Міністерство освіти і науки України
Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

КВАЛІФІКАЦІЙНА РОБОТА

за освітнім ступенем «бакалавр»

на тему:

«Система безпеки на базі Інтернету речей з вебінтерфейсом на WebIOPi»

Виконав:
студент IV-го курсу
групи ІПЗ-41
спеціальності 121 «Інженерія
програмного забезпечення»
Шульга Денис Олексійович
Керівник:
к.т.н., доцент Шинкарчук Н.В.

Рівне – 2026

АНОТАЦІЯ ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ

Шульга Д.О. Система безпеки на базі Інтернету речей з вебінтерфейсом на WebIOPi. Кваліфікаційна робота на здобуття ступеня «бакалавр» за спеціальністю 121 «Інженерія програмного забеспечення» – Рівненський державний гуманітарний університет. Рівне 2026. 51 с.

Кваліфікаційна робота присвячена розробці системи безпеки на базі Інтернету речей із використанням програмного середовища WebIOPi та одноплатного комп'ютера Raspberry Pi 3 Model B. Основна увага приділена створенню вебінтерфейсу для віддаленого моніторингу та керування компонентами системи безпеки через локальну мережу або мережу Інтернет. У роботі розглянуто теоретичні основи побудови систем безпеки на базі Інтернету речей, їх структуру, принципи функціонування, засоби зв'язку, питання захисту даних та надійності роботи. Досліджено апаратні та програмні можливості Raspberry Pi 3 Model B, а також процес розробки вебінтерфейсу для керування такими компонентами системи, як датчик руху, інфрачервоний бар'єрний датчик перешкоди KY-032, LED-світлодіод, активний зумер та інші виконавчі пристрої. У результаті виконання роботи розроблено прототип системи безпеки, що забезпечує виявлення несанкціонованого доступу, подачу звукової та світлової сигналізації, а також дистанційне керування та спостереження за станом системи через вебінтерфейс.

Ключові слова: Інтернет речей, система безпеки, WebIOPi, Raspberry Pi, вебінтерфейс, GPIO, датчики, сигналізація.

ANNOTATION TO THE QUALIFICATION WORK

Shulga D.O. Security system based on the Internet of Things with a web interface on WebIOPi. Qualification work for the degree of bachelor" in specialty 121 "Software Engineering" - Rivne State Humanitarian University. Rivne 2026. 51 p.

The qualification work is devoted to the development of a security system based on the Internet of Things using the WebIOPi software environment and a single-board computer Raspberry Pi 3 Model B. The main attention is paid to the creation of a web interface for remote monitoring and control of security system components via a local network or the Internet. The work considers the theoretical foundations of building security systems based on the Internet of Things, their structure, principles of operation, means of communication, issues of data protection and reliability of operation. The hardware and software capabilities of the Raspberry Pi 3 Model B were investigated, as well as the process of developing a web interface for controlling such system components as a motion sensor, an infrared barrier sensor KY-032, an LED, an active buzzer, and other actuators. As a result of the work, a prototype of a security system was developed that provides detection of unauthorized access, sound and light signaling, as well as remote control and monitoring of the system status via a web interface.

Keywords: Internet of Things, security system, WebIOPi, Raspberry Pi, web interface, GPIO, sensors, alarm.

ЗМІСТ

ВСТУП

Актуальність роботи. Сьогодні системи безпеки стають все більш важливими для захисту будинків, офісів та інших об’єктів. Звичайні охоронні системи часто є дорогими та мають обмежені можливості. Вони не завжди підтримують віддалений доступ і потребують спеціального обладнання. Технології Internet of Things (Інтернету речей) дозволяють створювати доступні та зручні системи безпеки. Такі системи можуть автоматично виявляти небезпеку, передавати інформацію користувачу та керувати виконавчими пристроями через мережу інтернет.

Для реалізації такої системи доцільно використати Raspberry Pi 3 Model B. це недорога та функціональна платформа, яка підтримує підключення різних датчиків і використаних пристроїв. Для створення вебінтерфейсу використовується фреймворк WebIOPi, який дозволяє керувати GPIO-портами через браузер.

Мета роботи: розробити систему безпеки на базі Інтернету речей з вебінтерфейсом на WebIOPi. В першому розділі потрібно провести огляд літературних джерел і розглянути основи побудови системи безпеки на базі технології Інтернету речей. В другому розділі описати сценарій проєктування архітектури та вибір компонентів системи, зокрема використані датчики та супровідні електронні пристрої. В третьому розділі здійснити налаштування фреймворку Інтернету речей WebIOPi (платформа для створення вебінтерфейсів) на одноплатному комп’ютері Raspberry Pi 3 Model B. В четвертому розділі відобразити матеріали практичної частини роботи: проєктування, розробка та реалізація вебінтерфейсу системи безпеки на базі WebIOPi.

Об’єкт дослідження є процес функціонування системи безпеки на базі Інтернету речей.

Інструментом дослідження є апаратні та програмні засоби, що використовуються для побудови системи безпеки. Основною апаратною платформою обрано Raspberry Pi 3 Model B, який забезпечує обробку даних, керування підключеними пристроями та взаємодію з користувачем. Для створення

вебінтерфейсу та організації доступу до GPIO-портів використовується WebIOPi. Програмна частина системи реалізується за допомогою мови програмування Python, а для розробки інтерфейсу користувача застосовуються HTML, CSS та JavaScript. У процесі дослідження використовуються RFID-зчитувач MFRC522, інфрачервоний бар'єрний датчик перешкоди KY-032, RGB LED-модуль KY-016, пасивний зумер та інші датчики й виконавчі пристрої, які забезпечують виявлення подій та формування сигналізації.

Предметом дослідження є методи та засоби проектування й реалізації системи безпеки з вебінтерфейсом для моніторингу та керування апаратними компонентами.

Завдання дослідження. Для досягнення поставленої мети в роботі необхідно провести аналіз сучасних систем безпеки на базі Інтернету речей та дослідити принципи їх побудови. Важливим завданням є вивчення апаратних і програмних можливостей Raspberry Pi 3 Model B та функціональних можливостей WebIOPi для організації віддаленого керування GPIO-портами. У ході роботи потрібно обґрунтувати вибір датчиків і виконавчих пристроїв, розробити схему їх підключення та створити програмне забезпечення для обробки сигналів і керування системою. Окремим завданням є розробка вебінтерфейсу, який забезпечуватиме моніторинг стану системи та дистанційне керування її компонентами. Завершальним етапом дослідження є тестування створеної системи, аналіз результатів її роботи та оцінка ефективності запропонованого рішення.

Апробація дослідження. Результати виконання кваліфікаційної роботи, окремі її аспекти та одержані узагальнення і висновки були оприлюднені на звітній науковій конференції викладачів, співробітників і здобувачів вищої освіти Рівненського державного гуманітарного університету за 2025 рік (м. Рівне, 2026).

Структура роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатку. У першому розділі розглянуто теоретичні основи побудови систем безпеки на базі Інтернету речей,

проаналізовано сучасні підходи до організації охоронних систем, їх структуру, принципи функціонування, засоби зв’язку та питання інформаційної безпеки. У другому розділі наведено характеристику апаратного та програмного забезпечення, обґрунтовано вибір одноплатного комп’ютера Raspberry Pi 3 Model B, програмного середовища WebIOPi, а також описано використані датчики та виконавчі пристрої. У третьому розділі виконано проектування системи безпеки, розроблено структурну схему, алгоритм роботи системи та логіку взаємодії між її компонентами. У четвертому розділі представлено практичну реалізацію системи, описано процес створення програмного забезпечення та вебінтерфейсу, наведено результати тестування й оцінку працездатності розробленого рішення. Робота завершується висновками, у яких узагальнено отримані результати дослідження.

РОЗДІЛ 1
ОСНОВИ ПОБУДОВИ СИСТЕМИ БЕЗПЕКИ НА БАЗІ ІНТЕРНЕТУ РЕЧЕЙ

1.1. Поняття систем безпеки та їх класифікація

Система безпеки комплекс технічних, організаційних, правових та інженерних засобів, спрямованих на захист життя, здоров'я людей, майна, інформації та навколишнього середовища від різноманітних загроз. Вони забезпечують порядок, сповіщають про небезпеку та допомагають запобігти негативним наслідкам [1].

Основні складові системи безпеки нашого дослідження є датчики (сенсори), приймально-контрольовані пристрої а також виконавчі пристрої. Основою системи безпеки об'єкту нашого дослідження є датчики руху, диму, розбиття скла, відкриття дверей, відеокамери, замки, сирени та інші. За допомогою цих компонентів можна розробити охоронну систему яка буде забезпечувати стабільну роботу та довгу підтримку та покращення. Кожен із цих датчиків можна побудувати як окремі системи безпеки, але й також їх комбінувати між собою постійно покращуючи та облегчити їх використання для споживача.

До основних засобів виявлення загроз належать датчики руху, датчики диму, датчики розбиття скла, магнітоконтактні датчики відкриття дверей і вікон, а також відеокамери спостереження. Датчики руху дозволяють виявити присутність сторонніх осіб у контрольованій зоні. Датчики диму забезпечують раннє виявлення ознак пожежі. Датчики розбиття скла реагують на характерні звукові коливання або вібрації, що виникають під час пошкодження віконних конструкцій. Магнітоконтактні датчики фіксують зміну положення дверей або вікон, а відеокамери забезпечують візуальний контроль за об'єктом у реальному часі. Виконавчі пристрої призначені для реагування на виявлені загрози. До них належать сирени, світлові індикатори, електромеханічні замки, реле керування та інші пристрої, які активуються за командою контролера. У разі спрацювання одного або

кількох датчиків система може ввімкнути звукову сигналізацію, заблокувати двері, активувати освітлення або надіслати повідомлення користувачу через вебінтерфейс чи мережеві сервіси. Завдяки цьому забезпечується швидке реагування на небезпечну ситуацію.

Кожен із перелічених датчиків може використовуватися як самостійна система безпеки для вирішення окремого завдання. Наприклад, датчик диму може працювати як автономний пожежний сповіщувач. Проте найбільша ефективність досягається шляхом інтеграції кількох типів датчиків у єдину систему. Поєднання різних сенсорів дозволяє суттєво підвищити точність виявлення загроз, зменшити ймовірність помилкових спрацювань та розширити функціональні можливості системи.

Сучасні системи безпеки дедалі частіше будуються на основі концепції Інтернету речей. У таких системах усі компоненти об’єднані в єдину мережу та можуть передавати дані через Інтернет. Це забезпечує користувачеві можливість дистанційно контролювати стан об’єкта, переглядати журнал подій, змінювати налаштування та отримувати миттєві повідомлення про тривожні ситуації. Для реалізації таких функцій можуть використовуватися програмні платформи на кшталт WebIOPi, які забезпечують керування GPIO-портами та створення веб-інтерфейсів для взаємодії з апаратною частиною системи.Отже, система безпеки є складною багатокомпонентною структурою, яка об’єднує датчики, контролер, канали зв’язку та виконавчі пристрої. Її ефективність визначається правильним вибором компонентів, якістю програмного забезпечення та можливістю інтеграції різних підсистем. Комбінування кількох типів сенсорів і використання сучасних IoT-технологій дозволяє створити гнучку, масштабовану та зручну у використанні систему, здатну забезпечити надійний захист об’єкта та швидке реагування на потенційні загрози.

1.2. Загрози та сценарії використання систем безпеки

Сучасні системи безпеки є важливим елементом захисту різних об’єктів, починаючи від приватних будинків і закінчуючи промисловими підприємствами та складськими приміщеннями. Їх основне призначення полягає у своєчасному виявленні потенційних загроз і забезпеченні швидкого реагування на них. Умови експлуатації таких систем можуть суттєво відрізнятися, однак принцип їх роботи залишається спільним, фіксація події, її обробка та подальше інформування користувача або активація відповідних виконавчих механізмів.

До найпоширеніших загроз, які враховуються під час проєктування систем безпеки, належить несанкціоноване проникнення на територію або в приміщення. Це може бути як навмисне вторгнення з метою крадіжки або пошкодження майна, так і випадкове потрапляння сторонніх осіб у контрольовану зону. Особливо актуальними такі ситуації є для об’єктів, що залишаються без постійного нагляду у нічний час або протягом тривалого періоду.

Окрім фізичного проникнення, системи безпеки також можуть реагувати на появу руху в заборонених або обмежених зонах. У таких випадках використовується датчик руху або інфрачервоні сенсори, які дозволяють виявляти зміни в навколишньому середовищі. Це дає змогу зафіксувати присутність людини або об’єкта там, де його бути не повинно. Подібні сценарії широко застосовуються у системах охорони складів, лабораторій, серверних приміщень та приватних будинків. Окрему увагу також приділяють ситуаціям, пов’язаним із пошкодженням або втручанням у роботу обладнання. Це можуть бути спроби вимкнення датчиків, розриву з’єднань або фізичного впливу на пристрої системи. У більш складних випадках враховуються навіть аварійні ситуації, наприклад раптове зникнення живлення або вихід з ладу окремих компонентів, що також може потребувати сповіщення користувача.

У системах, побудованих на базі Інтернету речей, сценарії роботи реалізуються через взаємодію датчиків, мікроконтролерів та виконавчих пристроїв. Наприклад, при спрацюванні інфрачервоного датчика перешкоди система фіксує зміну стану в контрольованій зоні та передає сигнал на обробку мікрокомп'ютеру. Далі відбувається аналіз отриманої інформації, після чого система приймає рішення про подальші дії. Це може бути активація світлової індикації за допомогою LED-індикатора, запуск звукового сигналу через бузер або оновлення статусу системи у вебінтерфейсі користувача.

Важливою особливістю таких систем є їх здатність працювати в режимі реального часу. Це означає, що між виникненням події та реакцією системи проходить мінімальна кількість часу. Завдяки цьому підвищується ефективність реагування та зменшується ризик виникнення критичних наслідків. Крім того, використання мережевих технологій дозволяє користувачу отримувати інформацію про стан об'єкта віддалено, незалежно від його місцезнаходження. У сучасних умовах системи безпеки все частіше інтегруються з технологіями Інтернету речей, що дозволяє об'єднувати в єдину мережу різні пристрої та забезпечувати їх взаємодію. Це відкриває можливість створення більш гнучких та масштабованих рішень, які можуть адаптуватися до конкретних умов експлуатації. Наприклад, система може бути розширена додатковими датчиками або модулями оповіщення без суттєвої зміни загальної архітектури [2].

Таким чином, аналіз можливих загроз і сценаріїв використання є важливою складовою процесу проєктування систем безпеки. Саме на основі цього аналізу визначаються вимоги до апаратної та програмної частини системи, а також формується логіка її роботи. Це дозволяє створити ефективне рішення, здатне забезпечити надійний контроль за об'єктом та своєчасне реагування на потенційні загрози.

1.3. Безпека Інтернету речей

Інтернет речей тільки почав розвиватись, не відстаючи від часу та можливостей цей фактор є як і позитивний так і негативний.

З позитивної точки зору це добре так як даний вид технології постійно оновлюється та розвивається. Негативним є те що зі швидкістю розвитку страждає безпека системи. Знадобиться багато часу що б закрити певні недоліки з доступом та зробити систему надійною та недоступною для інших недобросовісних користувачів. Порушення конфіденційності через мережі зв'язку в даний момент часу є дуже частим явищем. Велику загрозу несе керування пристроїв за допомогою машинної взаємодії. Жодну написану людиною програму не можна вважати стовідсотково безпечною для користування. Постійно пишуться патчі для виправлення помилок для забезпечення безпеки користувачів та їхніх даних. Така ж доля чекає датчики в інтернет-пристроях. Найнебезпечніші кібератаки є DDoS-атаки, її мета це захоплення системних ресурсів та залишає трудним доступ до інформації іншим користувачам. DDoS-атака напад на комп'ютерну систему з наміром зробити комп'ютерні ресурси недоступними користувачам, для яких комп'ютерна система була призначена. Даний тип атаки працює за принципом штучного перенавантаження мережевого ресурсу великою кількістю запитів, що надходять одночасно з багатьох різних джерел. На рисунку 1.1. зображено схему за яким принципом працює DDoS-атака.

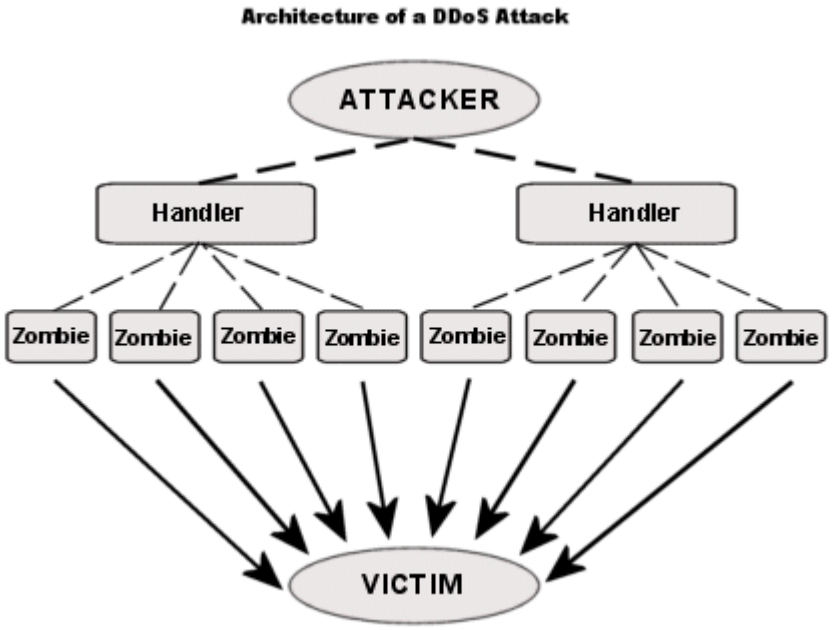

Рисунок 1.1. Принцип роботи DDoS-атаки

Ще одним видом видом загроз є відсутність шифрування даних. Навіть зараз багато пристроїв Інтернету речей (IoT), як і раніше, використовують незахищені протоколи зв'язку, такі як HTTP, Telnet або незашифрований MQTT, для передачі даних. Чи це заводське налаштування, системи або прямі трансляції з камер, незашифрований трафік може бути перехоплений зловмисниками за допомогою простих інструментів, таких як аналізатори пакетів що дозволяє майже без особливого підходу перехопити дані.

Після перехоплення отримані дані можуть бути, змінені або навіть використані для атак із заміною зв'язку. У критично важливих системах, такі як системи шутчного інтелекту або системи охорони здоров'я, такі помилки не тільки ставлять під загрозу конфіденційність даних, але можуть призвести до реальних порушень [3].

Також потрібно слідкувати за оновленнями пристроїв так як розробники з кожним оновленням пишуть патчі безпеки, так як даний момент постійно тестується та покращується.

Створення надійних паролів є те важливим нюансом. Неважкі паролі або комбінації можна ввести за кілька спроб або ж для того хакери пишуть спеціальне

програмне забезпечення для розшифрування. Враховуючи всі моменти для обходу захисту тої чи іншої системи. Тому чим важчим буде пароль тим важче зловмисникам буде важче розшифрувати пароль та мати доступ для даних.

Відсутність стандартизації є проблемою для Інтернету речей (IoT), оскільки вона ускладнює взаємодію та роботу різних пристроїв і систем. Пристрої IoT виготовляються багатьма різними виробниками та можуть використовувати широкий спектр протоколів зв’язку та форматів даних, що ускладнює розробку єдиного стандарту, який би працював на всіх пристроях. Відсутність стандартизації Відсутність стандартизації створює проблеми сумісності, що може призвести до проблем сумісності та обмеженої функціональності. Наприклад, якщо два пристрої IoT не можуть взаємодіяти один з одним, оскільки вони використовують різні протоколи або формати даних, вони не можуть працювати разом для досягнення спільної мети. Відсутність стандартизації також може спричинити ризики для безпеки, оскільки хакерам може бути легше використовувати вразливості в системах, які не використовують стандартні протоколи безпеки. Крім того, це може перешкоджати розробці нових програм і послуг, які можуть принести цінність бізнесу та споживачам. Загалом, відсутність стандартизації в галузі IoT може призвести до зниження ефективності, збільшення витрат і уповільнення темпів інновацій. Тому існує потреба в стандартизованих протоколах зв’язку, форматах даних і заходах безпеки, щоб забезпечити повну реалізацію потенціалу IoT. Проблеми сумісності Проблеми сумісності можуть бути поширеною проблемою для пристроїв Інтернету речей (IoT). Існує кілька причин, чому можуть виникати проблеми сумісності в IoT, зокрема: Несумісність протоколів: пристрої IoT використовують різні протоколи зв’язку, що може призвести до проблем несумісності між пристроями різних виробників. Протоколи безпеки: пристрої IoT мають різні протоколи безпеки, що також може призвести до проблем несумісності. Наприклад, якщо один пристрій використовує безпечніший протокол шифрування, ніж інший, вони можуть не мати змоги спілкуватися один з одним. Оновлення

прошивки: пристрої IoT часто отримують оновлення прошивки, що іноді може спричинити проблеми сумісності, якщо прошивка одного пристрою оновлюється, а інший залишається на старішій версії. Вимоги до живлення: пристрої IoT мають різні вимоги до живлення, що може призвести до проблем сумісності. Наприклад, якщо один пристрій потребує більше енергії, ніж інший, вони можуть не мати змоги спілкуватися один з одним. Фізична сумісність: пристрої IoT можуть бути несумісними з існуючою інфраструктурою, такою як датчики, які потребують певного типу проводки або роз'ємів. Щоб вирішити проблеми сумісності з Інтернетом речей, виробники та постачальники програмного забезпечення спільно працюють над розробкою стандартних протоколів, які можуть використовуватися всіма пристроями Інтернету речей, а також над створенням пристроїв, які є більш гнучкими та адаптованими до різних типів мереж та інфраструктури. IEEE також відіграє важливу роль у створенні стандартів. IEEE (Інститут інженерів з електротехніки та електроніки) – це професійна організація, яка відіграє вирішальну роль у стандартизації Інтернету речей (IoT). IEEE має кілька робочих груп, що займаються розробкою стандартів для пристроїв, мереж та протоколів IoT. Окрім розробки стандартів для IoT, IEEE також надає ресурси та підтримку спільноті IoT. IEEE проводить конференції та семінари з IoT, а також публікує дослідницькі роботи, книги та статті про технології та застосування IoT. Також важливо, щоб користувачі досліджували та купували пристрої від авторитетних виробників, які, як відомо, добре працюють з іншими пристроями в їхній екосистемі IoT [4].

1.4. Порівняння традиційних методів безпеки та безпеки Інтернету речей

Сучасні системи безпеки в нас час активно набирають обертів та не відстають від розвитку цифрових технологій. Традиційні системи безпеки постійно

покращуються, навіть змінюються для забезпечення безпечної роботи з пристроями IoT. Дані рішення дозволяють як і автоматизувати так і забезпечити вищий рівень захисту.

Традиційні методи захисту використовують автоматичні або напівавтоматичні комплекси технічних засобів. Застосовуються для охоронних систем, систем сигналізації, систем доступу тощо. До їхніх переваг можна віднести їхню відносну стабільність та подекуди неповну залежність до мережі. Ще одна перевага таких систем є те що вони керуються та налаштовуються локально, наприклад на підприємствах чи в приватному будинку. Незважаючи на показники надійності такі системи важко розгортати у великі масштаби.

IoT-системи безпеки базуються на використанні мережевих пристроїв, які взаємодіють між собою через Інтернет. Основу таких систем складають сенсори, контролери (наприклад Raspberry Pi 3 Model B), виконавчі механізми та програмне забезпечення.

Головною відмінністю IoT-систем є їх здатність до віддаленого моніторингу та управління в режимі реального часу. Користувач може отримувати інформацію про стан системи через веб-інтерфейс або мобільний додаток, а також керувати пристроями незалежно від свого місцезнаходження. Такі системи забезпечують високий рівень автоматизації, можливість інтеграції з іншими сервісами та використання хмарних технологій для обробки даних.

Сучасні системи безпеки на основі Інтернету речей ґрунтуються на широкому спектрі технологій та протоколів, які забезпечують ефективну комунікацію, обробку, зберігання та захист даних, що надходять від численних сенсорів і виконавчих пристроїв. Сенсорні пристрої є основою такої системи, вони фіксують фізичні параметри об’єкта, такі як рух, температура, рівень освітленості, звук та інші показники навколишнього середовища, і передають цю інформацію для подальшої обробки. Вибір конкретних сенсорів і способу їх підключення залежить від специфіки об’єкта, умов експлуатації та вимог до швидкості реагування системи.

Підключення пристроїв здійснюється за допомогою протоколів передачі даних, таких як Wi-Fi, Ethernet, Bluetooth Low Energy, ZigBee, LoRaWAN і NB-IoT, кожен з яких має свої переваги та обмеження. Наприклад, ZigBee та LoRaWAN забезпечують низьке енергоспоживання і широкий радіус дії, що особливо важливо для автономних сенсорів у великих будівлях чи промислових об’єктах, тоді як Wi-Fi та Ethernet забезпечують високу пропускну здатність, необхідну для передавання потокового відео та великих обсягів даних, що надходять від камер відеоспостереження. Мережевий рівень системи не лише передає дані, але й забезпечує їх пріоритетність, що гарантує оперативну реакцію на критичні події, а також застосування резервних каналів для забезпечення відмовостійкості та надійності роботи.

Для організації обробки даних сучасні системи застосовують комбінацію локальної обробки на пристроях (edge computing) та централізованого аналізу на серверах або у хмарних платформах. Локальна обробка дозволяє миттєво реагувати на загрози, наприклад активувати сигналізацію, закрити двері або направити виконавчий механізм, тоді як централізована обробка забезпечує аналіз великих масивів даних, прогнозування потенційних загроз та побудову аналітичних моделей. Все більше впроваджуються алгоритми штучного інтелекту та машинного навчання, які дозволяють системі не лише реагувати на події, але й прогнозувати їх виникнення, аналізуючи історичні дані і визначаючи шаблони нормальної поведінки об’єкта. Це підвищує точність системи, зменшує кількість хибних спрацьовувань і дозволяє фокусувати ресурси на дійсно критичних подіях. При цьому важливою є сумісність з різними типами сенсорів та обладнання різних виробників, що забезпечується використанням стандартних протоколів і відкритих інтерфейсів для взаємодії компонентів системи.

Передача даних у системі безпеки на основі IoT повинна бути захищеною, оскільки підключення до мережі Інтернет створює ризики несанкціонованого доступу, перехоплення даних або втручання у роботу системи. Для цього

застосовуються різні криптографічні методи шифрування, автентифікація користувачів, контроль прав доступу та механізми аудиту дій. Шифрування забезпечує безпеку даних при їх передачі та зберіганні, автентифікація гарантує, що доступ до системи отримують лише уповноважені особи, а контроль доступу дозволяє обмежити можливості кожного користувача відповідно до його ролі. Регулярне оновлення програмного забезпечення та прошивок пристроїв дозволяє усувати вразливості, підвищувати безпеку та підтримувати сумісність з новими протоколами і технологіями. Для забезпечення ефективної інтеграції з хмарними сервісами та зовнішніми інформаційними системами використовуються протоколи RESTful API, WebSockets та інші сучасні методи обміну даними, що дозволяють створювати централізоване управління, підтримувати постійне двонаправлене з'єднання і забезпечувати високу оперативність реагування системи на події.

Важливою складовою системи є передача потокового відео та аудіо від камер і мікрофонів. Для цього застосовуються протоколи і алгоритми стиснення даних, такі як H.264 або H.265, що дозволяють передавати якісне відео при зменшенні обсягу інформації. Це особливо актуально для моніторингу великих територій та об'єктів з численними точками спостереження.

Таким чином, технології та протоколи IoT, які застосовуються у сучасних системах безпеки, формують комплексну інфраструктуру, що забезпечує збір, передачу, обробку та аналіз даних у режимі реального часу, захист інформації, інтеграцію з хмарними сервісами та зовнішніми системами, а також енергоефективну і автономну роботу сенсорів і виконавчих пристроїв. Завдяки цьому такі системи є гнучкими, масштабованими і здатними ефективно працювати в умовах сучасного цифрового середовища, забезпечуючи високий рівень безпеки об'єктів та контроль над усіма критичними процесами.

Енергетична ефективність є важливим компонентом архітектури IoT-систем безпеки. Багато сенсорів і виконавчих пристроїв працюють автономно, тому оптимізація споживання енергії дозволяє подовжити термін роботи батарей,

зменшити витрати на обслуговування і підтримувати безперервну роботу системи. Сучасні рішення використовують адаптивне опитування сенсорів, регулюючи частоту збору даних залежно від часу доби, активності об’єкта чи історичних шаблонів поведінки, що дозволяє досягати балансу між швидкістю реагування і економією ресурсів.

Системи також передбачають інтеграцію з мобільними застосунками та вебінтерфейсами, що забезпечує користувачам доступ до інформації в режимі реального часу, можливість віддаленого керування, отримання сповіщень і аналітики. Це дозволяє не лише контролювати стан системи, але й здійснювати превентивні заходи, коригувати алгоритми роботи сенсорів та виконавчих пристроїв і приймати обґрунтовані рішення на основі аналітики великих даних.

Для практичної реалізації сучасних IoT-систем безпеки широко використовуються платформи на базі Raspberry Pi у поєднанні з WebIOPi. В таких рішеннях Raspberry Pi виконує роль контролера, який отримує дані від сенсорів, обробляє їх локально і передає в хмару або на вебінтерфейс. WebIOPi забезпечує легкий доступ до керування пристроями через веб-браузер, дозволяючи моніторити стан сенсорів, активувати виконавчі механізми і отримувати сповіщення про критичні події. Така інтеграція демонструє практичне застосування усіх описаних протоколів, технологій, алгоритмів обробки даних і методів захисту інформації, забезпечуючи повноцінну систему безпеки, готову до роботи в реальних умовах.

РОЗДІЛ 2
ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА ВИБІР КОМПОНЕНТІВ СИСТЕМИ

2.1. Поняття та сутність Інтернету речей

Виконуючи дослідження з технологіями Інтернет речей, було досліджено одноплатний комп’ютер за допомогою якого на даний момент часу використовуються кожного дня навіть не помічаючи того. Технології інтернету речей зустрічаються майже в кожному пристрої який є в нашому домі. Кавоварки, охоронні системи, система розумного дому все працює за допомогою інтернет речей.

Інтернет речей структура мережі, що складається з пов’язаних між собою фізичних пристроїв та програмного забезпечення, які мають пристрої прийому та передачі даних, і також спеціальне програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом та комп’ютерами в автономному режимі, за допомогою використання стандартних протоколів зв’язку, та без особливого втручання людини для керування. Також не забуваємо про безпеку, яка є важливим пунктом системи. Є плюси та мінуси безпеки інтернету речей про які ми поговоримо в наступних пунктах [5].

Інтернет речей працює за принципом збору та обробки даних. Збирає дані з фізичних датчиків, передає їх на сервер, та видає користувачу на його спеціально розроблений інтерфейс.

Працювати ми будем з пристроєм інтернету речей Raspberry Pi 3 Model B, який забезпечить нас необхідним функціоналом для виконання даного дослідження.

2.2. Raspberry Pi 3 Model B

Основним інструментом для виконання нашого дослідження є одноплатний комп’ютер Raspberry Pi 3 Model B. За допомогою його ми зможемо скласти систему з фізичних датчиків та запрограмувати їх для виконання певного функціоналу нашої системи.

Одноплатний комп’ютер було розроблено в 2016 році розроблений Raspberry Pi Foundation (рис. 2.1.).

Рисунок 2.1. Одноплатний комп’ютер Raspberry Pi 3 Model b

Апаратна частина досліджуваного пристрою складає:

1. Чотирьох ядерний процесор Broadcom BCM2837 з тактовою частотою 1,2 ГГц, 64-bit.
2. 1 ГБ оперативної пам’яті.
3. Модуль бездротового зв’язку BCM43438 та Bluetooth Low Energy.
4. Базовий ethernet порт зі швидкістю до 100 мбіт в секунду.
5. Контактна група яка складається з 40 pin для підключення GPIO.
6. Чотири USB порти 2-гої генерації.
7. Стереовихід для вводу та виводу звуку.
8. Повнорозмірний HDMI порт.

9. Порт CSI для підключення камери.
10. Порт DSI для підключення сенсорного дисплея.
11. Порт для Micro - SD карти з встановленою операційною системою.
12. Порт Micro USB для живлення до 2.5 А [6].

Даних параметрів вистачає для стабільної не завжди швидкої роботи з даним видом комп'ютера.

2.3. Переваги та недоліки користування одноплатним комп'ютером

Користування одноплатним комп'ютерами Raspberry Pi має переваги, але й без недоліків теж не обійшлось. Вони доступні по ціні, це їхня дуже вагома перевага.

Тож дайте пройдемось по основних пунктах переваг та недолік даного пристрою.

Почнімо з переваг:

- доступна вартість, перевага для багатьох користувачів особливо в даний момент часу ціна даного комп'ютера, користувачі які не мають можливості купити собі ноутбук чи комп'ютер на операційній системі windows або Mac OS, для не важких завдань;
- дані комп'ютери дуже компактні, тому цей фактор є дуже важливий для робочого місця з обмеженим простором;
- наявність модулів бездротового зв'язку, ще один фактор для економії простору та організації робочого місця;
- енергоефективний процесор для виконання не складних завдань, та економному споживанню електроенергії;
- підтримка операційної системи на базі ядра Linux, що забезпечує гнучкість для розробки та не високими вимогами до характеристик комп'ютера;
- розширені можливості для підключення сторонніх пристроїв таких як камери, датчики, реле та інші;

- велика користувацька база, та підтримка яка забезпечить вирішення багатьох проблем самостійно.

А зараз перейдемо до недоліків користування одноплатним комп’ютером:

- обмежена продуктивність пристрою, для базових завдань його вистачає, але коли приходить час для складних обчислень та багатозадачності, то в цьому випадку його потужності вже не вистачає для швидкої та комфортної роботи;
- дуже важливий пункт для кожного комп’ютера сьогодення це обсяг оперативної пам’яті, в нашому випадку 1 гігабайта вже недостатньо для запуску багатьох програм;
- залежність від Micro SD карти яка вже на сьогодні дуже повільний накопичувач для операційної системи н фоні сучасних SSD з майже такими розмірами та значно швидшою роботою;
- проблеми з електроживленням 5V = 2.5A на сьогодні дуже мало, цей фактор також впливає на швидкодію пристрою;
- ускладнює роботу з даним пристроєм перегрів який є дуже частим при довгій та важкій роботі з об’єктом дослідження, неактивне охолодження дуже погано впливає на продуктивність;
- обмеження мережевих пристроїв так як вони працюють на пару з USB через один контролер;
- останнім вагомим недоліком є операційна система Linux для недосвідчених користувачів є певні складнощі експлуатації даного пристрою так як ця система вимагає певних знань для роботи з нею [7].

2.4. LEDіндикатор

LED-індикатор напівпровідниковий електронний компонент, який випромінює світло під час проходження через нього електричного струму. Його основне призначення полягає у візуальному відображенні стану роботи електронного пристрою. Світлодіоди широко застосовуються в побутовій електроніці, промислових системах автоматизації, пристроях Інтернету речей та системах безпеки (рис. 2.2.).

Рисунок 2.2. LEDіндикатор

Принцип роботи LED-індикатора базується на явищі електролюмінесценції. Усередині компонента знаходиться p-n перехід, у якому під час подачі прямої напруги відбувається рекомбінація носіїв заряду, внаслідок чого виділяється енергія у вигляді світлового випромінювання. Колір світіння визначається матеріалом напівпровідника і може бути червоним, зеленим, синім, жовтим або білим.

Конструктивно світлодіод має два виводи: анод і катод. Анод підключається до позитивного полюса джерела живлення, а катод до землі. Для правильного функціонування та захисту від перевищення струму світлодіод підключають через

струмообмежувальний резистор, номінал якого зазвичай становить від 220 до 330 Ом. Робоча напруга більшості світлодіодів знаходиться в межах від 1,8 до 3,3 В залежно від кольору, а типовий робочий струм становить 10–20 мА.

LED-індикатори характеризуються низьким енергоспоживанням, високою яскравістю, швидкою реакцією на зміну електричного сигналу та тривалим терміном служби, який може перевищувати 50 000 годин. Завдяки цим властивостям вони є одним із найпоширеніших елементів для індикації стану електронних систем.

У системі безпеки, побудованій на базі Raspberry Pi, LED-індикатор використовується як засіб візуального оповіщення користувача. У нормальному режимі роботи світлодіод може бути вимкненим або світитися постійно, сигналізуючи про готовність системи. Після спрацювання інфрачервоного датчика перешкоди, наприклад KY-032, світлодіод загоряється або починає блимати, що свідчить про виявлення об’єкта та активацію режиму тривоги.

Таким чином, LED-індикатор є простим, надійним і ефективним компонентом, який забезпечує наочне відображення поточного стану системи безпеки та підвищує зручність її експлуатації.

2.5. Активний звуковий модуль Buzzer

Ще одним важливим елементом системи безпеки на базі Інтернету речей є засіб звукового оповіщення. Для реалізації звукової сигналізації у розробленій системі використовується активний звуковий модуль (Active Buzzer), який забезпечує генерацію гучного звукового сигналу у разі виникнення тривожних подій. Застосування такого модуля дозволяє оперативно привернути увагу користувача до несанкціонованого доступу, спрацювання датчиків або інших критичних ситуацій (рис 2.3.).

Рисунок 2.3. Активний Buzzer

Активний buzzer являє собою електронний компонент, що містить вбудований генератор коливань. На відміну від пасивного бузера, який потребує подачі сигналу певної частоти для створення звуку, активний модуль починає працювати одразу після подачі постійної напруги. Це значно спрощує його використання, оскільки для керування достатньо лише подати логічний сигнал високого рівня на керуючий контакт.

Конструктивно модуль зазвичай складається з п’єзоелектричного випромінювача, транзисторного ключа, генератора звукової частоти та вбудованого струмообмежувального резистора. Більшість модулів типу KY-012 або аналогічних мають три виводи: контакт живлення (VCC), контакт заземлення (GND) та сигнальний вивід (S). При подачі логічної одиниці на сигнальний контакт модуль генерує звуковий сигнал фіксованої частоти, який добре чутно в приміщенні.

Робоча напруга активного бузера зазвичай становить 3,3-5 В, що забезпечує його сумісність із Raspberry Pi 3 Model B. Це дозволяє підключити модуль безпосередньо до GPIO-виводів одноплатного комп’ютера та керувати його роботою програмними засобами. Для підключення контакт VCC з’єднується з шиною живлення, GNDіз загальною землею, а сигнальний контакт, з одним із цифрових GPIO-портів.

Принцип роботи активного бузера полягає у перетворенні електричних коливань, сформованих внутрішнім генератором, у механічні коливання п’єзоелектричного елемента. У результаті створюються звукові хвилі, що сприймаються людським слухом як безперервний тон. Частота звуку визначається параметрами внутрішнього генератора та, як правило, знаходиться в діапазоні від 2 до 4 кГц, що забезпечує достатню гучність і хорошу чутність.

У системі безпеки активний buzzer виконує функцію звукового сповіщення про тривожні ситуації. Наприклад, при спрацюванні датчика руху або відкриття дверей Raspberry Pi активує модуль, який генерує гучний звуковий сигнал. Це дозволяє не лише миттєво інформувати користувача про подію, а й психологічно впливати на потенційного порушника, привертаючи увагу оточуючих.

Завдяки простоті керування можна реалізувати різні режими звукового оповіщення. Бузер може працювати безперервно, подавати короткі переривчасті сигнали або формувати певні послідовності звуків. Наприклад, один короткий сигнал може означати успішний запуск системи, два сигнали - зміну режиму роботи, а довгий безперервний сигнал - стан тривоги.

До основних переваг активного бузера належать простота підключення, низьке енергоспоживання, компактні розміри, доступна вартість та відсутність необхідності генерувати складні сигнали керування. Модуль є надійним і не потребує додаткових налаштувань, що робить його особливо зручним для використання у вбудованих системах та IoT-проєктах.

Недоліком активного бузера є обмежена функціональність, оскільки він відтворює лише звук фіксованої частоти. На відміну від пасивних звукових випромінювачів, активний модуль не дозволяє генерувати мелодії або змінювати тональність сигналу. Однак для реалізації охоронної сигналізації така можливість не є критичною, оскільки основною вимогою є гучне та чітке звукове сповіщення.

Таким чином, активний звуковий модуль є важливою складовою системи безпеки на базі Інтернету речей. Його використання забезпечує ефективне звукове

оповіщення про тривожні події, підвищує оперативність реагування користувача та покращує загальну функціональність розробленого програмно-апаратного комплексу.

2.6. Інфрачервоний бар’єрний датчик перешкоди KY-032

Одним із основних елементів системи безпеки на базі Інтернету речей є засоби виявлення присутності об’єктів у контрольованій зоні. Для реалізації цієї функції у розробленій системі використовується інфрачервоний бар’єрний датчик перешкоди KY-032 (рис. 2.4.). Даний модуль призначений для безконтактного виявлення об’єктів, що знаходяться перед сенсором і широко застосовується у робототехніці, системах автоматизації та охоронних пристроях.

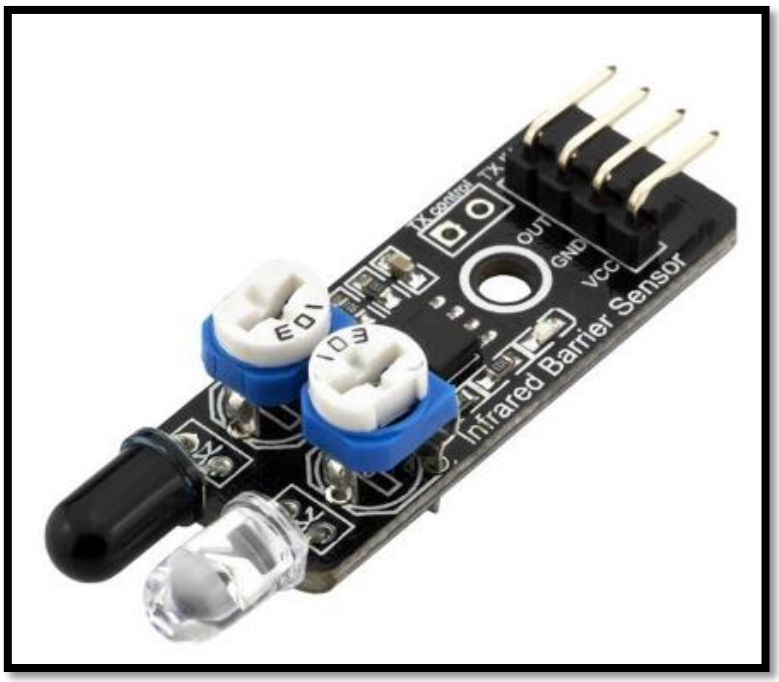

Рисунок 2.4. Інфрачервоний бар’єрний датчик KY-032

Модуль KY-032 працює за принципом відбиття інфрачервоного випромінювання. У його конструкції передбачено два основні елементи: інфрачервоний світлодіод-випромінювач та фотоприймач. Випромінювач постійно генерує інфрачервоне світло, яке є невидимим для людського ока. Якщо перед

датчиком з’являється перешкода, частина випромінювання відбивається від поверхні об’єкта та потрапляє на фотоприймач. На основі цього формується цифровий сигнал, який передається до Raspberry Pi.

Конструктивно модуль має чотири основні виводи: живлення (VCC), заземлення (GND), цифровий вихід (OUT) та контакт EN (Enable), який використовується для вмикання або вимикання випромінювача. Робоча напруга модуля становить 3,3–5 В, що забезпечує його повну сумісність із одноплатним комп’ютером Raspberry Pi 3 Model B. Вихідний сигнал має цифровий характер, тому підключення до GPIO-виводів не потребує додаткових перетворювачів.

Однією з особливостей KY-032 є можливість регулювання параметрів роботи за допомогою двох підлаштувальних резисторів. Перший резистор дозволяє змінювати чутливість датчика, а другий налаштовувати частоту випромінювання. Завдяки цьому можна адаптувати модуль до різних умов експлуатації, зменшити вплив зовнішнього освітлення та підвищити точність спрацьовування.

Дальність виявлення об’єктів зазвичай становить від 2 до 40 см, залежно від кольору, розміру та матеріалу поверхні. Найкраще датчик реагує на світлі та матові об’єкти, які добре відбивають інфрачервоне випромінювання. Темні або блискучі поверхні можуть знижувати ефективність виявлення.

У системі безпеки датчик KY-032 використовується для контролю визначеної зони. При появі стороннього об’єкта або людини перед сенсором Raspberry Pi отримує відповідний сигнал та виконує запрограмовані дії. Наприклад, система може активувати звукову сигналізацію, привести в дію LED-світлодіод, записати подію до журналу або відобразити повідомлення у вебінтерфейсі.

Однією з переваг датчика є швидке спрацьовування та відсутність механічних контактів, що забезпечує високу надійність і тривалий термін експлуатації. Модуль споживає незначний струм, має компактні розміри та легко інтегрується у макетні та готові пристрої. Завдяки цифровому виходу програмна обробка сигналу є простою і не потребує складних алгоритмів.

До основних переваг датчика KY-032 належать:

- безконтактне виявлення об’єктів;
- простота підключення;
- сумісність із Raspberry Pi та Arduino;
- регулювання чутливості;
- низьке енергоспоживання;
- швидка реакція на появу перешкоди;
- доступна вартість.

Серед недоліків можна виділити залежність точності роботи від зовнішнього освітлення та характеристик поверхні об’єкта. Крім того, дальність виявлення є відносно невеликою, тому датчик найбільш ефективний для контролю локальних зон.

Отже, інфрачервоний бар’єрний датчик перешкоди KY-032 є простим і ефективним засобом безконтактного виявлення об’єктів. У складі системи безпеки на базі Інтернету речей він виконує функцію первинного сенсора виявлення присутності, забезпечуючи своєчасне реагування системи на появу сторонніх об’єктів у контрольованій зоні.

Розділ 3
НАЛАШТУВАННЯ ФРЕЙМВОРКУІНТЕРНЕТУ РЕЧЕЙ WEBIOPI HARASPBERRY PI 3 MODEL B

3.1. Вибір програмної платформи для реалізації вебінтерфейсу системи безпеки

У процесі розроблення системи безпеки на базі Інтернету речей одним із ключових етапів є вибір програмної платформи, яка забезпечує взаємодію між апаратною частиною пристрою та користувачем. Оскільки розроблена система передбачає керування датчиками та виконавчими пристроями через мережу, необхідно було обрати програмне середовище, здатне реалізувати вебінтерфейс, забезпечити доступ до GPIO-виводів Raspberry Pi та підтримувати виконання користувацьких скриптів. Для вирішення цього завдання було обрано фреймворк WebIOPi.

WebIOPi є спеціалізованим програмним засобом, призначеним для організації віддаленого керування GPIO-портами одноплатних комп'ютерів сімейства Raspberry Pi через веббраузер. Назва WebIOPi утворена від поєднання слів «Web», «I/O» та «Pi», що підкреслює його основне призначення - забезпечення вебдоступу до інтерфейсів введення та виведення Raspberry Pi. Фреймворк включає вбудований HTTP-сервер, бібліотеки для роботи з GPIO, а також програмний інтерфейс API для створення власних вебзастосунків.

Основною причиною вибору WebIOPi є його орієнтація на проєкти Інтернету речей, у яких необхідно поєднати апаратне керування та мережевий доступ. Фреймворк дозволяє користувачу відкривати вебсторінку у браузері персонального комп'ютера, планшета або смартфона та в режимі реального часу керувати станом GPIO-виводів Raspberry Pi. Це дає змогу здійснювати увімкнення світлодіодів, активацію звукової сигналізації, зчитування стану датчиків руху та виконання інших

операцій без безпосереднього підключення до пристрою. Важливою перевагою WebIOPi є підтримка мови програмування Python. За допомогою Python-скриптів можна реалізувати власну логіку роботи системи, обробку подій, взаємодію з датчиками та формування відповідних команд для виконавчих пристроїв. У межах розробленої системи безпеки Python використовується для керування LED-світлодіодом, активним бузером та іншими електронними компонентами. Завдяки цьому WebIOPi забезпечує не лише вебдоступ до GPIO, а й повноцінне програмне середовище для створення функціональних IoT-рішень [8].

Ще однією важливою особливістю WebIOPi є використання сучасних вебтехнологій. Фреймворк підтримує інтеграцію з HTML, CSS та JavaScript, що дозволяє розробляти зручні та інтуїтивно зрозумілі користувацькі інтерфейси. Наприклад, у межах даного проєкту створено вебсторінку, яка містить кнопки для увімкнення та вимкнення сигналізації, індикатори поточного стану системи та елементи керування периферійними пристроями. Усі дії користувача автоматично передаються на Raspberry Pi, де обробляються відповідними Python-скриптами.

Для обґрунтування вибору WebIOPi доцільно порівняти його з іншими програмними платформами, які можуть використовуватися у подібних проєктах. Серед найбільш поширених альтернатив можна виділити Flask, Node-RED та MQTT-брокери.Flask є легковаговим вебфреймворком для Python, який надає широкі можливості для створення вебзастосунків. Його перевагою є висока гнучкість, однак для роботи з GPIO необхідно самостійно інтегрувати сторонні бібліотеки та реалізовувати механізми обробки HTTP-запитів. Це збільшує складність розробки та вимагає більш глибоких знань програмування.

Node-RED є візуальним середовищем для побудови IoT-систем. Воно дозволяє створювати логіку роботи шляхом з’єднання функціональних блоків без написання великої кількості коду. Проте Node-RED споживає більше системних ресурсів та має складнішу структуру налаштування, що може бути недоцільним для невеликих навчальних проєктів.

На відміну від зазначених альтернатив, WebIOPi поєднує в собі вебсервер, інструменти роботи з GPIO та підтримку Python-скриптів. Це дозволяє швидко створити повноцінну систему керування без необхідності встановлення великої кількості додаткових компонентів. Саме ця інтегрованість стала визначальним фактором при виборі програмної платформи є певні переваги:

- простота встановлення та налаштування;
- підтримка Raspberry Pi та GPIO;
- можливість використання Python-скриптів;
- вбудований HTTP-сервер;
- підтримка HTML, CSS та JavaScript;
- віддалене керування через веббраузер;
- низьке споживання системних ресурсів;
- зручна інтеграція з електронними компонентами.

Серед недоліків фреймворку можна виділити відсутність офіційної активної підтримки та необхідність внесення змін до вихідного коду для забезпечення сумісності з сучасними версіями Raspberry Pi OS і Python 3. Однак ці обмеження не є критичними, оскільки спільнота користувачів надала необхідні інструкції та виправлення для успішного використання фреймворку.

У контексті розробленої системи безпеки WebIOPi забезпечує реалізацію таких функцій:

- дистанційне керування RGB-світлодіодом;
- активація та деактивація активного бузера;
- зчитування стану датчиків;
- відображення інформації у вебінтерфейсі;
- захист доступу за допомогою логіна та пароля;
- автоматичний запуск після завантаження операційної системи.

Використання WebIOPi дозволяє значно спростити процес розробки, оскільки більшість базових механізмів уже реалізовані у складі фреймворку. Розробнику необхідно лише налаштувати конфігураційні файли, створити Python-скрипти та розробити вебінтерфейс відповідно до поставлених вимог. Отже, у результаті аналізу існуючих програмних платформ для реалізації вебінтерфейсу системи безпеки було обрано фреймворк WebIOPi. Його використання обумовлене простотою інтеграції з Raspberry Pi 3 Model B, підтримкою Python, можливістю керування GPIO через веббраузер та наявністю вбудованого вебсервера. Застосування WebIOPi забезпечує ефективну взаємодію між апаратною та програмною частинами системи, що робить його оптимальним рішенням для створення системи безпеки на базі Інтернету речей.

3.2. Підготовка операційної системи Raspberry Pi OS

У процесі розроблення системи безпеки на базі Інтернету речей одним із базових етапів є підготовка програмного середовища, яке забезпечує стабільну роботу всіх апаратних і програмних компонентів. В якості основної операційної системи використовується Raspberry Pi OS, яка оптимізована для роботи з одноплатними комп’ютерами та підтримує всі необхідні інтерфейси, включаючи GPIO, мережу та периферійні пристрої. Апаратною платформою виступає Raspberry Pi 3 Model B.

Встановлення операційної системи починається з підготовки носія інформації а саме карти пам’яті microSD. Саме на неї записується образ системи, який у подальшому використовується як основне середовище завантаження Raspberry Pi. Для забезпечення стабільної роботи рекомендується використовувати карту пам’яті класу не нижче Class 10, що гарантує достатню швидкість читання та запису даних.

Наступним етапом є завантаження образу операційної системи з офіційного джерела. Найбільш доцільним варіантом для подібних IoT-проєктів є Raspberry Pi OS Lite, оскільки дана версія не містить графічного інтерфейсу користувача, що

дозволяє суттєво зменшити навантаження на процесор та оперативну пам’ять. Оскільки керування системою здійснюється через вебінтерфейс, наявність графічного середовища не є необхідною. Після завантаження образу операційної системи виконується його запис на карту пам’яті за допомогою спеціального програмного забезпечення Raspberry Pi Imager. Дана утиліта дозволяє автоматизувати процес встановлення та зменшує ризик помилок під час підготовки носія. Під час запису користувач має можливість попередньо налаштувати основні параметри системи, зокрема ім’я пристрою, дані для підключення до Wi-Fi мережі, часовий пояс та доступ через SSH. Це дозволяє значно спростити подальше адміністрування системи, оскільки після першого запуску пристрій уже готовий до віддаленого керування. Після завершення запису карта пам’яті встановлюється у Raspberry Pi, після чого пристрій підключається до джерела живлення. Перший запуск системи займає певний час, оскільки відбувається ініціалізація файлової системи та базових служб операційної системи. Якщо попередньо було активовано SSH, пристрій одразу стає доступним у локальній мережі без необхідності підключення монітора або клавіатури.

Подальша робота з системою здійснюється дистанційно за допомогою протоколу SSH. Це дозволяє виконувати всі налаштування через термінал, підключаючись до Raspberry Pi з персонального комп’ютера. Такий підхід є типовим для серверних та IoT-рішень, оскільки забезпечує гнучкість і зручність адміністрування. Після першого входу в систему обов’язковим є оновлення програмного забезпечення. Це включає оновлення списку пакетів та встановлення актуальних версій системних компонентів. Виконання цього етапу є важливим для забезпечення стабільної роботи операційної системи та сумісності з майбутніми програмними модулями, зокрема WebIOPi.

Далі встановлюються базові інструменти розробника, серед яких інтерпретатор Python, менеджер пакетів pip, а також системні бібліотеки, необхідні для компіляції додаткових компонентів. Оскільки WebIOPi базується на Python та використовує

взаємодію з GPIO, наявність цих компонентів є обов’язковою умовою його коректної роботи.Наступним важливим етапом є налаштування системних параметрів через конфігураційне меню Raspberry Pi. У цьому меню здійснюється зміна пароля користувача, налаштування регіону, перевірка мережевих параметрів та активація SSH. У разі необхідності також можуть бути увімкнені додаткові інтерфейси, такі як SPI або I2C, які використовуються для підключення зовнішніх датчиків та модулів. Особлива увага приділяється перевірці роботи GPIO-виводів. Це необхідно для підтвердження того, що операційна система коректно розпізнає апаратну частину пристрою. На цьому етапі може використовуватися як системна утиліта для роботи з пін-контролером, так і прості тестові Python-скрипти, які перевіряють можливість керування вихідними сигналами.

Завершальним етапом підготовки є створення робочого каталогу проєкту. У цьому каталозі надалі зберігатимуться всі файли, пов’язані з реалізацією системи безпеки: конфігурації WebIOPi, Python-скрипти для керування пристроями та файли вебінтерфейсу. Такий підхід дозволяє структурувати проєкт і спрощує подальшу розробку та супровід системи. Таким чином, після виконання всіх описаних дій операційна система Raspberry Pi OS повністю готова до встановлення фреймворку WebIOPi та подальшої реалізації вебінтерфейсу системи безпеки. Проведена підготовка забезпечує стабільну роботу апаратної платформи, коректну взаємодію з GPIO та можливість віддаленого керування пристроєм через мережу.

3.3. Встановлення та адаптація WebIOPi для роботи з Raspberry Pi 3 Model B

Однією з ключових переваг використання Raspberry Pi є можливість віддаленого керування пристроєм через мережу. Завдяки компактним розмірам і низькому енергоспоживанню, Raspberry Pi можна розміщувати практично в будь-якому місці, де є живлення та доступ до Інтернету. Це робить його зручним

рішенням для реалізації файлових серверів, медіацентрів, систем автоматизації та, найголовніше, IoT-пристроїв з віддаленим керуванням.

У даному проєкті для реалізації віддаленого керування використовується фреймворк WebIOPi, який є одним із найзручніших інструментів для роботи з GPIO на Raspberry Pi через вебінтерфейс. Його можна розглядати як універсальний інструмент, що дозволяє керувати входами та виходами мікрокомп'ютера безпосередньо з браузера або сторонніх додатків [9].

WebIOPi створений спеціально для роботи з GPIO-пінами Raspberry Pi, а також для взаємодії з датчиками та виконавчими пристроями. Основною ідеєю фреймворку є забезпечення простого доступу до апаратної частини пристрою через вебтехнології (рис. 3.1.).

Рисунок 3.1. Схема управління контактною групою GPIO через вебінтерейс

Фреймворк WebIOPi має широкий набір функцій, які роблять його зручним для IoT-проєктів:

- підтримка віддаленого доступу до Raspberry Pi через Інтернет (у деяких версіях через Weaved IoT Kit);
- реалізація на мові Python, що дозволяє легко створювати власні скрипти керування пристроями;
- підтримка інтерфейсів Serial, SPI та I2C для підключення зовнішніх модулів;

- можливість роботи з різними датчиками, аналого-цифровими перетворювачами та розширювачами портів;
- наявність JavaScript/HTML бібліотек для створення власного вебінтерфейсу;
- підтримка Python та Java клієнтів для взаємодії між пристроями;
- можливість використання протоколу CoAP, який є сучасним рішенням для IoT-систем;
- вбудовані інструменти для тестування та налагодження GPIO.

Завдяки цьому WebIOPi часто використовуються як базова платформа для навчальних і прототипних IoT-систем.

Перед встановленням фреймворку необхідно завантажити архів WebIOPi у форматі .tar.gz у домашню директорію користувача Raspberry Pi. Після цього виконується його розпакування та запуск інсталяційного скрипта. Процес встановлення включає автоматичне завантаження необхідних залежностей через менеджер пакетів apt-get. Перед інсталяцією рекомендується зупинити вже запущені служби WebIOPi, якщо вони існують. Це дозволяє уникнути конфліктів під час оновлення або перевстановлення системи. Після завантаження архіву виконуються команди розпакування та встановлення. Інсталятор автоматично налаштовує систему та встановлює необхідні бібліотеки. Для деяких моделей Raspberry Pi (зокрема Pi 2, Pi 3 та B+) рекомендується використовувати патч, який відкриває доступ до всіх 40 GPIO-виводів. Це забезпечує повну сумісність із сучасними платами [10]. Після встановлення фреймворк запускається за допомогою команди в терміналі. Під час запуску можна вказувати додаткові параметри, такі як конфігураційний файл, рівень логування або скрипти користувача. Наприклад, стандартний запуск може виглядати як запуск із режимом налагодження та використанням стандартного конфігураційного файлу. Після запуску WebIOPi піднімає локальний вебсервер, через який здійснюється доступ до GPIO. Для зручності використання WebIOPi може працювати у фоновому режимі як системна служба. У цьому випадку він

автоматично запускається при старті операційної системи та не потребує ручного запуску через термінал.

Керування службою здійснюється стандартними командами запуску та зупинки сервісів у Linux-системах. Це особливо зручно для IoT-пристроїв, які повинні працювати автономно без постійного втручання користувача. Для забезпечення повної автономності системи WebIOPi можна налаштувати таким чином, щоб він автоматично запускався після увімкнення Raspberry Pi. Це дозволяє використовувати пристрій як постійно працюючий сервер керування. У разі необхідності автоматичний запуск можна також вимкнути, щоб керувати сервісом вручну. Після запуску WebIOPi стає доступним через локальну мережу. Для підключення достатньо відкрити браузер і ввести IP-адресу Raspberry Pi з відповідним портом. За замовчуванням використовується порт 8000. У браузері відкривається вебінтерфейс, через який можна керувати GPIO-виводами. Для входу використовується стандартна авторизація, яка передбачає логін та пароль. Після входу користувач отримує доступ до панелі керування, де можна змінювати стан GPIO.

У вебінтерфейсі WebIOPi користувач може переглядати схему GPIO Raspberry Pi у вигляді, аналогічному фізичному розташуванню контактів на платі. Це значно спрощує роботу, оскільки дозволяє інтуїтивно керувати виходами (рис. 3.2.).

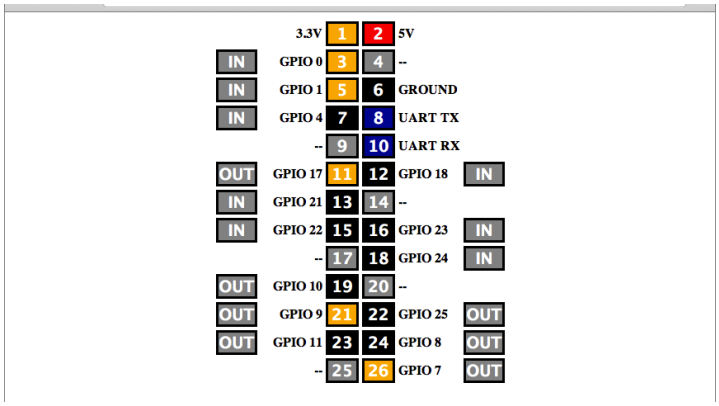

Рисунок 3.2. Керування GPIO через вебінтерфейс

3.4. Забезпечення сумісності WebIOPi із сучасними версіями Raspberry Pi OS

Фреймворк WebIOPi був розроблений для попередніх версій операційної системи Raspberry Pi OS та інтерпретатора Python. Оскільки офіційна підтримка проєкту була припинена, під час встановлення на сучасні версії операційної системи можуть виникати помилки, пов’язані з несумісністю окремих програмних компонентів. Тому перед початком практичного використання необхідно виконати низку додаткових налаштувань, що забезпечують коректну роботу фреймворку.

Однією з найпоширеніших проблем є зміна структури стандартних бібліотек Python. У нових версіях Python деякі функції були перенесені або перейменовані, унаслідок чого інсталяційні скрипти WebIOPi можуть завершуватися з помилками. Для усунення цієї проблеми виконується редагування окремих файлів фреймворку та заміна застарілих конструкцій на сучасні аналоги. Після внесення змін інсталяція проходить успішно, а основні модулі WebIOPi стають придатними для використання.

Ще однією важливою умовою є забезпечення доступу до GPIO-портів одноплатного комп’ютера Raspberry Pi 3 Model B. Для роботи з апаратними контактами користувач, від імені якого запускається WebIOPi, повинен мати відповідні права доступу. У сучасних версіях Raspberry Pi OS це досягається шляхом додавання користувача до системної групи gpio. Після перезавантаження операційної системи WebIOPi отримує можливість керувати станом цифрових виводів без додаткових привілеїв.

Також важливо враховувати зміни в механізмах запуску системних служб. У сучасних редакціях Raspberry Pi OS використовується система ініціалізації systemd, яка відповідає за автоматичний запуск програм під час завантаження операційної системи. Для забезпечення стабільної роботи WebIOPi налаштовується як окремий

сервіс, що запускається автоматично після старту системи та відновлює роботу після перезавантаження пристрою.

Під час практичного використання можуть виникати проблеми, пов’язані із зайнятими мережевими портами, неправильними параметрами конфігураційного файлу або відсутністю доступу до каталогів із вебсторінками. Для їх діагностики застосовуються системні журнали та консольні повідомлення. Аналіз цих повідомлень дозволяє швидко виявити причину несправності та усунути її без перевстановлення програмного забезпечення.

Окрему увагу необхідно приділити безпеці роботи вебсервера. Стандартні облікові дані доступу до WebIOPi повинні бути замінені на власні, а також рекомендовано обмежити доступ до пристрою лише в межах локальної мережі або через захищені канали зв’язку. Це дозволяє запобігти несанкціонованому керуванню GPIO-портами та підвищує загальну надійність системи безпеки.

Таким чином, забезпечення сумісності WebIOPi із сучасними версіями Raspberry Pi OS включає усунення програмних несумісностей, налаштування прав доступу до GPIO, організацію автоматичного запуску сервісу та впровадження базових заходів інформаційної безпеки. Виконання зазначених дій забезпечує стабільну роботу фреймворку та створює надійну основу для подальшої розробки програмних модулів системи безпеки на базі технологій Інтернету речей.

РОЗДІЛ 4
РОЗРОБКА ТА РЕАЛІЗАЦІЯ ВЕБІНТЕРФЕЙСУ СИСТЕМИ БЕЗПЕКИ НА БАЗІ WEBIOPI

4.1. Загальна характеристика та архітектура

Практична реалізація системи безпеки спрямована на створення гнучкого та відмовостійкого рішення, яке інтегрує фізичні сенсори моніторингу навколишнього середовища з сучасними вебтехнологіями. В основі розробки лежить концепція «Інтернету речей» (IoT), де кожен апаратний компонент є частиною єдиної екосистеми, керованої через стандартизовані протоколи передачі даних.

Вибір мікрокомп'ютера Raspberry Pi як обчислювального центру обумовлений його високою продуктивністю, наявністю розширеного інтерфейсу GPIO та повноцінної операційної системи на базі ядра Linux. Це дозволяє використовувати високорівневі мови програмування та готові фреймворки для побудови серверної частини, що значно скорочує цикл розробки та підвищує стабільність роботи ПЗ.

Архітектура системи побудована за модульним принципом і складається з трьох ключових рівнів:

- апаратний рівень охоплює фізичні пристрої, інфрачервоний датчик детекції перешкод, світлодіодний індикатор статусу та активний зумер звукового оповіщення. Цей рівень відповідає за безпосередню взаємодію з фізичним світом.
- рівень управління та сервісів базується на фреймворку WebIOPi. Він виконує роль сполучної ланки, яка перетворює низькорівневі сигнали від датчиків у структуровані дані та надає REST API для доступу до них. На цьому рівні реалізовано основну логіку обробки подій та макроси керування.
- рівень представлення реалізований у вигляді адаптивного вебінтерфейсу. Він забезпечує візуалізацію стану системи в реальному часі та надає

користувачеві інструменти для дистанційного керування модулями безпеки через будь-який сучасний браузер.

Така структура забезпечує масштабованість проекту: за потреби систему можна доповнити новими датчиками або змінити логіку оповіщення без кардинальної перебудови всього програмного комплексу. Основний акцент при реалізації було зроблено на забезпеченні автономності роботи серверної частини, що гарантує спрацювання сигналізації навіть за умови відсутності активного клієнтського підключення до вебінтерфейсу.

4.2. Схема підключення апаратних компонентів

Процес розробки системи безпеки на апаратному рівні розпочинається з детального проектування архітектури з'єднань, де центральним інтегруючим вузлом виступає мікрокомп'ютер Raspberry Pi. Фізична реалізація системи базується на використанні інтерфейсу GPIO, який виступає посередником між обчислювальним ядром та зовнішнім середовищем. При формуванні апаратної частини було застосовано комплексний підхід, що передбачає поділ компонентів на вузли зчитування інформації та вузли виконавчої реакції.

Фундаментальним елементом підсистеми детекції є інфрачервоний датчик перешкод KY-032. Його інтеграція в загальну схему вимагала особливої уваги до рівнів напруги, оскільки Raspberry Pi є чутливим до перевантажень на вхідних каскадах. Робота датчика ґрунтується на циклічному випромінюванні інфрачервоних хвиль, які, відбиваючись від потенційного об'єкта, потрапляють на фотоприймач. Внутрішня схемотехніка датчика, що включає потенціометр та компаратор, дозволяє проводити апаратне налаштування чутливості безпосередньо на самому модулі, що забезпечує гнучкість системи при її розгортанні в різних типах приміщень.

Для забезпечення зворотного зв'язку з користувачем у схему інтегровано модулі світлового та звукового сповіщення. Світлодіодний індикатор виконує роль

візуального маркера стану системи. Його підключення до лінії GPIO 14 було реалізовано з використанням послідовного резистивного захисту. Включення резистора номіналом 220 Ом у розрив анодного ланцюга дозволило стабілізувати робочий струм, запобігаючи деградації напівпровідникового кристала світлодіода та захищаючи вихідний транзисторний ключ мікрокомп'ютера від короткого замикання. Паралельно з візуальним каналом функціонує канал акустичного впливу, представлений активним зумером.

Організація ліній живлення та сигнальних магістралей відображена у таблиці 4.1, де зафіксовано чіткий розподіл контактів. Важливою особливістю проектування стало використання роздільних ліній живлення: датчик KY-032 під'єднано до шини 3.3В для забезпечення логічної сумісності сигналів, тоді як активний зумер підключено до шини 5В, що дозволило досягти максимальної амплітуди звукового сигналу при спрацюванні тривоги.

Загальна топологія з'єднань, представлена на відповідних графічних матеріалах (рис 4.1.), демонструє замкнену структуру системи.

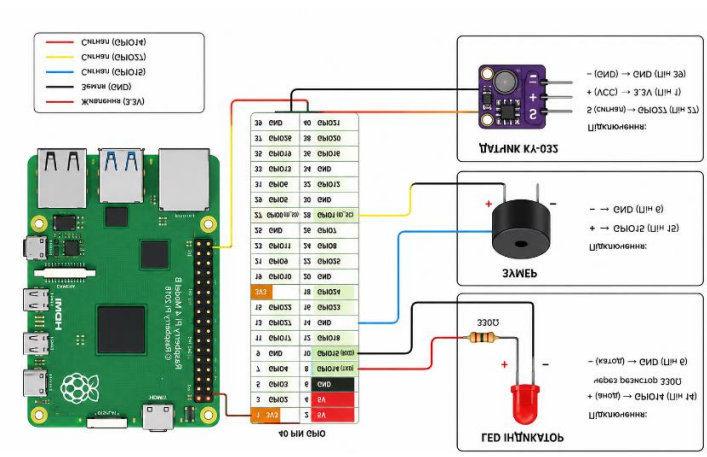

Рисунок 4.1. Схема підключень системи безпеки

Взаємодія компонентів на фізичному рівні відбувається за принципом зміни логічних станів. У стані очікування датчик утримує на сигнальній лінії високий рівень напруги. При появі об'єкта в зоні моніторингу відбувається замикання внутрішнього ключа датчика на «землю», що призводить до падіння напруги на відповідному вході Raspberry Pi. Ця зміна фізичного стану стає тригером для

активації вихідних ланцюгів, що подають напругу на світлодіод та зумер, ініціюючи режим тривоги. Такий підхід до апаратної реалізації забезпечує мінімальні затримки між виявленням загрози та активацією засобів оповіщення.

4.3. Алгоритмічне забезпечення та логіка функціонування системи

Розробка алгоритмічного забезпечення є критично важливим етапом, оскільки саме програмна логіка визначає інтелектуальні можливості системи безпеки та її здатність адекватно реагувати на зовнішні події. При проектуванні алгоритму основна увага приділялася забезпеченню безперервності моніторингу та створенню чіткої ієрархії взаємодії між вхідним сенсором та виконавчими пристроями. Логічна структура системи побудована таким чином, щоб мінімізувати час затримки між фіксацією порушення периметра та активацією тривожних сповіщувачів.

Функціонування системи базується на принципі циклічного опитування стану цифрового входу мікрокомп’ютера. На програмному рівні це реалізовано через безперервний цикл, у якому центральний обчислювальний модуль звертається до інтерфейсу GPIO 27 для зчитування поточного потенціалу. Оскільки інфрачервоний датчик KY-032 працює за принципом інвертованої логіки, алгоритм налаштовано на ідентифікацію низького рівня сигналу як тригерної події. Такий підхід дозволяє підвищити надійність системи: у разі обриву лінії зв’язку або виходу датчика з ладу, система зможе зафіксувати аномальну зміну сигналу, що є стандартною практикою при побудові професійних охоронних рішень.

Окремим аспектом логічної побудови є впровадження механізму програмної фільтрації завад. Враховуючи специфіку роботи інфрачервоних сенсорів, які можуть демонструвати короткочасні сплески або хибні спрацювання через зміну освітленості чи електромагнітні наведення, алгоритм включає етап верифікації загрози. Це реалізується шляхом повторного зчитування стану порту через мінімальний часовий інтервал після первинної фіксації «логічного нуля». Тільки за

умови підтвердження низького рівня сигналу в обох ітераціях, система ініціює перехід до фази активного оповіщення.

Реакція системи на підтверджену загрозу має комплексний характер. Алгоритм передбачає одночасну зміну логічних станів на вихідних портах GPIO 14 та GPIO 15. Подача високого рівня напруги на ці виходи активує світлодіодний індикатор та звуковий зумер, створюючи комбінований ефект психологічного та інформаційного впливу. Паралельно з фізичним оповіщенням, програмне середовище оновлює внутрішні змінні статусу, які стають доступними для зчитування через веб-інтерфейс. Це забезпечує синхронізацію реального стану об’єкта з його візуальним представленням у браузері користувача.

Завершальним етапом логічного циклу є процедура деактивації тривоги. У розробленому алгоритмі реалізовано принцип примусового скидання, згідно з яким система не повертається у стан спокою автоматично після зникнення об’єкта з поля зору датчика. Таке рішення гарантує, що власник системи обов’язково отримає інформацію про інцидент, навіть якщо він був короткочасним. Вихід із режиму тривоги та переведення виконавчих пристроїв у низький логічний рівень здійснюється лише після отримання авторизованої команди від користувача через панель керування. Такий підхід дозволяє зберігати контроль над ситуацією до моменту повного з’ясування причин спрацювання охоронного модуля.

4.4. Технічна апробація та результати розробки

Фінальним етапом роботи стало функціональне тестування зібраного комплексу та верифікація його працездатності в реальних умовах. Результатом розробки є автономний пристрій моніторингу, що поєднує в собі компактність апаратної частини та наочність програмного інтерфейсу. На розробленому фізичному прототипі (рис. 4.2.) продемонстровано компактне розміщення

компонентів. Основний обчислювальний модуль Raspberry Pi забезпечує живлення та обробку сигналів від датчика KY-032. Завдяки використанню коротких з’єднувальних провідників типу «мама-мама», вдалося досягти високої надійності контактів та мінімізувати габарити пристрою. Датчик встановлено таким чином, щоб інфрачервоний випромінювач та приймач мали пряму видимість охоронної зони, що є критичним для коректного відстеження об’єктів.

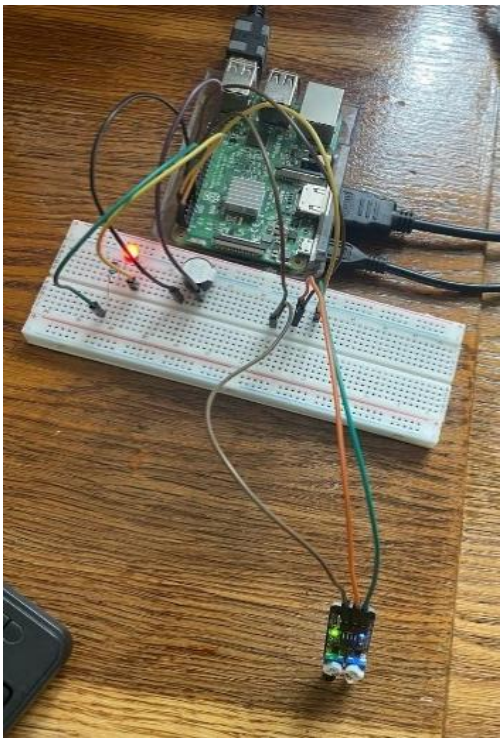

Рисунок 4.2. Монтажна схема проєкту

Клієнтська частина системи, доступ до якої здійснюється через веббраузер, виступає головним інструментом візуалізації. На відміну від стандартних консольних додатків, розроблений інтерфейс дозволяє спостерігати за станом об’єкта дистанційно.

При входженні об’єкта в зону дії датчика, центральний графічний елемент на сторінці миттєво змінює свій стан. Це реалізовано за допомогою асинхронних запитів, що дозволяє спостерігати за динамікою подій без затримок. Окрім колірної зміни, інтерфейс виводить чітке повідомлення про статус системи («у стані спокою» або «тривога»), що робить моніторинг інтуїтивно зрозумілим навіть для

недосвідченого користувача. Статус з’єднання: Система відображає актуальність підключення до сервера WebIOPi, підтверджуючи, що дані, які бачить користувач, є актуальними на поточний момент (рис. 4.3.).

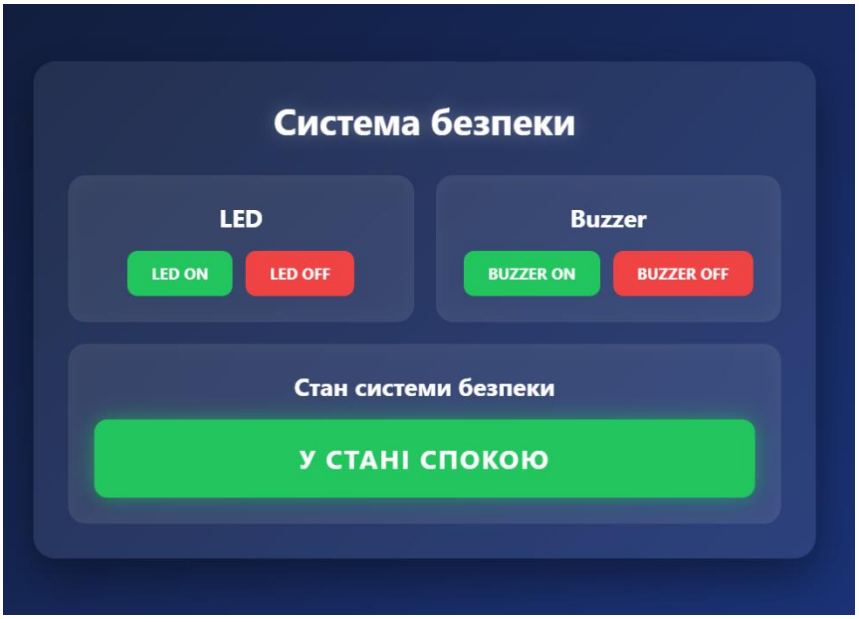

Рисунок 4.3. Графічний інтерфейс системи у стані спокою

Експериментальна перевірка підтвердила, що розроблений програмно-апаратний комплекс повністю відповідає поставленим завданням. Час відгуку системи від моменту фізичного перетину променя датчика до візуальної зміни статусу на екрані становить менше 0.5 секунди, що є відмінним показником для систем моніторингу на базі HTTP-протоколу. Відсутність локальних звукових сигналів робить роботу пристрою конфіденційною, що дозволяє використовувати його для непомітного спостереження за приміщенням.

ВИСНОВКИ

У процесі дослідження було досягнуто значних результатів у дослідженні архітектури IoT-систем, опануванні сучасних інструментальних засобів та розробці практичного рішення для автоматизації безпеки об’єктів. Проведено ґрунтовний аналіз теоретичних засад функціонування систем безпеки в концепції Інтернету речей. Це дозволило сформувати цілісне розуміння методів виявлення загроз, принципів побудови багаторівневих систем захисту та актуальних викликів, пов’язаних із вразливістю мережевих пристроїв.

Опановано функціональні можливості фреймворка WebIOPi, який було обрано як базове середовище для розробки вебінтерфейсу. Досліджено механізми взаємодії фреймворка з портами GPIO одноплатного комп’ютера Raspberry Pi, що дозволило реалізувати дистанційне керування охоронними модулями через протокол HTTP.

Досліджено апаратну базу та програмне забезпечення мікрокомп’ютера Raspberry Pi 3 Model B. Налаштовано операційну систему Raspberry Pi OS та розроблено програмні сценарії на мові Python. Це забезпечило обробку сигналів від датчиків руху, інфрачервоних бар’єрів, сформувавши ядро системи безпеки.

Спроєктовано та реалізовано повнофункціональний вебінтерфейс для моніторингу стану безпеки в реальному часі. Інтерфейс дозволяє користувачеві віддалено відстежувати активність датчиків, керувати звуковим та світловим оповіщенням, а також контролювати доступ до об’єкта. Завдяки застосуванню принципів адаптивного дизайну (HTML/CSS/JS), забезпечено коректне відображення панелі керування на різних типах пристроїв від ПК до смартфонів.

Розроблена система безпеки на базі WebIOPi є економічно ефективним, гнучким та масштабованим рішенням. Вона доводить можливість створення надійних охоронних комплексів з відкритим кодом, які можуть бути адаптовані до специфічних потреб користувачів та інтегровані в ширші екосистеми автоматизації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Якими бувають системи безпеки. URL: https://fonts.googleapis.com/css2?family=Rubik:ital,wght@0,300;0,400;0,500;0,700;0,900;1,300;1,400;1,500;1,700;1,900&display=swap (дата звернення: 27.09.2025).
2. Загрози кібербезпеки і системи контролю доступу - Worldvision.com.ua. URL: https://worldvision.com.ua/articles/ugrozi-kiberbezopasnosti-i-sistemi-kontrolya-dostupa (дата звернення 14.10.2025).
3. Безпека інтернету речей – Вікіпедія. URL: https://uk.wikipedia.org/w/index.php?title=Безпека_інтернету_речей&oldid=44084420 (дата звернення 23.10.2025).
4. Why is lack of standardization a problem with IoT. URL: https://www.linkedin.com/pulse/why-lack-standardization-problem-iot-volkmar-kunerth/ (дата звернення 17.11.2025).
5. Головні ризики безпеки Інтернету речей у 2025 році та як від них захиститися.URL: https://www.proxis.ua/uk/show-article/786/(дата звернення 09.12.2025).
6. Raspberry Pi URL. https://www.raspberrypi.com/products/raspberry-pi-3-model-b/ (дата звернення 19.01.2026).
7. Мікрокомп’ютер Raspberry Pi – інструмент дослідника: Могильний С.Б. Київ 2014 343ст. (дата звернення15.02.2026).
8. The Raspberry Pi Internet of Things Toolkit - Now in two flavors. URL: https://webiopi.trouch.com/ (дата звернення 11.03.2026).
9. Використання веб-інтерфейсу для дистанційного керування Raspberry Pi. Заняття 13-14. mikrotik.kpi.ua. 2015. URL: https://mikrotik.kpi.ua/index.php/courses-list/category-raspberry/76-using-the-web-interface-for-remote-control-raspberry-pi-session-13-14 (дата звернення 24.04.2026).
10. Installation - WebIOPi – trouch. URL: https://webiopi.trouch.com/INSTALL.html (дата звернення 02.05.2026).

ДОДАТОК А

```
<!DOCTYPE html>
<html>
<head>
<title>IoT Security System</title>
<script src="/webiopi.js"></script>
</head>
<body>
<h1>IoT Security System</h1>

<!-- ================= MANUAL CONTROL ================ -->
<h2>Manual Control</h2>
<h3>LED</h3>
<button onclick="ledOn()">LED ON</button>
<button onclick="ledOff()">LED OFF</button>
<p>Status: <span id="ledStatus">OFF</span></p>
<h3>Buzzer</h3>
<button onclick="buzzerOn()">BUZZ ON</button>
<button onclick="buzzerOff()">BUZZ OFF</button>
<p>Status: <span id="buzzerStatus">OFF</span></p>
<hr>

<!-- ================= SENSOR ================ -->
<h2>Sensor (Auto Mode)</h2>
<p>KY-032 Status: <span id="sensorStatus">UNKNOWN</span></p>

<script>
	let ledState = false;
	let buzzerState = false;
	webiopi().ready(function() {
	  // LED
	  webiopi().setFunction(14, "out");
	  // Buzzer
	  webiopi().setFunction(15, "out");
	  webiopi().setFunction(15, "out");
	  // Sensor
	  webiopi().setFunction(18, "in");

	  setInterval(readSensor, 300);
	});

	// ================= MANUAL LED ================
	function ledOn() {
	  ledState = true;
	  webiopi().digitalWrite(14, 1);
	  document.getElementById("ledStatus").innerHTML = "ON";
	}
	function ledOff() {
```

```
    ledState = false;
    webiopi().digitalWrite(14, 0);
    document.getElementById("ledStatus").innerHTML = "OFF";
}

// ================= MANUAL BUZZER ================
function buzzerOn() {
    buzzerState = true;
    webiopi().digitalWrite(15, 1);
    webiopi().digitalWrite(15, 1);
    document.getElementById("buzzerStatus").innerHTML = "ON";
}
function buzzerOff() {
    buzzerState = false;
    webiopi().digitalWrite(15, 0);
    webiopi().digitalWrite(15, 0);
    document.getElementById("buzzerStatus").innerHTML = "OFF";
}
// ================= SENSOR LOGIC ================
function readSensor() {

    webiopi().digitalRead(21, function(value) {

        let obstacle = (value == 0);

        if (obstacle) {
            document.getElementById("sensorStatus").innerHTML =
                "⚠ OBSTACLE DETECTED";

            // ● AUTO OVERRIDE (тільки якщо вручну не вимкнено)
            if (!ledState) {
                webiopi().digitalWrite(14, 1);
            }

            if (!buzzerState) {
                webiopi().digitalWrite(15, 1);
                webiopi().digitalWrite(15, 1);
            }

        } else {
            document.getElementById("sensorStatus").innerHTML =
                "✓ CLEAR";

            // якщо не ручний режим – вимикаємо
            if (!ledState) {
                webiopi().digitalWrite(14, 0);
            }
```

```
                if (!buzzerState) {
                    webiopi().digitalWrite(15, 0);
                    webiopi().digitalWrite(15, 0);
                }
            }
        });
    }
</script>

</body>
</html>
```