взаємодіяти мережеві протоколи і обладнання. Існує три основних типи носіїв, які використовуються пристроями IoT для комунікації:
мідні носії, волоконно-оптичні кабелі, та найпоширенішим способом комунікації всередині системи розумний будинок є бездротовий зв'язок.
Найбільш поширені бездротові технології, що використовуються для зв'язку між компонентами розумного будинку є Bluetooth LE, Zigbee та Z-Wave. В порівнянні Bluetooth LE з Zigbee та Z-Wave, перший досить рідко використовується у побудові бездротової мережі в системах охорони через малу дальність дії. Натомість Zigbee та Z-Wave досить поширені, та мають досить схожі характеристики окрім радіусу дії – 30 та 100 метрів відповідно.
Радіус дії ZigBee є достатнім для більшості квартир, за необхідності може бути значно розширений за допомогою розширювачів мережі, має більш широкий асортимент пристроїв з меншою вартістю, що дозволяє створювати як бюджетні рішення, так і преміум сегменту.

Контролер (сервер обробки даних) – прилад (хмарний сервер), який аналізує інформацію, що поступила від охоронних датчиків та виконує заздалегідь запрограмовані в ній функції, що виконуються на підставі зібраної інформації.

Обробку даних, що надходять від пристроїв системи охорони будинку можна розділити на дві моделі:

- Модель хмарних обчислень – це сервіс, що пропонує доступ на вимогу до спільного пулу налаштованих обчислювальних ресурсів. Часто розгорнуті поза приміщеннями, ці ресурси можуть бути швидко доступним, задіяючи при цьому мінімальні зусилля управління.
- Модель туманних обчислень – це модель обчислень, що ідентифікує розподілену обчислювальну інфраструктуру ближче до мережевої периферії. Це дозволяє периферійним пристроям запускати програми локально та приймати рішення невідкладно. Це, в свою чергу, зменшує навантаження

даними в мережах, оскільки необроблені дані не потрібно надсилати через мережеві з'єднання. Це підвищує стійкість, дозволяючи пристроям IoT працювати, навіть коли втрачається підключення до Інтернету. Також підвищується безпека, запобігаючи перенесенню конфіденційних даних за межі системи.

1.2 Функціональне моделювання інформаційної системи

Функціональне моделювання системи дозволяє розробити оптимальну конфігурацію системи задля підвищення її ефективності, визначення основних функцій, які повинні виконуватися системою охорони, а також з'ясувати, як вони повинні взаємодіяти між собою.

Для розробки функціональної моделі системи скористаємось нотацією IDEF0, яка використовується для опису функцій і процесів у системі та відображає функції та їх взаємозв'язки. Основною особливістю IDEF0 проектування є ієрархічне представлення об'єктів, що значно полегшує розуміння предметної області. На рис. А.1 – це перший рисунок Додатку А, зображена діаграма нульового рівня системи охорони будинку.

Центральний блок на рисунку позначає не деталізований головний процес – охорона будинку. Стрілками зліва описується вхідна інформація, що буде використовуватися або перетворюватися певними процесами для отримання результату. Стрілки зверху – це перелік правил, інструкцій або стратегій , якими керується робота. Стрілки знизу відображають усі ресурси, які задіяні у цій системі, за допомогою яких буде виконана певна робота. Стрілки праворуч – це результат роботи системи, яким може бути виконавча дія або інформація, що виробляється.

З діаграми на рис. А.1 – це перший рисунок Додатку А, можна прийти до висновку, що система охорони будинку складається з автоматизованого збору та обробки інформації за допомогою мережі та серверу, результатом якої є інформація з приладів та, за необхідності, сигналізація тривоги.

Для здійснення подальшого аналізу системи та виявлення можливих проблем, необхідно розбити складну систему на більш прості компоненти, що

допоможе проаналізувати їх функції та залежності. Для цього створюється декомпозиція IDEF0-моделі, яка є корисним інструментом для аналізу та покращення системи охорони будинку

На рис. А.2 – це другий рисунок Додатку А, зображена декомпозиція нульового рівня інформаційної системи на якій зображена чітка послідовність дій. В першому блоці відбувається збір даних, які передаються в блок 2 для аналізу. Результатом роботи 2 блоку є два результати, перше – це формування повідомлення для оповіщення, друге – це повернення збору даних, якщо не було змін в показниках датчиків. Блок 3 відповідає за повідомлення та сигнал тривоги.

На рис. А.3 – це третій рисунок Додатку А, зображена декомпозиція 2 блоку першого рівня на якій відображається послідовність дій, яка використовується для аналізу даних. На діаграмі зазначено, що перевірка даних датчиків проходить одна за одною, результатом перевірки є інформація про відсутність змін або повідомлення (масив даних) з інформацією про стан датчиків.

На рис. А.4 – це четвертий рисунок Додатку А, уточняється як саме працює блок 3 «Оповіщення». Після отримання їм повідомлення (масив даних) з інформацією про стан датчиків, в блоку 1 відбувається перевірка стану сигналізації на предмет активації. Якщо система активована, виконуються блоки 3 та 4 де формується тривожне повідомлення у додаток (наприклад на смартфоні) та вмикається гучномовець для подачі сигналу тривоги. Якщо система не активована виконується блок 2, який формує повідомлення для додатку з інформацією про стан датчиків.

Висновки до Розділу 1

Провівши аналіз інформаційних потреб було визначено модель системи охорони розумного будинку. Моделлю обробки даних має бути саме туманні обчислення, що має власну інфраструктуру, яка ближче до мережевої периферії. За рахунок цього підвищується надійність, дозволяючи пристроям працювати автономно, навіть коли втрачається підключення до Інтернету. Бездротовою технологією, що використовуються для зв'язку між компонентами розумного

будинку найкраще обрати ZigBee, радіус дії якої є не найбільшим, але достатнім для більшості квартир, за необхідності може бути значно розширений за допомогою розширювачів мережі, має більш широкий асортимент пристроїв з меншою вартістю.

За допомогою функціонального моделювання розроблено оптимальну конфігурацію системи, яку можна поступово вибудовувати та аналізувати задовго до її втілення. Використання декомпозиції моделі значно полегшує розуміння предметної області.

Розділ 2 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОХОРОНИ РОЗУМНОГО БУДИНКУ НА БАЗІ ТЕХНОЛОГІЙ IOT

2.1 Структура інформаційної системи

Для опису структури інформаційної системи охорони може бути використана методологія побудови діаграм IDEF3, яка орієнтована на процеси та потоки даних та призначена для опису логіки взаємодії інформаційних потоків. Стандарт IDEF3, використовує графічний опис інформаційних потоків, взаємин між процесами обробки інформації і об'єктів, що є частиною цих процесів [15].

На рис. Б.1 – це перший рисунок Додатку Б, зображена діаграма в нотації IDEF3 системи охорони будинку – процесу, який виділяє послідовність дій системи. Прямокутником на діаграмі позначаються дії, що в IDEF3 називається одиницею роботи. Кожній дії призначається унікальний номер. Стрілками на діаграмі виділяють істотні взаємовідносини між діями. Однією з особливостей даної методології є перехрестя які є інструментом для побудови логіки динамічних процесів.

З'єднання «Ексклюзивне АБО» J8 на діаграмі об'єднує в собі три дії – початок процесу та два повернення в ході роботи процесу, згідного логіки «Ексклюзивне АБО», тільки одна вихідна дія повинна завершитися, тобто на J1 потрапить тільки щось одне. Наступне перехрестя J1 «Логічне І» означає, що всі чотири дії в блоках 1-4 будуть ініційовані. Перехрестя J2 «Логічне І» описує обов'язкове завершення кожної дії. Після збору інформації з датчиків переходимо до послідовності: консолідація інформації, її передачі на сервер та подальшого аналізу. Перехрестя J3 означає, що після аналізу даних має виконатись лише одна дія, або блок 8, або, згідно J4, блок 9 та повернення на початок. Перехрестя J7, J5 та J6 описують подальші дії як повернення на початок та, на підставі вибраного алгоритму, виконується тільки блок 9, або блоки 9 та 10 разом.

На прикладі декомпозиції блоку 7, що зображено на рис. Б.2 – це другий рисунок Додатку Б, можна зрозуміти як саме відбувається аналіз.

Сформована інформація у вигляді масиву даних передається на блок 7, в блоках 12-14 відбувається її перевірка на предмет її присутності та цілісності після чого відбувається вибір послідовності, заздалегідь запрограмованих, дій, відповідно до отриманих даних. За допомогою розгалужувача J11 ми показуємо такий вибір. Виконуються дії описані в блоку 16 або в 17 та 18 разом. На виході маємо дані для сповіщення та опис алгоритму або інформацію про помилку.

2.2 Технічне забезпечення інформаційної системи охорони розумного будинку на базі технологій IoT

У параграфі 1.1. було розглянуто структуру системи охорони будинку і з’ясовано, що вона складається з багатьох складових – датчиків, виконуючих пристроїв, шлюзу, контролеру, мережевого середовища. Для об'єднання пристроїв був вибраний протокол Zigbee з двох причин: по-перше він бездротовий, а по-друге він досить енергоефективний, Zigbee-датчик працювати від батареї типу CR кілька років, що не потребує зовнішнє живлення. Для реалізації свого проекту будуть використовуватися наступні засоби технічного забезпечення:

- датчик відкриття дверей MiJia (MCCGQ01LM), зображений на рис. В.1 – це перший рисунок Додатку В, який складається з двох частин і працює за принципом чутливості до зміни магнітного поля одна частина датчика містить геркон, а друга - постійний магніт;
- датчик руху Sonoff SNZB-03, зображений на рис. В.2 – це другий рисунок Додатку В, який можна розмістити в будь-якому місці будинку для виявлення руху;
- датчик вібрації Aqara (DJT11LM), зображений на рис. В.3 – це третій рисунок Додатку В, який є багатофункціональний чутливий датчик для системи охорони будинку, який фіксує нахили, що можна використовувати для контролю за відкриттями вікон, а також падіння та вібрацію, що можна використати для ідентифікації розбиття вікон;
- шлюз Sonoff ZBDongle-E Zigbee USB, зображений на рис. В.4 – це четвертий рисунок Додатку В, який може замінити стандартні шлюзи та може

створювати мережі із Zigbee-пристроїв без прив'язки до якогось конкретного виробника. Його можна використовувати як шлюз у платформах з відкритим кодом системи розумний дім, наприклад Home Assistant. Інтегрується за допомогою zigbee2mqtt та дозволяє контролювати більше сотні пристроїв в одній системі;

- міні комп’ютер NiPoGi GK3V, зображений на рис. В.5 – це п’ятий рисунок Додатку В, що в якості сервера дасть нам можливість використовувати платформи з відкритим кодом системи розумний дім, таких як, наприклад Home Assistant, яка є універсальною відкритою операційною системою для керування та автоматизації пристроїв IoT різних виробників. Це локальне рішення, що не залежить від хмари чи наявності інтернету, та підтримує велику кількість протоколів;
- сирена Tervix Pro Line ZigBee Siren, зображена на рис. В.6 – це шостий рисунок Додатку В, що працює від 12В, та має вбудований акумулятор, який дозволяє працювати автономно до 24 годин;
- реле Tuya ZigBee Relay Module, зображене на рис. В.7 – це сьомий рисунок Додатку В, працює від 7-32В, має «сухий контакт», який буде використаний для автоматичного відкриття та закриття електромеханічного замка.
- маршрутизатор Xiaomi Mini, зображений на рис. В.8 – це восьмий рисунок Додатку В, який забезпечить доступу до глобальної мережі інтернет та є досить продуктивним дводіапазонним роутером, за допомогою якого можна легко налаштувати бездротову мережу практично для будь-яких потреб.

Важливим етапом при розробці та впровадженні системи охорони є будування структури комп’ютерної мережі системи охорони. За допомогою цієї структури стає можливим заздалегідь спланувати розміщення пристроїв, їх кількість та взаємодію між ними. На рис. Б.3 – це третій рисунок Додатка Б, зображено структуру комп’ютерної мережі системи охорони.

2.3 Підвищення надійності інформаційної

У системах розумного будинку, безперервне живлення є необхідною умовою для забезпечення надійності та стабільності функціонування всіх підсистем та пристроїв, що входять до складу системи. Безперервне живлення є критичною умовою роботи системи у будь-який час, навіть у випадку виникнення аварійних ситуацій у мережі електропостачання. Актуальність ДБЖ в системі розумного будинку пояснюється наявністю великої кількості електричних пристроїв, що працюють в мережі, та їх значної чутливості до змін напруги та струму. Крім того, ДБЖ забезпечує збереження інформації та налаштувань системи у разі відключення електропостачання. Це важливо для забезпечення швидкого відновлення роботи системи після відновлення живлення.

Можливість контролювати заряд батареї та джерело живлення з будь-якої точки світу, можливість автоматичного включення сервера обробки даних, наприклад, розгорнутих на міні ПК, після їх повного виключення – все це стане в нагоді усім власникам систем охорони будинку, максимально усуне з цього процесу людину та значно знизить ризики, пов’язані з перебоями та відключенням електроенергії.

Ми пропонуємо розробку контролеру живлення за допомогою плати NodeMCU на базі мікроконтролеру ESP8266, який має АЦП, пін A0, за допомогою якого ми зможемо напругу на батареї перетворити на зручну для обробки та передачі інформацію, а наявність цифрових сигналів стане нам в нагоді при отриманні інформації з ДБЖ про наявність зовнішнього живлення.

Для NodeMCU на базі мікроконтролера ESP8266 напруга логічної одиниці (HIGH) на вході або виході GPIO (General Purpose Input Output) становить 3,3 В, а батарея, що використовується в проекті – це свинцево-кислотний акумулятор, який має напругу 12 В. Для заміру напруги батареї нам знадобиться дільник напруги, який перетворить робочі напруги батареї 11 В – 12,6 В в доступні для обробки, наприклад зменшити в 5 разів. На рис. 1 приведена схема дільника напруги.

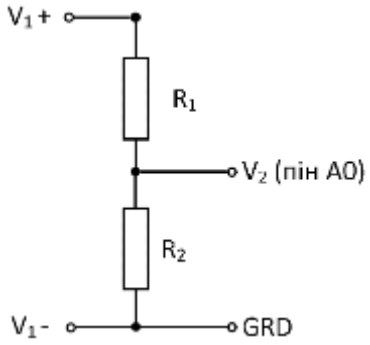

Рисунок 1 – Схема дільника напруги

NodeMCU в своїй схемі вже має резистор 320 кОм між A0 та GRD, максимальна напруга на акумуляторі – це напруга для заряджання акумуляторів, яка складає 14,5 В, знайдемо R_1 за формулою:

$$R_1 = R_2 \cdot \frac{V_1 - V_2}{V_2},$$

де R_2 – резистор, номіналом 320 кОм; V_1 – напруга на акумуляторі; V_2 – бажана напруга до 3,3 В після дільника.

У зв’язку з тим, що АЦП мікроконтролеру ESP8266 має розрядність 10 біт, зчитуючи інформацію з A0 ми будемо отримувати числа від 0 до 1023, які відповідають напрузі на акумуляторі, наприклад якщо застосувати R_1 = 1,086 МОм, відповідно число 1023, отримане з A0, буде відповідати 14,5 В на акумуляторі. Знаючи інформацію піна A0, за допомогою формули нижче, розраховуємо напругу на акумуляторі:

$$V_1 = V_0 \cdot \frac{C_1}{C_0} \cdot \frac{R_1 + R_2}{R_2},$$

де V_0 – напруга логічної одиниці (HIGH), яка становить 3,3 В; C_0 – розрядність АЦП, яка складає 1024 (10 біт); C_1 – показник з піна A0; R_1 та R_2 – резистори, які використовуються в дільнику напруги.

Знаючи напругу на акумуляторі, технічні характеристики свинцево-кислотних акумуляторів, відповідно до яких напруга на акумуляторі в 11 В відповідає повному розряду, а 13,8 В – повний заряд, за формулою розраховуємо заряд батареї у відсотках:

$$S = \frac{V_{ak} - V_{min}}{V_{max} - V_{min}} \cdot 100\%,$$

де V_{ak} – напруга на акумуляторі; V_{min} – напруга, яка відповідає заряду 0%; V_{max} – напруга, яка відповідає 100% заряду батареї.

Будь-яке ДБЖ обладнане світловим індикатором роботи від мережі у вигляді світлодіода, який живиться від 3 В. Під'їдавши один з цифрових пінів мікроконтролера до цього індикатора, ми отримаємо інформацію у вигляді true, що відповідає роботі від мережі, або false – робота від батареї.

Для реалізації автоматизованої роботи мікроконтролера написано відповідний код на C++, представлений у Додатку Г, завдяки якому він зможе міряти заряд батареї, отримувати інформацію про роботу від мережі чи батареї та передавати ці дані по протоколу HTTP, що забезпечить широкі можливості для інтегрування у різні системи розумного будинку.

За допомогою отриманих даних від мікроконтролеру, можна значно розширити автоматизацію розумного будинку, наприклад при наближенні до критично низького заряду при роботі від батареї правильно вимикати сервер обробки даних, що зменшить ризик втрати даних та псування пристрою, а при відновлені електропостачання надсилати команду на вмикання серверу. Інформація для платформи буде представлено у вигляді даних з сенсору sensor.mainspower – відсоток заряду батареї з атрибутами adc, що відображає дані с піна A0 та mainspower – індикатор роботи від мережі, дані з піна D1.

2.4 Алгоритми функціонування інформаційної системи

Сучасні технології дозволяють автоматизувати процес забезпечення безпеки за допомогою системи управління будинком - Home Assistant. Це програмне забезпечення для керування розумним будинком, яке дозволяє об'єднати різні пристрої та сервіси у єдину систему автоматизації. Для цього використовуються різноманітні функціональні блоки та алгоритми, які забезпечують роботу системи охорони будинку в режимі 24/7. В даному контексті, розглянемо основні алгоритми функціонування системи охорони.

Невід’ємним компонентом будь-якої системи охорони є панель керування сигналізацією. В даній системі цей компонент представлено у вигляді об’єкту, який може приймати різні стани, які відображають поточний стан сигналізації, наприклад «ввімкнена» або «вимкнена». На підставі цих даних створено алгоритм, який при постановці будинку на охорону подає команду на закриття електромеханічних замків, а при знятті з охорони відповідно відкриває їх, відключає сирену, якщо вона була ввімкнена та сповіщає про свої дії в месенджер. Лістинг коду мовою YAML, що використовується для опису автоматизацій в Home Assistant, представлено в Додатку Д.

Для постійного контролю за станами датчиків створено алгоритм, який при їх спрацьовуванні сповіщає про це в месенджер, описуючи, який саме датчик спрацював, та, для більшої зручності, за необхідності користувач мав можливість відключити ці повідомлення. Для реалізації цієї можливості було створено новий об’єкт input_boolean.triger_message_telegran, який приймає стани «on» або «off», які відображають необхідність в повідомленнях. Лістинг коду представлено в Додатку Е. Аналогічний код, що забезпечить відправку повідомлення про тривогу та увімкне сирену, буде код відправки станів датчиків, але для перевірки необхідності в цих діях буде об’єкт alarm_control_panel.home_alarm, який має стан відповідно до статусу сигналізації, а для активації тривоги буде використано сервіс switch.turn_on для об’єкту switch.tervix_pro_line_zigbee_siren, що увімкне сирену.

Важливим додатком в автоматизацію системи стане можливість своєчасного відключення сервера обробки даних при розряді батареї ДБЖ до

критичного рівня при відсутності мережевого джерела, що зменшить ризики виходу з ладу дорогого обладнання та сповістить власника про відсутність захисту будинку. Для реалізації цієї автоматизації необхідно моніторити стан sensor.mainspower, який отримує інформацію з ДБЖ про заряд батареї, та його атрибут mainspower, який приймає значення true або false, що відповідає наявності напруги в електромережі. При значенні заряду батареї менше 20% та відсутності напруги в електромережі застосовуємо вбудовані сервіси homeassistant.turn_off та shell_command.turn_off_computer, що забезпечать безпечне вимикання серверу, а notify.telegram_id повідомить про це власника, відповідний код представлено в Додатку Є.

Висновки до Розділу 2

В цьому розділі було визначено інформаційну інфраструктуру системи, яка складається з різних пристроїв для збору, обробки, передачі інформації та зв’язок між ними. Проаналізувавши ринок IoT пристроїв було сформовано вичерпний список необхідних компонентів для створення інформаційної системи охорони. Було реалізовано можливість підвищення надійності систем на базі технологій IoT, завдяки якій ДБЖ стало розумним пристроєм, яке може міряти заряд батареї, розуміє яке джерело живлення використовується, передавати цю інформацію на сервер та після поновлення живлення вмикати його. Для автоматичної роботи системи було описано алгоритми її дій, написано відповідний код автоматизації.

Всі коди, що описують алгоритми автоматизації прописуються у файлі automations.yaml, для редагування коду можна використовувати як сторонній редактор так і вбудований.

Розділ 3 ІНТЕРФЕЙС ІНФОРМАЦІЙНОЇ СИСТЕМИ ОХОРОНИ РОЗУМНОГО БУДИНКУ НА БАЗІ ТЕХНОЛОГІЙ IOT

3.1 Встановлення та налаштування сервера обробки даних

Для початку роботи с операційною системою Home Assistant необхідно обрати платформу на якій вона буде розвернута. Вичерпний список платформ, що підтримуються представлено на сайті https://www.home-assistant.io/installation/. Для розгортання сервера на платформі «Міні комп'ютер NiPoGi GK3V», яка була обрана в Розіділі 2.2, потрібно налаштувати BIOS ПК на використання режиму завантаження UEFI та вимкнено Secure Boot, а потім записати образ диска ОС Home Assistant на «завантажувальний носій», з якого завантажуватиметься ПК під час запуску Home Assistant, наприклад жорсткий диск.

Для початку необхідно створити «живу операційну систему» на USB-пристрої, наприклад Ubuntu, інструкція за посиланням https://ubuntu.com/tutorials/try-ubuntu-before-you-install#1-getting-started, та завантажити поточну операційну систему. Наступним кроком буде завантаження додатка Balena Etcher за посиланням https://www.balena.io/etcher та його запуск. Після закуску в першому кроці обираємо метод встановлення Flash from URL та вказуємо посилання на образ https://github.com/home-assistant/operating-system/releases/download/10.1/haos_generic-x86-64-10.1.img.xz. Коли Balena Etcher завантажить образ, вибираємо місце розміщення ОС та натискаємо Flash. Після закінчення інсталяції вимикаємо ПК, виймаємо USB пристрій з «живою» ОС та вмикаємо ПК [24].

Після загрузки ОС Home Assistant вона буде доступна у внутрішній мережі за посиланням http://ip_ha:8123, який буде відображено на екрані. Наступним кроком буде перший запуск, налаштування та встановлення необхідних інтеграцій таких як HACS та zigbee2mqtt, що значно розширить можливості з використання різних пристроїв. Для організації доступу із будь-якої точки світу використаємо статичний IP та відповідні налаштування маршрутизатору, наприклад при посиланні на ip_router:8000 отримаємо доступ до ip_ha:8123. Якщо не можливо

використовувати статичний IP можна скористатися сервісом, наприклад https://www.noip.com.

3.2 Інтерфейс інформаційної системи

Для створення інтерфейсу системи було використано вбудовані можливості платформи Home Assistant та мова YAML, близька до мов розмітки, яка орієнтований на зручність введення-виведення типових структур даних багатьох мов програмування [14].

Інтерфейс повністю адаптовано під WEB-браузер, смартфон та планшет. Додатки платформи присутні в Google Play та App Store. На рис. 3.1 зображено перший запуск додатка на iPad OS.

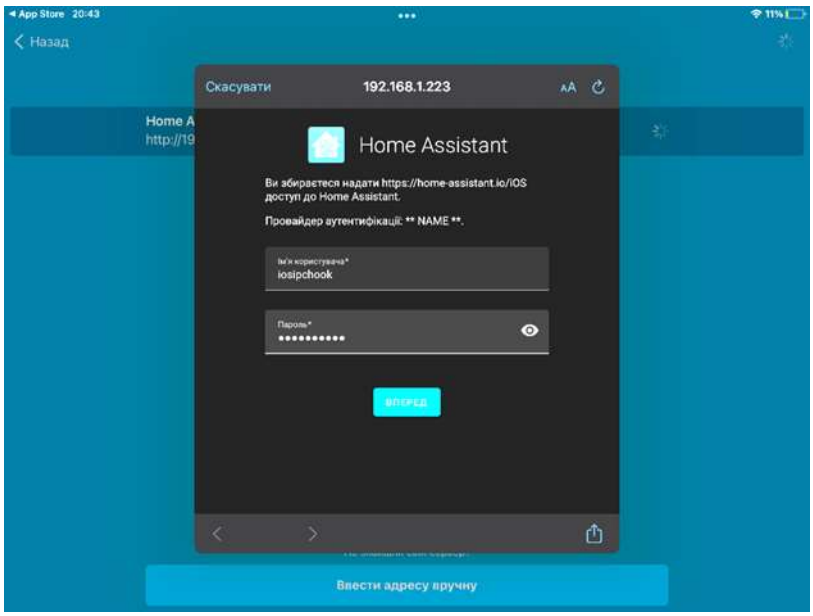

Рисунок. 3.1 – Зображення першого запуску додатка на iPad OS

Для користувача створено дві закладки, за допомогою яких можна слідкувати за показниками датчиків, які розділені по типам, для зручності іконки міняють колір при спрацюванні датчика, деякі мають анімацію, наприклад двері будуть відображені на іконці як відкриті/закриті, при постановці на сигналізацію, система автоматично зачинить електромеханічні замки, а іконки дверей отримають відповідне зображення, що буде свідчити про їх блокування. Також в інтерфейсі реалізована можливість керування електромеханічними замками та

повідомленнями, які за необхідності можна відключити, але це не стосується повідомлень про спрацювання сигналізації та постановки та зняття з охорони. На рис. 3.2 зображена перша та друга закладка інтерфейсу на Android смартфоні.

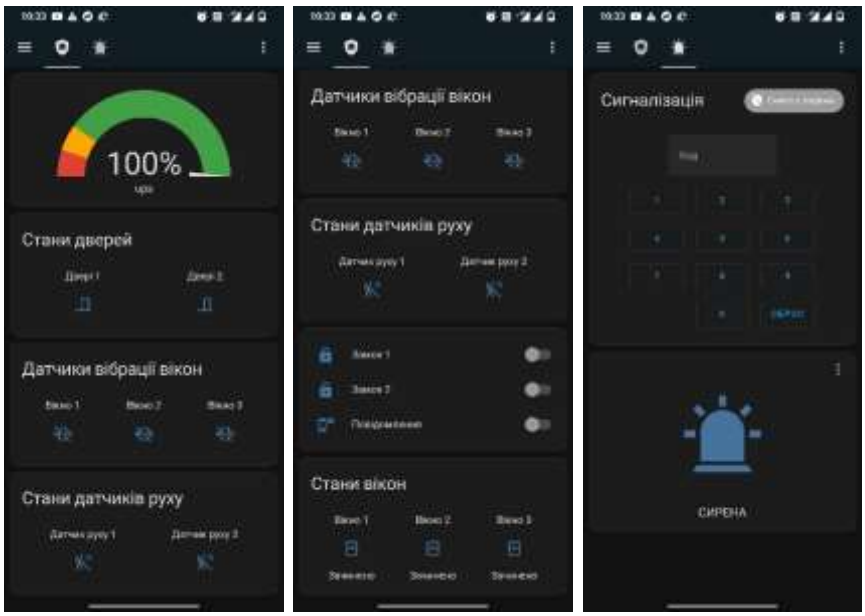

Рисунок. 3.2 – Зображення першої та другої закладки інтерфейсу

За допомогою другої закладки можна, використовуючи пароль, ставити та знімати з охорони об’єкт охорони, контролювати роботу сирени, та, за необхідності, відключити її. При постановці на охорону блокуються замки дверей, що буде відображено на першій закладці.

Висновки до Розділу 3

В даному розділі була описана вичерпна послідовність дій для встановлення та налаштування ОС Home Assistant. За допомогою вбудованих можливостей платформи та мови YAML було реалізовано інтерфейс інформаційної системи охорони будинку, який адаптовано для смартфону та планшету. Для зручності, в інтерфейсі створено дві закладки – для спостерігання та керування пристроями, активації сигналізації.

ВИСНОВКИ

В кваліфікаційній роботі на тему «Інформаційна система охорони розумного будинку на базі технологій IoT» була вивчена предметна область, завдяки чому було обрано вичерпний перелік необхідних компонентів для створення системи охорони, прийнято рішення про відмову використання хмарних обчислень на перевагу туманним, які відбуваються поруч, що значно підвищує безпеку та автономність системи. Для запровадження реалізації проекту було використано платформу Home Assistant, що забезпечить масштабування системи, використання обладнання не прив’язаного до виробника, а відкритий код та мова написання платформи Python дадуть нам безмежні можливості, які будуть обмежені, хіба що, нашою фантазією.

Для детального вивчення покрокових дій було розроблено діаграми різних методологій і стандартів, що допомогло в створенні алгоритмів автоматизацій. Запропоновано та впроваджено розумне ДБЖ, яке значно розширить можливості автоматизації та підвищить безпеку для серверу обробки даних.

Створено систему охорону будинку на базі технологій IoT, яка спроможна до масштабування та інтеграцій в інші IoT системи. Система яка працює, майже, з будь-якими пристроями та виробниками, спроможна відправляти інформацію про стан системи у месенджер та доступна з будь-якої точки світу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Катренко А. В. Системний аналіз: підр. За ред. В. В. Пасічника. Львів : Новий Світ-2000, 2011. 395 с.

2. Ушакова І. О. Основи системного аналізу об’єктів та процесів комп’ютеризації : навчальний посібник. Ч. 2 Харків : Вид. ХНЕУ, 2008. – 324 с.

3. Бродський Ю. Б., Молодецька К. В., Николюк О. М. Системний аналіз в економіці : навч. посіб. Житомир : ЖНАЕУ, 2014. 175 с.

4. Кутковецький В.Я. Ймовірнісні процеси і математична статистика в автоматизованих системах. Глава 3. Ймовірність безвідмовної роботи АСУ та її елементів. Навчальний посібник. – Миколаїв: Вид-во МДГУ, 2002. – 150 с. https://lib.chmnu.edu.ua/pdf/posibnuku/163/6.pdf (дата звернення: 30.12.2022).

5. Тиртишніков, О.І., Нікулін, М.Б., Корж, Ю.М. Теорія електричних кіл. Методи аналізу лінійних електричних кіл: навч. посіб. Полтава : ПолтНТУ імені Юрия Кондратюка, 2011. - Ч. 1. – 77 с. http://reposit.pntu.edu.ua/handle/PoltNTU/2660 (дата звернення: 10.03.2023).

6. Anna Johannson. Security in the Smart Home: Веб-сайт. URL: https://www.isaca.org/resources/news-and-trends/isaca-now-blog/2020/security-in-the-smart-home (дата звернення: 28.11.2022).

7. Bardia Eshghi. Top 10 IoT Communication Protocols in 2023: Веб-сайт. URL: https://research.aimultiple.com/iot-communication-protocol (дата звернення: 16.12.2022).

8. Terri Williams. How IoT Can Make Your Home Safer and More Secure: Веб-сайт. URL: https://www.techopedia.com/how-iot-can-make-your-home-safer-and-more-secure/2/34078 (дата звернення: 29.11.2022).

9. Інтернет речей: Веб-сайт. URL: https://uk.wikipedia.org/wiki/Інтернет_речей (дата звернення: 10.12.2022).

10. Курс IOT-IST-19Z. Веб-сайт. URL: https://netacad.com (дата звернення: 15.11.2022)

11. Курс Moodle ПНУ Проектування інформаційних систем. Веб-сайт. URL:http://beta.znau.edu.ua/ (дата звернення: 05.11.2022).

12. Курс Moodle ПНУ Системний аналіз. Веб-сайт. URL:http://beta.znau.edu.ua/ (дата звернення: 9.11.2022).

13. Охранная сигнализация. Веб-сайт. URL: https://ru.wikipedia.org/wiki/Охранная_сигнализация (дата звернення: 12.12.2022).

14. YAML. Веб-сайт. URL: https://uk.wikipedia.org/wiki/YAML (дата звернення: 12.04.2023)

15. Управління IT-проектами. Методичні вказівки. Веб-сайт. URL: https://core.ac.uk/download/pdf/84838992.pdf (дата звернення: 15.12.2022).

16. Датчик. Веб-сайт. URL: https://uk.wikipedia.org/wiki/Датчик (дата звернення: 15.12.2022).

17. Все про сучасні системи сигналізації: види, принцип роботи. Веб-сайт. URL: https://vencon.ua/ua/articles/vse-o-sovremennyh-sistemah-signalizacii-vidy-princip-raboty (дата звернення: 14.12.2022).

18. Інтернет речей. Веб-сайт. URL: https://uk.wikipedia.org/wiki/Інтернет_речей (дата звернення: 10.12.2022).

19. Курс IOT-IST-19Z. Веб-сайт. URL: https://netacad.com (дата звернення: 15.11.2022).

20. Курс Moodle ПНУ Проектування інформаційних систем. Веб-сайт. URL:http://beta.znau.edu.ua/ (дата звернення: 05.11.2022).

21. Курс Moodle ПНУ Системний аналіз. Веб-сайт. URL:http://beta.znau.edu.ua/ (дата звернення: 9.11.2022).

22. ESP8266 Technical Reference. Веб-сайт. URL: https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf (дата звернення: 13.04.2023).

23. Espressif. Production Testing Guide. Веб-сайт. URL: https://www.espressif.com/sites/default/files/documentation/production_Testing_Guide__EN.pdf (дата звернення: 13.04.2023).

24. Install Home Assistant Operating System. Веб-сайт. URL: https://www.home-assistant.io/installation/generic-x86-64/ (дата звернення: 13.03.2023).

25. Configuration.yaml. Веб-сайт. URL: https://www.home-assistant.io/docs/configuration/ (дата звернення: 13.03.2023).

26. Automation YAML. Веб-сайт. URL: https://www.home-assistant.io/docs/automation/yaml/ (дата звернення: 14.04.2023).

27. HACS. Веб-сайт. URL: https://hacs.xyz/ (дата звернення: 17.04.2023).

28. Telegram. Веб-сайт. URL: https://www.home-assistant.io/integrations/telegram/ (дата звернення: 25.04.2023).

ДОДАТКИ

ДОДАТОК А

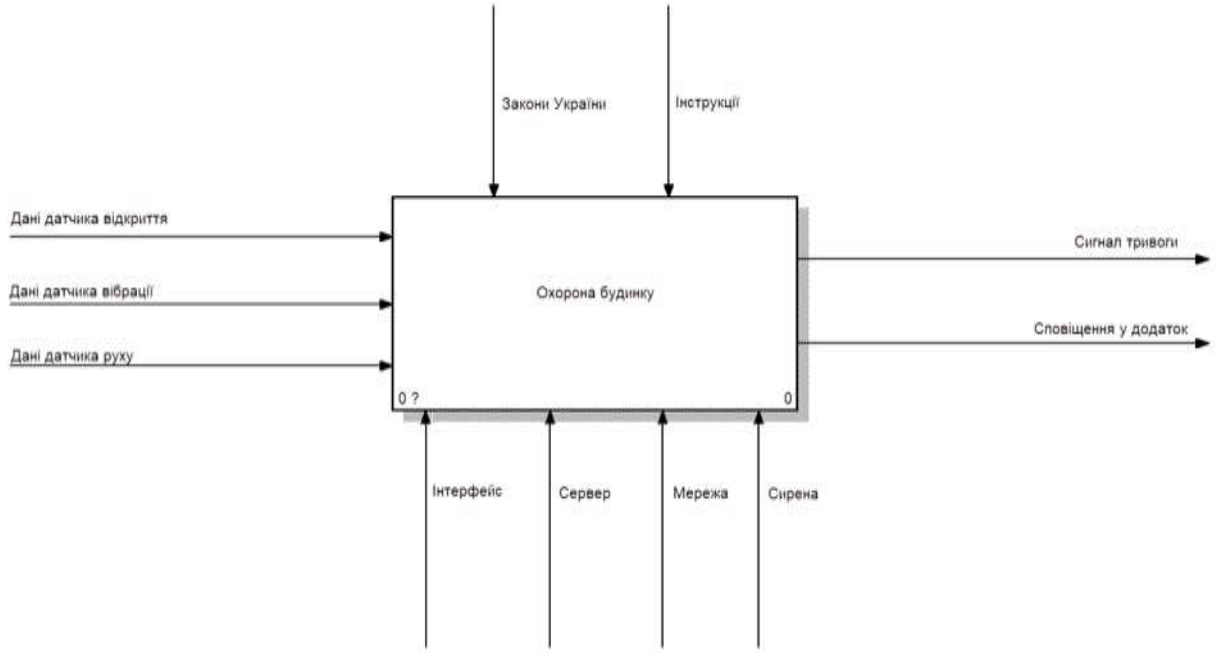

Рисунок А.1 – Діаграма нульового рівня IDEF0

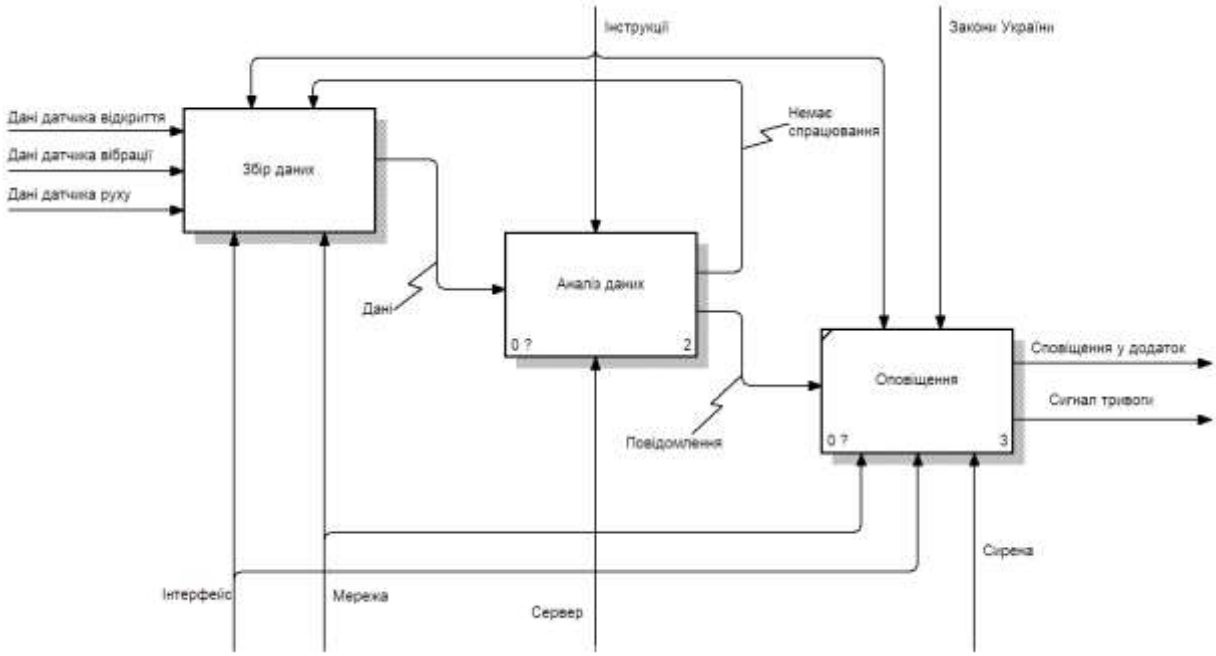

Рисунок А.2 – Декомпозиція нульового рівня

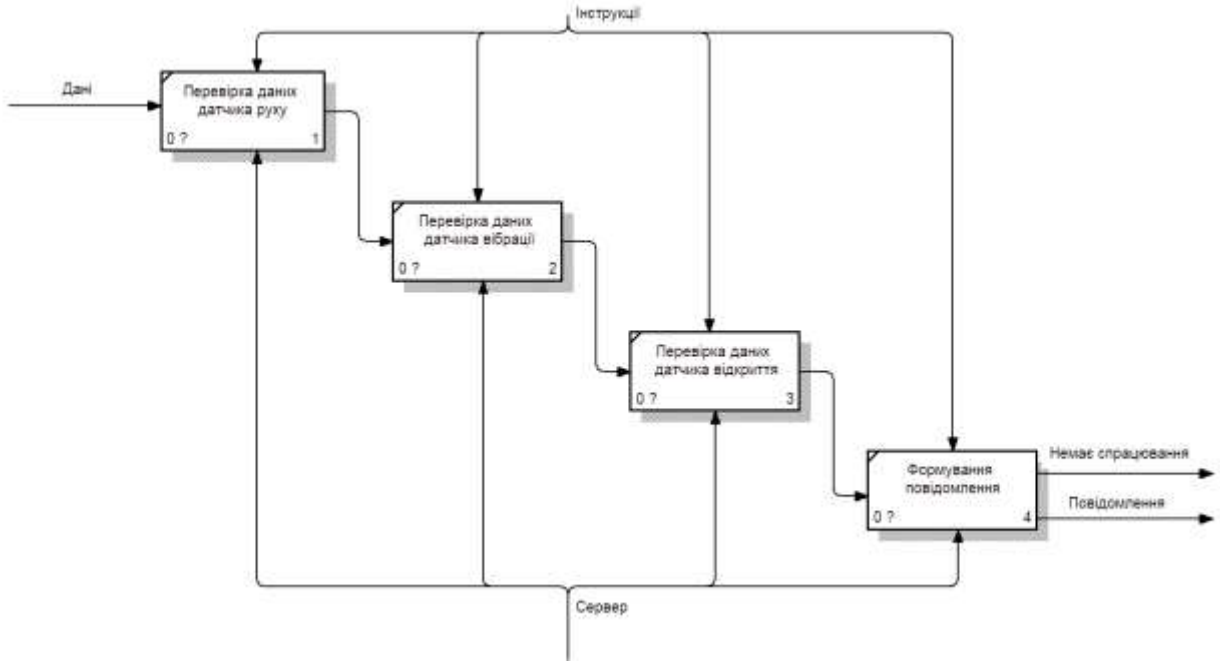

Рисунок А.3 – Декомпозиція блоку 2 першого рівня

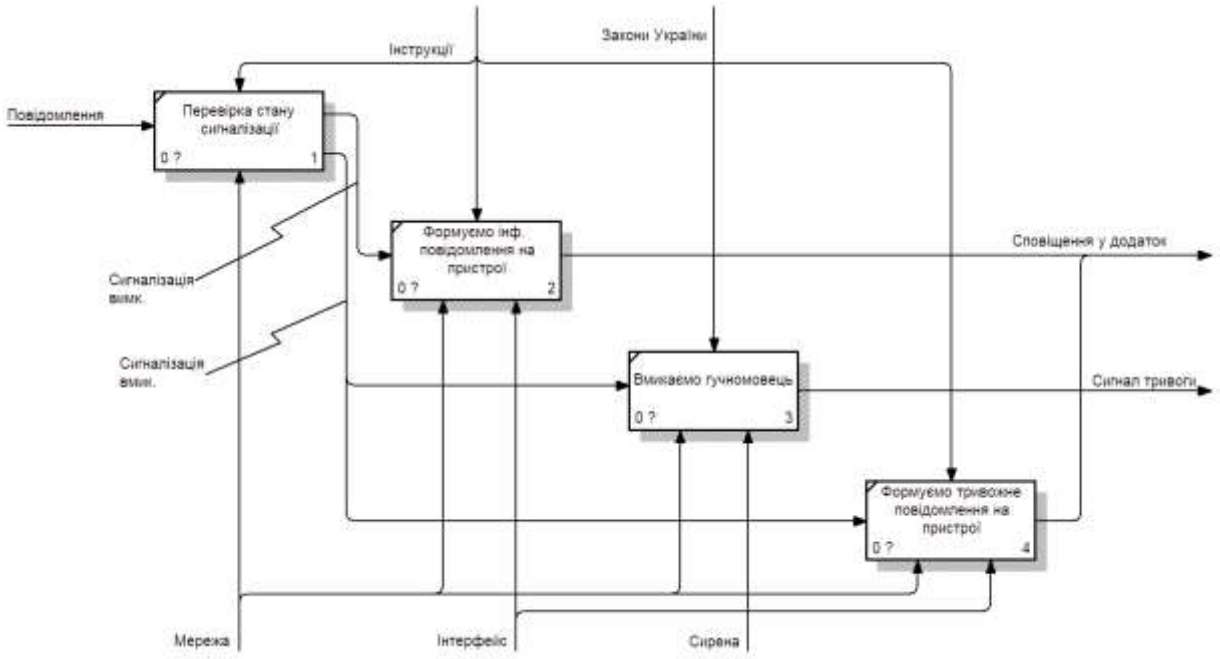

Рисунок А.4 – Декомпозиція блоку 3 першого рівня

ДОДАТОК Б

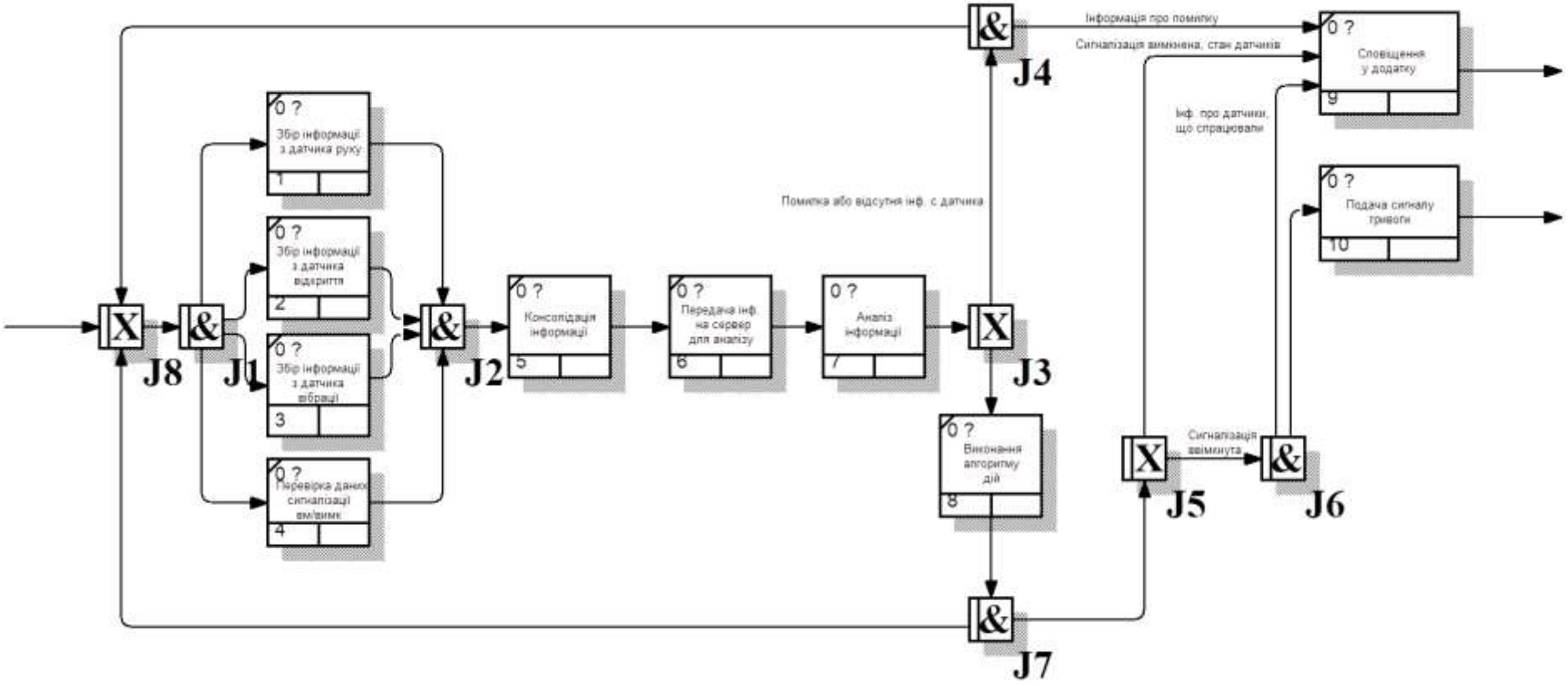

Рисунок Б.1 – Діаграма PFDD системи охорони будинку

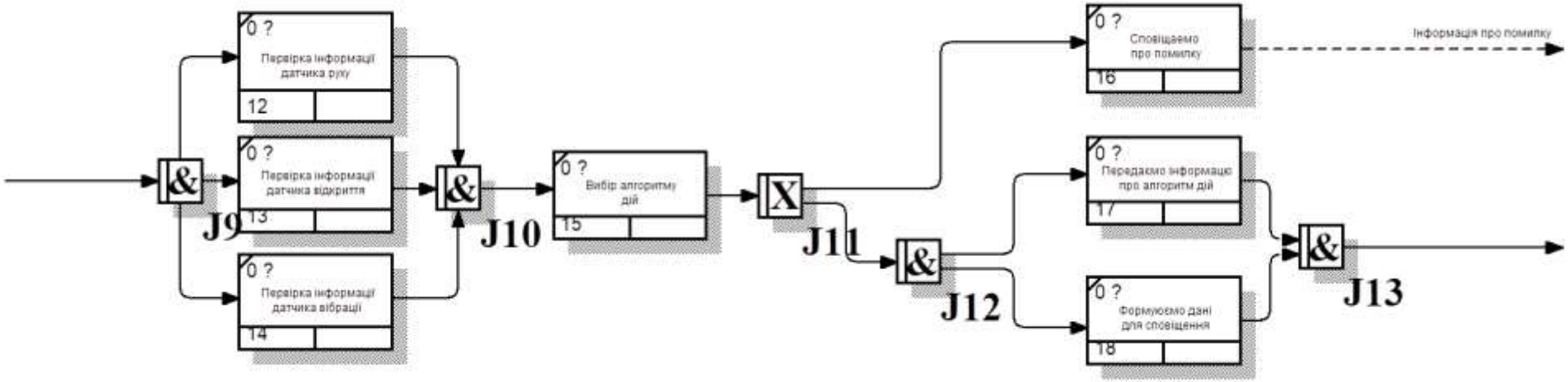

Рисунок Б.2 – Декомпозиція блоку 7 рис. Б.1

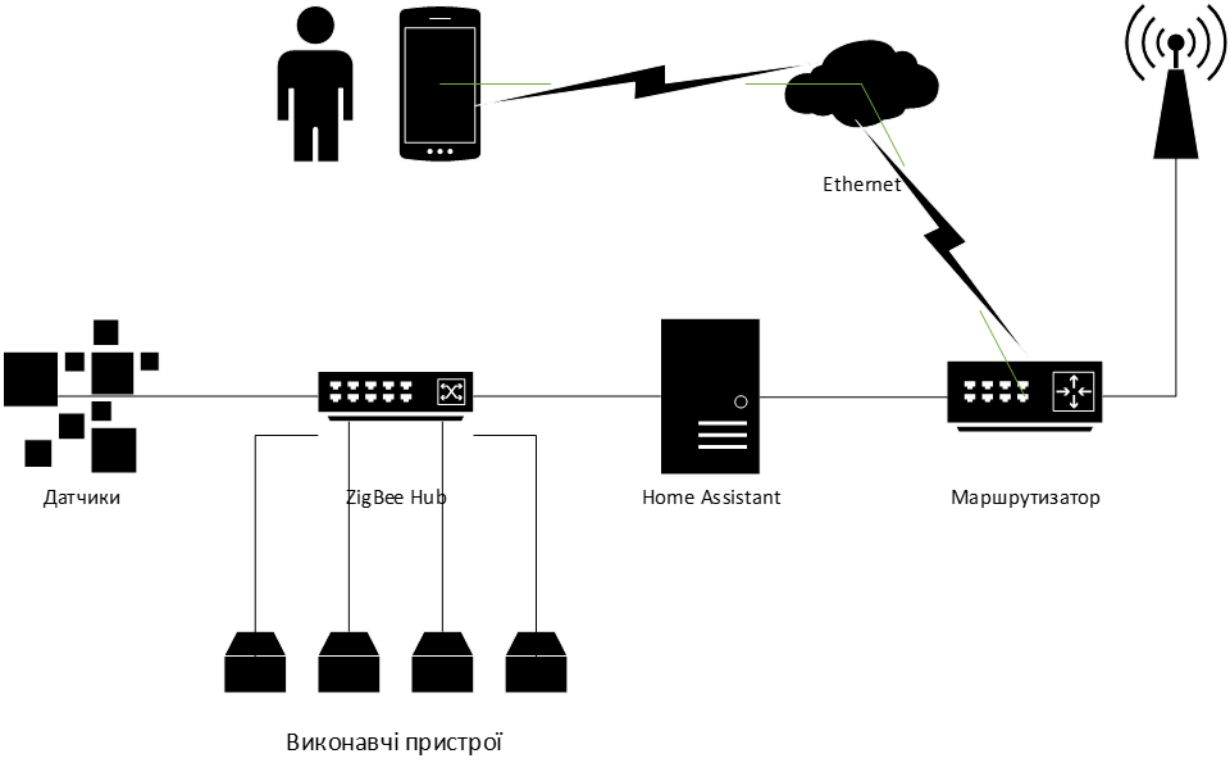

Рисунок Б.3 – Структура комп’ютерної мережі системи охорони будинку

ДОДАТОК В

Рисунок В.1 – Датчик відкриття дверей MiJia (MCCGQ01LM)

Рисунок В.2 – Датчик руху Sonoff SNZB-03

Рисунок В.3 – Датчик вібрації Aqara (DJT11LM)

Рисунок В.4 – Шлюз Sonoff ZBDongle-E Zigbee USB

Рисунок В.5 – Міні комп’ютер NiPoGi GK3V

Рисунок В.6 – Сирена Tervix Pro Line ZigBee Siren

Рисунок В.7 – 2CH Tuya ZigBee Relay Module

Рисунок В.8 – Маршрутизатор Xiaomi Mini

ДОДАТОК Г

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
#include <WiFiUdp.h>
#include <WakeOnLan.h>
#include <ArduinoJson.h>

WiFiUDP UDP;
WakeOnLan WOL(UDP);

const char* ssid = "_SSID_";
const char* password = "_PASSWORD_";
const char* wolMac = "00:00:00:00:00:00"; //MAC адреса сервера Home Assistant
const int batteryPin = A0; // пін АЦП
const int mainsPowerPin = D1; // цифровий пін
const int dividerResistor1 = 320000; //номінали резисторів R1
const int dividerResistor2 = 1000000;//та R2 в омах
const float minVoltage = 11; //Напруга батареї = 0%
const float maxVoltage = 12.6; //Напруга батареї = 100%
const char* token = "_YOUR_HOME_ASSISTANT_TOKEN_";
const char* UrlMainsPower = "http://192.168.1.223:8123/api/states/sensor.mainspower";
float getBatteryLevel() {
float voltage = analogRead(batteryPin) / 1023.0 * 3.3 * (dividerResistor1 + dividerResistor2) / dividerResistor1;
float batteryPercentage = (voltage - minVoltage) / (maxVoltage - minVoltage) * 100;
  if (batteryPercentage < 0) {return 0;}
  if (batteryPercentage > 100) {return 100;}
  return batteryPercentage;
}
void setup() {
Serial.begin(115200);
pinMode(mainsPowerPin, INPUT);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(1000);
}
WOL.setRepeat(3, 100);
float batteryPercentage = getBatteryLevel();
bool mainsPower = digitalRead(mainsPowerPin);
pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
digitalWrite(LED_BUILTIN, HIGH);
float batteryPercentage = getBatteryLevel();
bool mainsPower = digitalRead(mainsPowerPin);
StaticJsonDocument<200> jsonDoc;
jsonDoc["state"] = int(batteryPercentage);
JsonObject attributes = jsonDoc.createNestedObject("attributes");
attributes["unit_of_measurement"] = "%";
attributes["friendly_name"] = "ups";
attributes["state_class"] = "measurement";
attributes["device_class"] = "battery";
attributes["adc"] = analogRead(batteryPin);
attributes["mainspower"] = mainsPower;
String jsonString;
```

```
serializeJson(jsonDoc, jsonString);

WiFiClient client;
HTTPClient http;
http.begin(client, String(UrlMainsPower));
http.addHeader("Authorization", "Bearer " + String(token));
http.addHeader("Content-Type", "application/json");
int httpResponseCode = http.POST(jsonString);
if (httpResponseCode = 0) {
  if (batteryPercentage > 20 || mainsPower == true) {
WOL.sendMagicPacket(wolMac);
delay(30000);
}
}
delay(1000);
digitalWrite(LED_BUILTIN, LOW);
delay(2000);
}
```

ДОДАТОК Д

```
- id: '1682758904506'
  alias: Постановка/зняття з охорони
  description: 'Вмикання/вимиканн замків, сповіщення'
  trigger:
  - platform: state
    entity_id:
    - alarm_control_panel.home_alarm                    #перевіряємо стан панелі керування
  condition: []
  action:
  - choose:
    - conditions:
      - condition: state
        entity_id: alarm_control_panel.home_alarm
        state: armed_home                               #якщо стан «armed_home» (охорона)
      sequence:
      - service: notify.telegram_id                     #відправляємо повідомлення в
        data:                                           #телеграм
          message: Сигналізація активована. Замки зачинені.
          title: ALARM
      - service: switch.turn_on                         #використовуємо сервіс switch.turn_on для
        data:
          entity_id:
          - switch.sonoff_xxxxxxxxx1                    #реле, які керують електрозамками
          - switch.sonoff_xxxxxxxxx2
    - conditions:
      - condition: state
        entity_id: alarm_control_panel.home_alarm
        state: disarmed                                 #якщо стан «disarmed» (без охорони)
      sequence:
      - service: notify.telegram_id                     #відправляємо повідомлення в
        data:                                           #телеграм
          message: Знято з охорони
          title: ALARM
      - service: switch.turn_off                        #використовуємо сервіс switch.turn_off для
        data:
          entity_id:
          - switch.sonoff_xxxxxxxxx1                    #реле, які керують електрозамками та
          - switch.sonoff_xxxxxxxxx2
          - switch.tervix_pro_line_zigbee_siren         #для сирени
mode: single
```

ДОДАТОК Е

```
- id: '1682841343542'
  alias: Стани датчиків в телеграм
  description: 'Відправка повідомлень якщо активовано'
  trigger:
  - platform: state
    entity_id:
    - sensor.open_contact1                    #Перевіряємо стани атрибутів 'contact'
    - sensor.open_contact2                    #датчиків дверей на зміну з 'true' на 'false'
    attribute: contact
    to: 'false'
  - platform: state
    entity_id:
    - sensor.motion_occupancy1                #Перевіряємо стани атрибутів 'occupancy'
    - sensor.motion_occupancy2                #датчиків руху на зміну з false' на ' true '
    - sensor.motion_occupancy3
    attribute: occupancy
    to: 'true'
  - platform: state
    entity_id:
    - sensor.vibration_1                      #Перевіряємо стани атрибутів 'vibration'
    - sensor.vibration_2                      #датчиків вібрації на зміну з 'false' на ' true'
    - sensor.vibration_3
    attribute: vibration
    to: 'true'
  - platform: state
    entity_id:
    - sensor.vibration_1                      #Перевіряємо стани атрибутів 'angle_y'
    - sensor.vibration_2                      #датчиків вібрації на зміну даних
    - sensor.vibration_3
    attribute: angle_y
  condition:
  - condition: state
    entity_id: input_boolean.triger_message_telegran  #Перевіряємо необхідність відправлення
    state: 'on'                               #повідомлень
  action:
  - service: notify.telegram_id               #за необхідності відправляємо повідомлення
    data:                                     #з назвою датчика, який спрацював
      title: ALARM
      message: Спрацював {{ trigger.to_state.attributes.friendly_name }}
  mode: single
#Блоки «condition» та «action» для варіанту коду спрацювання сигналізації.
  condition:
  - condition: state
    entity_id: alarm_control_panel.home_alarm
    state: armed_home
  action:
  - service: notify.telegram_id
    data:
      title: ALARM
      message: ТРИВОГА! ПОРУШЕННЯ ПЕРИМЕТРУ ОХОРОНИ!
  - service: switch.turn_on
    data:
      entity_id: switch.tervix_pro_line_zigbee_siren
```

ДОДАТОК Є

```
- id: '1682841343555'
  alias: Безпека системи
  description: 'Вимикання сереверу при критичному заряді батареї'
  trigger:
    - platform: template
      value_template: "{{ states('sensor.mainspower') | int < 20 and is_state_attr('sensor.mainspower',
'mainspower', false) }}"
  action:
    - service: notify.telegram_id
      data:
        title: ALARM
        message: Критичний заряд батареї. Систему буде вимкнено.
    - service: homeassistant.turn_off
    - delay: 00:00:30
    - service: shell_command.turn_off_computer
```