

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМ. С. З. ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему:

«ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ РОЗУМНОГО БУДИНКУ З УРАХУВАННЯМ ПОТРЕБ ЛЮДЕЙ З ІНВАЛІДНІСТЮ»

Виконав: здобувач групи Іт-62
спеціальності 126 «Інформаційні системи та
технології»

_____ Іванило М. О.

(прізвище та ініціали)

Керівник: _____ Пташник В. В.

(прізвище та ініціали)

ДУБЛЯНИ-2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМ. С. З. ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)
д.т.н., професор, Тригуба А. М.
(вч. звання, прізвище, ініціали)
“ ” 202 року

З А В Д А Н Н Я НА КВАЛІФІКАЦІЙНУ РОБОТУ

Іванилу Мар'яну Осиповичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Проектування інформаційної системи розумного будинку з урахуванням потреб людей з інвалідністю»

керівник роботи к. т. н., доцент, Пташник В. В.

(наук.ступінь, вч. звання, прізвище, ініціали)

затверджені наказом Львівського НУП від 28.02.2025 року № 140/к-с.

2. Строк подання студентом роботи 8 грудня 2025 року

3. Вихідні дані до роботи: технічні характеристики сенсорних і виконавчих модулів «розумного будинку», документація до протоколів зв'язку, специфікації контролерів і модулів радіообміну, а також нормативні вимоги з безпеки, кіберзахисту, захисту даних і доступності інтерфейсів.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Вступ

1. Аналітичне дослідження методів і принципів проектування інформаційної системи розумного будинку з урахуванням потреб людей з інвалідністю

2. Аналіз технічних засобів та схем керування системами розумного будинку з урахуванням потреб людей з інвалідністю

3. Оптимізація складу та вибір схеми керування інформаційною системою розумного будинку для людей з інвалідністю

4. Охорона праці та безпека в надзвичайних ситуаціях

5. Економічна ефективність

Висновки

Список використаних джерел

5. Перелік графічного матеріалу

Графічний матеріал подається у вигляді презентації

6. Консультанти розділів

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата		Відмітка про виконання
		завдання видав	завдання прийняв	
1, 2, 3, 5	<i>Пташник В. В., к.т.н., доцент</i>			
4	<i>Городецький І. М., к.т.н., доцент</i>			

7. Дата видачі завдання 3 березня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Відмітка про виконання
1	<i>Складання інженерної характеристики системи розумного будинку з урахуванням потреб людей з інвалідністю</i>	<i>03.03.2025 – 15.04.2025</i>	
2	<i>Аналіз технічних засобів та схем керування системами розумного будинку з урахуванням потреб людей з інвалідністю</i>	<i>16.04.2025 – 31.05.2025</i>	
3	<i>Оптимізація складу та вибір схеми керування інформаційною системою розумного будинку для людей з інвалідністю</i>	<i>01.06.2025 – 30.09.2025</i>	
4	<i>Розгляд питань з охорони праці та безпеки у надзвичайних ситуаціях</i>	<i>01.10.2025 – 20.10.2025</i>	
5	<i>Оцінка економічної ефективності прийнятих рішень</i>	<i>21.10.2025 – 15.11.2025</i>	
6	<i>Завершення оформлення розрахунково-пояснювальної записки та презентаційного матеріалу</i>	<i>16.11.2025 – 30.11.2025</i>	
7	<i>Завершення роботи в цілому. Підготовка до захисту кваліфікаційної роботи</i>	<i>01.06.2025 – 08.12.2025</i>	

Здобувач

Іванило М. О.
(підпис) (прізвище та ініціали)

Керівник роботи

Пташник В. В.
(підпис) (прізвище та ініціали)

УДК 681.521 / 681.518

Проектування інформаційної системи розумного будинку з урахуванням потреб людей з інвалідністю. Іванило М. О. Кафедра інформаційних технологій – Дубляни, Львівський НУВМБ ім. С.З. Гжицького, 2025.

Кваліфікаційна робота: 88 сторінок текстової частини, 25 рисунків, 20 таблиць, 17 джерел літератури, 2 додатки

Мета кваліфікаційної роботи полягає у проектуванні інформаційної системи «розумного будинку», що забезпечує підвищення доступності, безпеки та комфортності житлового середовища для людей з інвалідністю шляхом інтеграції сенсорних пристроїв, виконавчих модулів і адаптивних інтерфейсів керування.

Об'єктом дослідження є процеси автоматизованого моніторингу та керування підсистемами житлового помешкання (освітлення, клімат, безпека, оповіщення) з урахуванням потреб користувачів з різними функціональними обмеженнями.

Предмет дослідження вивчає інформаційні технології та програмно-апаратні засоби, що забезпечують автоматизований моніторинг, керування та адаптивну взаємодію користувача з підсистемами «розумного будинку» з урахуванням потреб людей з інвалідністю.

У кваліфікаційній роботі розглянуто проектування інформаційних систем «розумного будинку», орієнтованих на доступність і безпеку житла. Проаналізовано архітектуру IoT-інфраструктури та визначено вимоги до системи з позицій універсального дизайну. Запропоновано структуру системи, інформаційні потоки та сценарії автоматизації з урахуванням потреб користувачів. Окремо враховано адаптацію інтерфейсів і оцінено надійність роботи в типових та позаштатних ситуаціях.

Ключові слова: інформаційна система, розумний будинок, IoT, люди з інвалідністю, доступність, універсальний дизайн, адаптивний інтерфейс.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІТИЧНЕ ДОСЛІДЖЕННЯ МЕТОДІВ І ПРИНЦИПІВ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ РОЗУМНОГО БУДИНКУ З УРАХУВАННЯМ ПОТРЕБ ЛЮДЕЙ З ІНВАЛІДНІСТЮ	8
1.1 Поняття, структура та типова класифікація систем «розумний будинок» .	8
1.2 Принципи універсального дизайну та доступності середовища для людей з інвалідністю	14
1.3 Методи проєктування інформаційних систем підтримки доступного житла	18
1.4 Огляд нормативно-правових, технічних та ергономічних вимог до систем «розумний будинок» для людей з інвалідністю	22
РОЗДІЛ 2 АНАЛІЗ ТЕХНІЧНИХ ЗАСОБІВ ТА СХЕМ КЕРУВАННЯ СИСТЕМАМИ РОЗУМНОГО БУДИНКУ З УРАХУВАННЯМ ПОТРЕБ ЛЮДЕЙ З ІНВАЛІДНІСТЮ	26
2.1 Аналіз сенсорів, виконавчих пристроїв та інтерфейсів користувача в системах «розумний будинок»	26
2.2 Технології передавання даних і протоколи зв'язку в інформаційних системах розумного будинку.....	30
2.3 Типові схеми керування підсистемами розумного будинку з урахуванням доступності для людей з інвалідністю	34
РОЗДІЛ 3 ОПТИМІЗАЦІЯ СКЛАДУ ТА ВИБІР СХЕМИ КЕРУВАННЯ ІНФОРМАЦІЙНОЮ СИСТЕМОЮ РОЗУМНОГО БУДИНКУ ДЛЯ ЛЮДЕЙ З ІНВАЛІДНІСТЮ.....	38
3.1 Архітектура прототипу та структура взаємодії компонентів	38
3.2 Реалізація модуля моніторингу та доступного керування	43
3.3 Розробка підсистеми аварійного оповіщення	47

3.4 Експериментальна оцінка працездатності прототипу та аналіз результатів	51
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	59
4.1 Аналіз умов праці під час монтажу та обслуговування електрообладнання	59
4.2 Заходи безпеки під час проведення електромонтажних робіт	61
4.3 Розробка заходів щодо безпеки у надзвичайних ситуаціях	64
РОЗДІЛ 5 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ	66
5.1 Аналіз ринку та передумови комерціалізації рішення	66
5.2 Розрахунок витрат на розробку, виготовлення та експлуатацію	68
ВИСНОВКИ	72
БІБЛІОГРАФІЧНИЙ СПИСОК	73
ДОДАТОК А КОД ВУЗЛА КЕРУВАННЯ ТА МОНІТОРИНГУ	76
ДОДАТОК Б ПРОГРАМНИЙ КОД НОСИМОГО МОДУЛЯ АВАРІЙНОГО ОПОВІЩЕННЯ	83

ВСТУП

Сучасні системи «розумного будинку» дедалі активніше інтегрують сенсори, виконавчі модулі та програмні сервіси, забезпечуючи автоматизований контроль мікроклімату, освітлення й безпеки житла. Для людей з інвалідністю такі рішення є особливо важливими, адже підвищують автономність, зменшують фізичні навантаження у побуті та забезпечують своєчасне оповіщення про небезпечні ситуації. Проектування відповідної інформаційної системи потребує поєднання IoT-архітектур, принципів універсального дизайну й доступних інтерфейсів, а також урахування надійності та кіберзахисту.

Мета роботи полягає у проектуванні інформаційної системи «розумного будинку» для підвищення доступності, безпеки та комфортності житлового середовища людей з інвалідністю шляхом інтеграції сенсорних пристроїв, виконавчих модулів і адаптивних інтерфейсів керування.

Об'єктом дослідження є процеси автоматизованого моніторингу та керування підсистемами житлового помешкання (освітлення, клімат, безпека, оповіщення) з урахуванням різних функціональних обмежень користувачів. Предметом дослідження є інформаційні технології та програмно-апаратні засоби, що забезпечують моніторинг, керування й адаптивну взаємодію користувача з підсистемами «розумного будинку».

Для досягнення мети необхідно виконати аналітичне дослідження методів і принципів проектування системи з урахуванням потреб людей з інвалідністю, проаналізувати технічні засоби та схеми керування, обґрунтувати оптимізацію складу і вибір схеми керування інформаційною системою, розглянути питання охорони праці та безпеки в надзвичайних ситуаціях під час монтажу й експлуатації, оцінити економічну ефективність запропонованого рішення та соціально-економічний ефект його впровадження.

РОЗДІЛ 1

АНАЛІТИЧНЕ ДОСЛІДЖЕННЯ МЕТОДІВ І ПРИНЦИПІВ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ РОЗУМНОГО БУДИНКУ З УРАХУВАННЯМ ПОТРЕБ ЛЮДЕЙ З ІНВАЛІДНІСТЮ

1.1 Поняття, структура та типова класифікація систем «розумний будинок»

Поняття «розумний будинок» тісно пов'язане з розвитком інформаційних технологій, Інтернету речей, бездротових сенсорних мереж та кіберфізичних систем. Під ним розуміють сукупність технічних і програмних засобів, що забезпечують автоматизований контроль інженерних систем будівлі, побутових приладів, систем безпеки та сервісів комфорту. На відміну від традиційної автоматизації, розумний будинок орієнтований на адаптацію до потреб конкретного користувача, інтеграцію різномірних пристроїв та централізоване або дистанційне керування. Для людей з інвалідністю це поняття доповнюється вимогами доступності, інклюзивності, підтримки самостійності та безпеки, а інформаційна система трактується як елемент реабілітаційної інфраструктури і засіб підвищення якості життя.

Інформаційна система розумного будинку для людей з інвалідністю реалізує функції збору даних про стан середовища й обладнання, аналізу та прийняття рішень, формування команд для виконавчих пристроїв, ведення журналів подій, взаємодії з користувачем через адаптовані інтерфейси та обміну даними з зовнішніми сервісами, зокрема системами «розумного міста» чи медичними інформаційними системами. Вона повинна враховувати різні типи порушень здоров'я й підтримувати альтернативні сценарії взаємодії: голосове керування, керування з мобільних пристроїв, великі контрастні елементи інтерфейсу, тактильні та світлові сигнали, можливість залучення асистентів.

З позицій теорії інформаційних систем розумний будинок є розподіленою системою з великою кількістю кінцевих вузлів, що формують потоки даних від фізичного середовища до рівня прийняття рішень (рис. 1.1). Фізичний рівень охоплює датчики температури, вологості, освітленості, руху, відкриття дверей і вікон, диму та газу, стану побутової техніки, а також носимі пристрої користувача, що особливо важливо в помешканнях людей з інвалідністю. Комунікаційний рівень забезпечує передавання даних дротовими та бездротовими технологіями. На рівні керування працюють контролери, які отримують інформацію від датчиків, аналізують її та формують команди для виконавчих механізмів. Рівень сервісів та інтерфейсів реалізує взаємодію з мешканцем, його опікунами й службами підтримки та інтегрується з хмарними платформами.



Рисунок 1.1 – Багаторівнева архітектура інформаційної системи «розумний будинок» для людей з інвалідністю

На цьому рисунку нижній рівень відображає фізичні пристрої, пов'язані з середовищем та органами чуття людини, середній рівень – локальні й хмарні сервіси збору та обробки інформації, а верхній – користувацькі інтерфейси та прикладні сервіси, інтегровані з медичними й соціальними системами. Такий

підхід спрощує опис і проектування інформаційної системи, оскільки дозволяє чітко розмежувати функції кожного рівня і забезпечити їх узгоджену взаємодію.

Розглянемо також функціональну схему розумного будинку з основними групами компонентів і потоками інформації між ними. На ній виділяють блок сенсорних пристроїв, блок керування та обробки даних, блок виконавчих пристроїв, блок користувацьких інтерфейсів і блок зовнішніх сервісів (рис. 1.2).



Рисунок 1.2 – Узагальнена функціональна структура системи типу «розумний будинок» з урахуванням потреб людей з інвалідністю

Сенсорна підсистема охоплює датчики мікроклімату, безпеки й життєдіяльності, а також кнопки тривоги та носимі пристрої. Підсистема виконавчих пристроїв включає освітлення з димуванням, модулі керування опаленням і вентиляцією, електроприводи дверей, штор, підйомників та керування побутовою технікою. Інтерфейсна підсистема реалізується через

стаціонарні панелі, мобільні застосунки, веб-інтерфейси, голосових асистентів і спеціальні засоби альтернативної комунікації для користувачів з обмеженими можливостями мовлення чи рухів. Блок зовнішніх сервісів забезпечує обмін даними з хмарними платформами, службами оповіщення, телемедициною та диспетчерськими центрами. Для узагальнення характеристик таких систем у контексті потреб людей з інвалідністю сформовано таблицю 1.1 з основними параметрами якості та функціональності.

Таблиця 1.1 – Основні характеристики систем типу «розумний будинок» з урахуванням потреб людей з інвалідністю

Параметр	Коротка характеристика	Значення для людей з інвалідністю
Рівень автоматизації	Від ручного керування до повністю автоматизованих сценаріїв	Зменшує потребу у фізичних зусиллях та кількість рутинних дій
Масштабованість	Можливість поетапного розширення функцій та додавання нових пристроїв	Дозволяє адаптувати систему до змін стану здоров'я та побутових потреб
Надійність та безвідмовність	Стійкість до відмов, наявність резервування та діагностики	Знижує ризики небезпечних ситуацій, пов'язаних із порушенням роботи системи
Доступність інтерфейсів	Наявність альтернативних каналів взаємодії, можливість налаштування інтерфейсу під користувача	Забезпечує автономність, незалежно від типу фізичних чи сенсорних обмежень
Інтеграція із зовнішніми сервісами	Обмін даними з медичними, соціальними, сервісними системами	Підтримує дистанційний моніторинг, реабілітацію та соціальний супровід

Класифікація систем типу «розумний будинок» може здійснюватися за різними ознаками. Однією з ключових є ступінь інтеграції: від окремих рішень для окремих приладів до комплексних систем, що охоплюють усі інженерні мережі. Важливою є й комунікаційна інфраструктура: дротові системи на базі шин KNX чи Modbus, гібридні рішення з Ethernet і Wi-Fi та повністю бездротові системи з протоколами ZigBee, Z-Wave, Thread або LoRaWAN. За способом керування виділяють системи з локальним управлінням, де рішення приймає

контролер у будинку, і хмарно-орієнтовані системи, де частина обробки та зберігання даних виконується на віддалених серверах.

Також важливо враховувати особливості підтримуваних користувачів. Системи для користувачів з обмеженою рухливістю мінімізують фізичні зусилля під час виконання побутових дій. Рішення, орієнтовані на людей з порушеннями зору, застосовують розширені голосові інтерфейси, тактильні елементи, адаптоване освітлення та навігацію. Системи для користувачів з порушеннями слуху роблять акцент на візуальні й світлові сповіщення та дублювання звукових сигналів вібрацією чи повідомленнями на мобільні пристрої. Для людей з когнітивними порушеннями важливими є нагадування, спрощені інтерфейси, автоматичне блокування небезпечних сценаріїв і можливість дистанційного контролю з боку опікунів.

Важливою класифікаційною ознакою є ступінь інтелектуальності системи. Прості системи працюють за фіксованими сценаріями з наперед визначеною реакцією на події. Адаптивні системи змінюють поведінку залежно від історії використання, стану користувача та контексту середовища. Найбільш складні рішення застосовують елементи штучного інтелекту й методи машинного навчання для прогнозування потреб, виявлення небезпечних ситуацій та формування оптимальних стратегій керування. Для людей з інвалідністю саме адаптивні та інтелектуальні системи забезпечують найвищий рівень безпеки й автономності, але потребують обґрунтування з позицій надійності, пояснюваності рішень та кібербезпеки.

З огляду на те, що інформаційна система розумного будинку є системою «людина–машина–середовище», ключову роль відіграє структура та організація інтерфейсів користувача. Сучасний розумний будинок підтримує мультиканальну взаємодію: керування з настінних панелей, смартфона, планшета, комп'ютера та за допомогою голосових команд. Для людей з інвалідністю це дає змогу обрати найбільш зручний спосіб керування залежно від фізичних можливостей (рис. 1.3): користувач з обмеженою рухливістю рук може користуватися голосовим керуванням, тоді як користувач із порушеннями

слуху – візуальними інтерфейсами з великими елементами. Тому у рамках кваліфікаційної роботи доцільно виділити окремий структурний компонент – модуль адаптивних інтерфейсів, який взаємодіє з ядром системи та налаштовується під конкретного користувача.

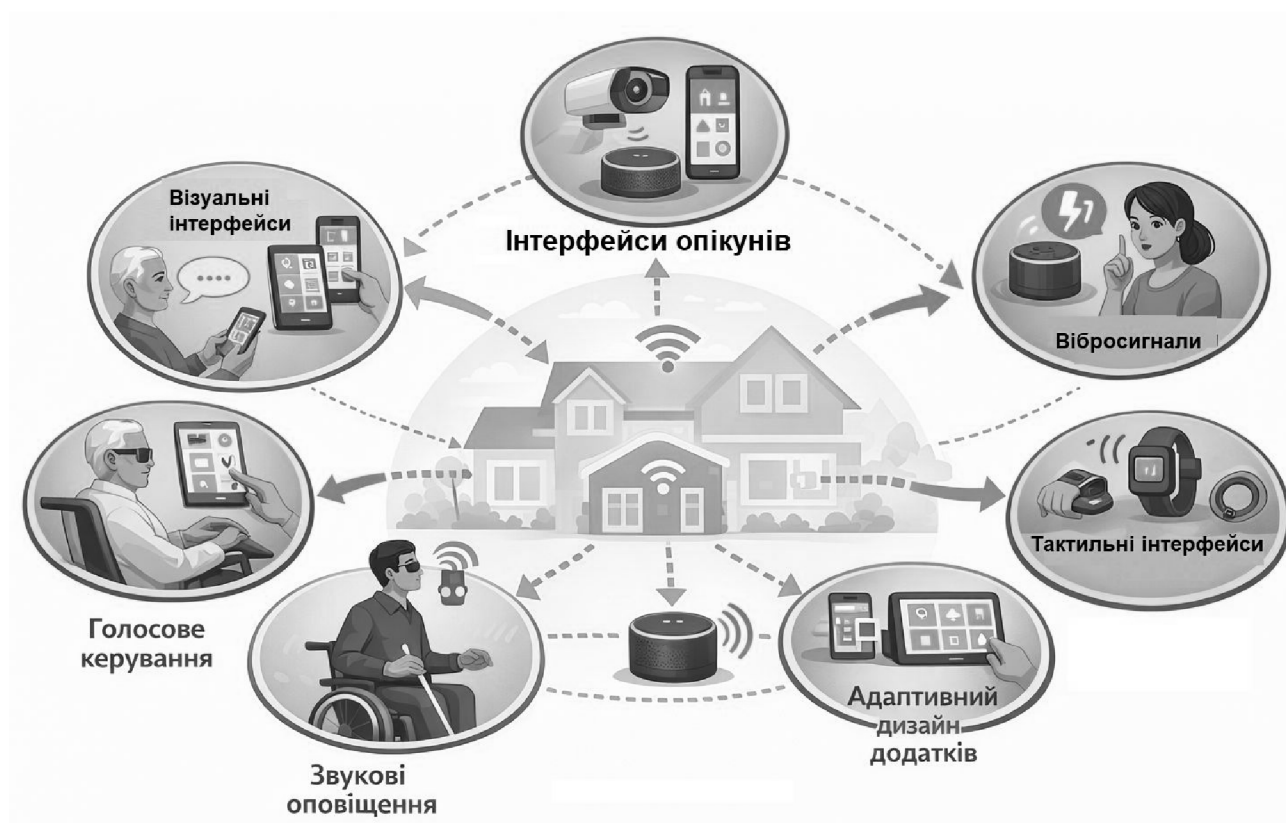


Рисунок 1.3 – Схема взаємодії користувачів з інвалідністю з інтерфейсами системи «розумний будинок»

Голосові інтерфейси призначені для людей з порушеннями опорно-рухового апарату, візуальні інтерфейси з великими шрифтами та контрастними кольорами – для користувачів із залишковим зором, інтерфейси з посиленими вібраційними та світловими сигналами – для людей з порушеннями слуху. Варто також передбачити окремий канал для опікунів чи медичного персоналу, які отримують критичні сповіщення та мають доступ до розширених функцій моніторингу.

Окреслені поняття та структурні особливості систем «розумний будинок» формують основу для подальшого аналітичного дослідження методів і принципів їх проєктування, а також вимоги до побудови інформаційної системи, орієнтованої на людей з інвалідністю.

1.2 Принципи універсального дизайну та доступності середовища для людей з інвалідністю

Концепція універсального дизайну сформувалася як відповідь на необхідність створення середовища, яке було б максимально зручним і безпечним для всіх користувачів, незалежно від їхнього віку, стану здоров'я чи функціональних обмежень. На відміну від підходу, коли для людей з інвалідністю пропонуються окремі спеціальні рішення, універсальний дизайн передбачає первинне закладання принципів доступності у саму структуру будівель, інженерних систем і інформаційних технологій. Для розумного будинку це означає, що всі елементи його інформаційної системи, засоби керування та житловий простір загалом мають проєктуватися так, щоб людина з інвалідністю могла користуватися ними самостійно або з мінімальною допомогою. У цьому контексті інформаційна система розумного будинку виступає ключовою ланкою між користувачем та фізичним середовищем, а її архітектура повинна враховувати принципи універсального дизайну на всіх рівнях – від сенсорів і виконавчих пристроїв до інтерфейсів та алгоритмів керування.

Класичний набір принципів універсального дизайну включає рівність у користуванні, гнучкість, простоту і зрозумілість, сприйнятливість інформації, толерантність до помилок, низький рівень фізичних зусиль, а також відповідні розміри і організацію простору. У застосуванні до розумного будинку принцип рівності у користуванні означає, що усі мешканці, незалежно від наявності

інвалідності, мають доступ до однакового набору функцій системи, без необхідності користуватися «спеціальними» або окремими рішеннями. Наприклад, сценарії керування освітленням, кліматом чи безпекою мають бути доступні як через звичайні настінні панелі, так і через голосові команди чи мобільний додаток з адаптованим інтерфейсом. Принцип гнучкості реалізується через можливість індивідуального налаштування інтерфейсів, чутливості датчиків, порогів спрацювання сигналізації, часу затримки реакції системи, що дозволяє врахувати особливості моторики, зору, слуху або когнітивних можливостей конкретного користувача.

Принцип простоти і зрозумілості вимагає, щоб логіка роботи системи була інтуїтивно зрозумілою, а інтерфейси – послідовними і передбачуваними. Для людей з когнітивними порушеннями надлишкова кількість опцій і складна структура меню можуть становити серйозну перешкоду, тому інформаційна система розумного будинку має підтримувати спрощені профілі керування з мінімальною кількістю дій.

Принцип сприйнятливості інформації передбачає, що повідомлення системи повинні бути доступні різними каналами – візуальним, звуковим, тактильним – і залишатися зрозумілими навіть за умов часткової втрати одного з аналізаторів. Це означає, що оповіщення про небезпеку, нагадування про прийом ліків чи інші важливі події повинні дублюватися різними типами сигналів, а текстова інформація – відображатися великим контрастним шрифтом з можливістю озвучення синтезатором мовлення.

Важливим для безпеки людей з інвалідністю є принцип толерантності до помилок, який вимагає, щоб система мінімізувала наслідки випадкових або некоректних дій користувача. Це може бути реалізовано через підтвердження потенційно небезпечних команд, встановлення обмежень на діапазон регулювання критичних параметрів, автоматичне блокування певних сценаріїв, якщо виявлено ризик для здоров'я або безпеки. Наприклад, система може не дозволяти повністю вимикати опалення в зимовий період без додаткового підтвердження, або автоматично відключати електроплиту за тривалої

відсутності активності на кухні. Принцип низького рівня фізичних зусиль безпосередньо пов'язаний із використанням дистанційних методів керування, автоматизованих сценаріїв та ергономічних пристроїв уведення, що особливо важливо для користувачів з обмеженою рухливістю. Відповідні розміри і організація простору доповнюють інформаційну складову системи: дверні прорізи, проходи, розміщення вимикачів, датчиків, термостатів і керуючих панелей повинні відповідати можливостям користувача на візку або з допоміжними засобами пересування.

Окремої уваги потребують принципи універсального дизайну в побудові інтерфейсів інформаційної системи розумного будинку. Інтерфейси мають бути не лише технічно доступними, але й когнітивно зрозумілими. Це означає використання чіткої ієрархії екранів, великих іконок, зрозумілих підписів, можливості масштабування елементів, а також підтримку альтернативних мовних режимів. Для людей з порушеннями зору важливо забезпечити сумісність інтерфейсу з екранними читачами, контрастний колірний дизайн і можливість керування за допомогою голосу (рис. 1.4).



Рисунок 1.4 – Приклад адаптованого інтерфейсу керування системою «розумний будинок» для користувача з порушеннями зору

Третій аспект, що безпосередньо пов'язаний з універсальним дизайном, стосується просторової організації житлового середовища. Інформаційна система розумного будинку не може компенсувати недоліки планування або бар'єри у фізичному середовищі, проте може суттєво підсилити ефект від правильно спроектованого безбар'єрного простору. Широкі дверні прорізи, відсутність порогів, достатній простір для маневрування інвалідного візка, правильне розміщення меблів та обладнання в поєднанні з автоматизованим керуванням освітленням, жалюзі, дверима чи підйомниками створюють цілісну систему підтримки.

Для систематизації взаємозв'язку між принципами універсального дизайну та функціями інформаційної системи розумного будинку подано узагальнену таблицю 1.2, у якій кожному принципу відповідають конкретні рішення на рівні інтерфейсів, служб оповіщення та алгоритмів керування.

Таблиця 1.2 – Відповідність принципів універсального дизайну функціям інформаційної системи розумного будинку

Принцип універсального дизайну	Приклад реалізації в інформаційній системі розумного будинку	Значення для людей з інвалідністю
Рівність у користуванні	Єдиний набір функцій, доступний через панель, мобільний додаток і голосове керування	Уникається сегрегація, користувач не відчуває себе «винятком»
Гнучкість і персоналізація	Налаштування профілів користувачів, порогів спрацювання датчиків, рівня автоматизації	Система пристосовується до індивідуальних можливостей та змін стану здоров'я
Сприйнятливості інформації	Дублювання оповіщень візуальними, звуковими та вібраційними сигналами	Забезпечується сприйняття критичних повідомлень за різних видів порушень
Толерантність до помилок	Підтвердження небезпечних команд, автоматичне блокування ризикованих сценаріїв	Зменшується ймовірність небезпечних ситуацій через випадкові дії
Низький рівень фізичних зусиль	Автоматичні сценарії керування освітленням, кліматом, дверима та побутовою технікою	Мінімізуються фізичні навантаження, полегшується виконання повсякденних дій

Таким чином, принципи універсального дизайну та доступності середовища задають методологічну рамку, у межах якої повинно здійснюватися

проектування інформаційної системи розумного будинку для людей з інвалідністю. Вони визначають вимоги до структури інтерфейсів, алгоритмів керування, способів подання інформації та просторової організації житла. Дотримання цих принципів дозволяє перейти від окремих фрагментарних рішень до цілісної системи підтримки, що не лише компенсує функціональні обмеження користувачів, а й створює для них умови максимальної автономності та безпеки.

1.3 Методи проектування інформаційних систем підтримки доступного житла

Проектування інформаційних систем підтримки доступного житла для людей з інвалідністю потребує поєднання класичних підходів програмної інженерії з принципами універсального дизайну, інклюзивних практик та специфіки Інтернету речей. На відміну від типових інформаційних систем, де акцент робиться передусім на функціональності та продуктивності, у випадку систем доступного житла першочерговими є безпека користувача, надійність, зручність і зрозумілість взаємодії, а також адаптація до різних типів функціональних обмежень. Тому важливо не лише обрати відповідну методологію розробки, а й інтегрувати в неї етапи аналізу потреб цільових груп, моделювання сценаріїв використання та тестування з участю реальних користувачів.

Класичні підходи до проектування інформаційних систем базуються на моделі життєвого циклу, яка включає етапи аналізу вимог, проектування, реалізації, тестування, впровадження та супроводу. Для систем підтримки доступного житла ці етапи доповнюються ретельним вивченням бар'єрів у фізичному та інформаційному середовищі, що впливають на людей з інвалідністю. На етапі аналізу вимог важливо не обмежуватися загальними технічними характеристиками, а проводити інтерв'ю, спостереження та

опитування користувачів, їхніх опікунів і фахівців з реабілітації, щоб зрозуміти реальні побутові сценарії, у яких інформаційна система має забезпечити підтримку. У подальшому ці вимоги трансформуються в модель прецедентів використання, діаграми бізнес-процесів і функціональні специфікації, що лягають в основу архітектурних рішень. Узагальнений підхід до життєвого циклу проєктування інформаційної системи доступного житла представлено у вигляді структурної схеми на рис. 1.5, де послідовно відображені етапи аналізу потреб користувачів, формулювання функціональних і нефункціональних вимог, архітектурного та детального проєктування, прототипування, тестування, впровадження і зворотного зв'язку. Варто підкреслити наявність циклів ітеративного вдосконалення, що особливо важливо, коли в процесі експлуатації виявляються нові потреби або змінюється стан здоров'я користувачів.



Рисунок 1.5 – Узагальнена схема життєвого циклу проєктування інформаційної системи доступного житла

Сучасні підходи до розробки систем розумного будинку дедалі частіше ґрунтуються на користувач-орієнтованому та партисипативному дизайні. Користувач-орієнтований дизайн передбачає розгляд людини з інвалідністю як центрального елемента системи, а не лише як «кінцевого споживача». Це означає, що майбутні користувачі залучаються до процесу створення системи на всіх ключових етапах: від формування вимог до оцінювання прототипів. Партисипативний підхід ще більше розширює цю участь, оскільки передбачає спільне прийняття рішень щодо функціональності, сценаріїв використання, зовнішнього вигляду інтерфейсів та пріоритетів розвитку системи. Для доступного житла це критично, адже зовнішній спостерігач часто не може повністю усвідомити всі труднощі, з якими стикається людина з інвалідністю у повсякденних ситуаціях.

Важливе значення у проєктуванні систем розумного будинку для людей з інвалідністю має моделювання. Використання діаграм прецедентів, діаграм класів, діаграм послідовностей та діаграм станів дозволяє формалізувати сценарії взаємодії користувача з системою, описати логіку роботи підсистем керування, відобразити інформаційні потоки між датчиками, контролерами, хмарними сервісами та інтерфейсами. Окремо доцільно моделювати інтерфейсні сценарії, наприклад послідовність дій користувача при виклику допомоги, зміні режиму освітлення або підтвердженні важливого повідомлення. Це дає змогу виявити потенційні точки помилок, надмірну кількість кроків, нечіткі стани та інші проблеми, які можуть стати критичними для користувачів з когнітивними або моторними обмеженнями.

Систематизовані дані щодо особливостей різних методів проєктування інформаційних систем, що можуть бути використані для підтримки розумного житла представлено у порівняльній таблиці 1.3.

У системах розумного будинку для людей з інвалідністю доцільно поєднувати елементи зазначених підходів. Каскадна або спіральна модель може використовуватися на рівні формалізації вимог, розроблення архітектурних рішень та узгодження з нормативно-правовими документами. Ітеративні методи

дозволяють реалізовувати окремі підсистеми та функціональні блоки з урахуванням постійного зворотного зв'язку. Користувач-орієнтований дизайн забезпечує адаптацію інтерфейсів і сценаріїв використання до специфічних потреб різних груп людей з інвалідністю. Важливу роль відіграє також модельно-орієнтоване проєктування, коли на основі формальних моделей автоматично генеруються частини програмного коду, конфігурації баз даних та інтерфейсів, що зменшує кількість помилок і полегшує супровід системи.

Таблиця 1.3 – Порівняння методів проєктування інформаційних систем підтримки доступного житла

Метод проєктування	Коротка характеристика	Переваги для доступного житла	Потенційні обмеження
Каскадна модель	Послідовні етапи від аналізу до впровадження	Чітка структура документації, зручна для формальних погоджень	Мала гнучкість, складно враховувати зміну потреб
Спіральна модель	Ітеративні цикли з оцінкою ризиків	Можливість поетапного уточнення вимог і врахування безпеки	Вища складність планування і координації
Ітеративні методи	Часті релізи, тісний зв'язок з замовником	Швидке реагування на зворотний зв'язок користувачів	Потребують дисципліни команди і доступності користувачів
Користувач-орієнтований дизайн	Акцент на дослідженні потреб і досвіді користувача	Найкраща відповідність реальним потребам людей з інвалідністю	Потребує значних зусиль для організації участі користувачів

З огляду на складність та багаторівневість інформаційних систем підтримки доступного житла, доцільно використовувати моделі, які інтегрують три основні аспекти: функціональний, інформаційний та інтерфейсний (рис. 1.6). Функціональний аспект описує, які задачі виконує система та які сервіси надає користувачам. Інформаційний аспект визначає структуру даних, що збираються, зберігаються та обробляються, зокрема дані сенсорів, журнали подій, профілі користувачів, історію змін параметрів середовища. Інтерфейсний аспект описує форми подання інформації та способи введення команд, включаючи голосові, тактильні та візуальні канали.

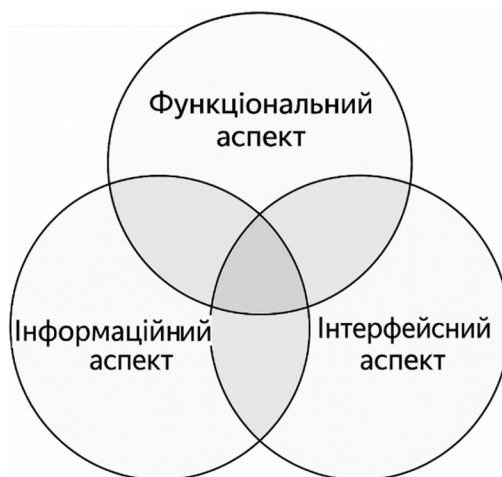


Рисунок 1.6 – Інтегрована модель функціонального, інформаційного та інтерфейсного аспектів системи підтримки доступного житла

Таким чином, методи проектування інформаційних систем підтримки доступного житла базуються на поєднанні класичних моделей життєвого циклу, ітеративних підходів, користувач-орієнтованого та партисипативного дизайну, а також активному використанні моделей для опису функцій, даних та інтерфейсів. Усі ці методи мають бути адаптовані до специфіки людей з інвалідністю, що передбачає пріоритет безпеки, надійності, простоти й зрозумілості системи, а також можливість її подальшої еволюції разом із змінами потреб користувачів.

1.4 Огляд нормативно-правових, технічних та ергономічних вимог до систем «розумний будинок» для людей з інвалідністю

Проектування інформаційної системи «розумного будинку» для людей з інвалідністю неможливе без урахування комплексу нормативно-правових, технічних та ергономічних вимог. Такі вимоги формують рамкові умови, у межах яких допускається розроблення, впровадження та експлуатація системи, і забезпечують не лише її технічну працездатність, а й відповідність принципам

доступності, безбар'єрності та безпеки. Нормативно-правова база задає загальні стандарти щодо прав осіб з інвалідністю, архітектурної доступності будівель, безпечної експлуатації електротехнічного та телекомунікаційного обладнання, а також захисту персональних даних. Технічні стандарти деталізують вимоги до надійності, сумісності та кібербезпеки систем, тоді як ергономічні критерії зосереджені на комфорті, зручності й зрозумілості взаємодії користувача з технічними засобами.

У контексті розумного будинку для людей з інвалідністю нормативно-правові вимоги мають багаторівневий характер. На міжнародному рівні ключову роль відіграють документи, що закріплюють права людей з інвалідністю на доступне житло та безбар'єрне середовище. Вони встановлюють принципи недискримінації, рівності можливостей, забезпечення розумного пристосування та універсального дизайну. На національному рівні ці положення конкретизуються у законодавстві щодо соціального захисту осіб з інвалідністю, у будівельних нормах і стандартах доступності житлових і громадських будівель, а також у документах, що регламентують вимоги до інженерних систем, електробезпеки, пожежної безпеки, використання телекомунікаційних і радіоелектронних засобів. Для інформаційних систем особливо важливими є вимоги щодо захисту персональних даних, адже розумний будинок збирає і обробляє чутливу інформацію про стан здоров'я, поведінкові патерни та повсякденну активність користувача.

Взаємозв'язок між міжнародними документами, національним законодавством, будівельними нормами та галузевими стандартами, які разом формують нормативно-правову основу для проєктування систем розумного будинку подано на рис. 1.7. З рисунку видно, що система має відповідати не одному документу, а цілому комплексу узгоджених вимог.

Технічні вимоги до систем «розумний будинок» охоплюють кілька груп аспектів. По-перше, це вимоги до безпеки та надійності обладнання, адже система має забезпечувати стабільну роботу навіть в умовах перебоїв живлення, радіоперешкод або часткової відмови окремих компонентів. По-друге, важлива

сумісність і інтероперабельність, оскільки сучасний розумний будинок складається з різноманітних пристроїв, що працюють за різними протоколами зв'язку. По-третє, значення набувають вимоги до кібербезпеки, адже віддалений доступ, хмарні сервіси та інтеграція з зовнішніми платформами створюють додаткові канали для потенційних атак. У системі, орієнтованій на людей з інвалідністю, порушення роботи або несанкціоноване втручання може мати безпосередній вплив на їхню безпеку, наприклад шляхом відключення аварійних сповіщень або блокування доступу до житлового приміщення.

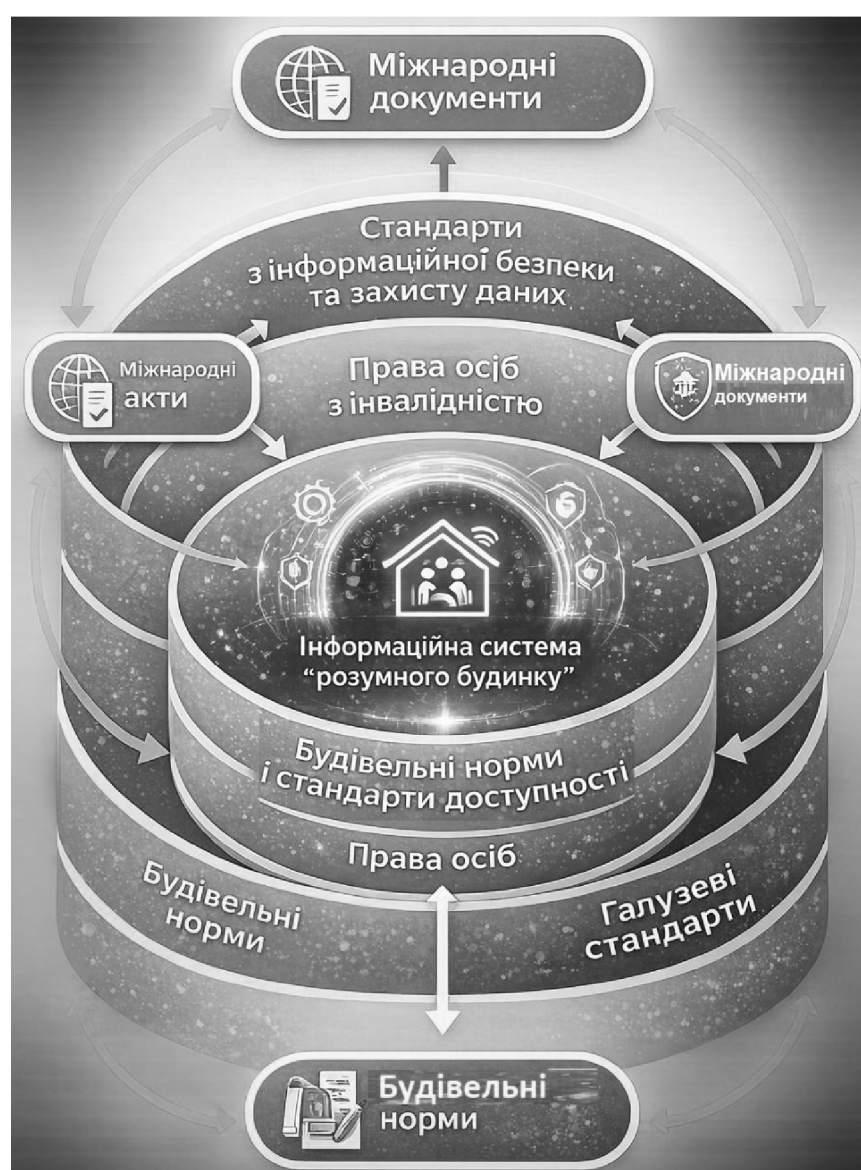


Рисунок 1.7 – Узагальнена схема нормативно-правової бази проєктування системи «розумний будинок» для людей з інвалідністю

Ергономічні вимоги до систем «розумний будинок» для людей з інвалідністю спрямовані на зручність, зрозумілість і доступність взаємодії. Вони охоплюють фізичні параметри (висота розміщення елементів керування, розміри кнопок, необхідне зусилля натискання) та когнітивні аспекти (однозначність позначень, логіка меню, кількість кроків для виконання дій). Для користувачів із порушеннями зору ключовими є контрастність, достатній розмір шрифту та підтримка озвучування, а для користувачів із порушеннями слуху – виразні візуальні й тактильні сигнали. Вимоги мають враховувати особливості моторики, мінімізувати ризик помилкових натискань і уникати надлишку дрібних елементів керування. Узагальнення цих положень наведено в таблиці 1.4.

Таблиця 1.4 – Основні групи вимог до систем «розумний будинок» для людей з інвалідністю

Група вимог	Зміст вимог	Значення для людей з інвалідністю
Нормативно-правові	Відповідність законодавству щодо прав осіб з інвалідністю, будівельним нормам, вимогам безпеки та захисту даних	Гарантує правомірність впровадження системи та захист прав користувача
Технічні	Вимоги до надійності, безпеки, сумісності, кібербезпеки, якості каналів зв'язку	Забезпечують стабільну роботу системи, безпечну експлуатацію та захист від збоїв і атак
Ергономічні	Вимоги до зручності, зрозумілості, фізичної й когнітивної доступності інтерфейсів	Підтримують автономність користувача, знижують ризик помилок та перевантаження

У підсумку можна зазначити, що огляд нормативно-правових, технічних та ергономічних вимог є ключовим етапом аналітичного дослідження в межах проєктування інформаційної системи розумного будинку для людей з інвалідністю. Саме на цьому етапі визначається, у яких межах і за якими правилами може функціонувати система, які стандарти вона повинна виконувати та як забезпечити баланс між технічною складністю, безпекою та зручністю.

РОЗДІЛ 2

АНАЛІЗ ТЕХНІЧНИХ ЗАСОБІВ ТА СХЕМ КЕРУВАННЯ СИСТЕМАМИ РОЗУМНОГО БУДИНКУ З УРАХУВАННЯМ ПОТРЕБ ЛЮДЕЙ З ІНВАЛІДНІСТЮ

2.1 Аналіз сенсорів, виконавчих пристроїв та інтерфейсів користувача в системах «розумний будинок»

Ефективність функціонування системи «розумний будинок» значною мірою визначається якістю та структурою сенсорних пристроїв, виконавчих механізмів і каналів взаємодії з користувачем. Це особливо актуально у контексті створення доступного житлового середовища для людей з інвалідністю, оскільки саме сенсори забезпечують первинний збір інформації про стан середовища, виконавчі механізми реалізують дії системи у фізичному просторі, а інтерфейси стають основним каналом комунікації між людиною та інтелектуальними сервісами. У системі, орієнтованій на людей із порушеннями рухливості, зору, слуху або когнітивних функцій, усі технічні компоненти мають формувати єдину синхронізовану структуру, здатну компенсувати обмеження користувача, підвищувати рівень безпеки та зменшувати залежність від сторонньої допомоги.

Сенсорні пристрої можна охарактеризувати як первинні джерела даних, що забезпечують можливість оцінювання параметрів мікроклімату, просторового положення об'єктів, рухової активності користувача та стану побутового обладнання. У розумному будинку для людей з інвалідністю сенсори відіграють роль не лише технічних компонентів, але й елементів системи підтримки життєдіяльності. Вони контролюють температурний режим, рівень освітленості, якість повітря, наявність диму або газу, фіксують відкриття дверей, вікон та реєструють переміщення користувача. Особливу роль відіграють сенсори життєдіяльності, до яких належать датчики падінь, носимі трекери активності,

пульсоксиметри, кнопки тривоги та інші пристрої дистанційного моніторингу. Їхнє використання дозволяє контролювати критичні стани, підтримувати зв'язок з опікунами або медичними службами та забезпечувати оперативне реагування у надзвичайних ситуаціях.

На рисунку 2.1 представлено узагальнену класифікаційну схему сенсорних пристроїв у системі «розумний будинок», у якій обладнання розділено на групи за призначенням: сенсори мікроклімату, сенсори безпеки, сенсори життєдіяльності та сенсори побутового обладнання. Така схема наочно демонструє різноманіття сенсорних джерел даних та їх зв'язок із головною логікою системи. А систематизацію характеристик вищезгаданих сенсорних пристроїв наведено у порівняльній таблиці 2.1 де зібрано основні параметри, що мають значення для проектування системи розумного житла.



Рисунок 2.1 – Класифікація сенсорних пристроїв у системі «розумний будинок»

Особливе місце в архітектурі доступного житла займають виконавчі пристрої, які переводять рішення системи у фізичні дії. До таких пристроїв відносяться електроприводи дверей і штор, системи дімування освітлення, автоматизовані модулі опалення та вентиляції, інтелектуальні розетки та керовані блоки побутової техніки. У житлі людей з інвалідністю виконавчі елементи виконують функцію компенсації обмежених фізичних можливостей, надаючи можливість автоматичного відкриття дверей, налаштування освітлення, регулювання температурного режиму та активації побутових приладів без необхідності прямої участі користувача.

Таблиця 2.1 – Основні технічні характеристики сенсорів у системі «розумний будинок»

Тип сенсора	Призначення	Важливі параметри для доступного житла	Особливості застосування
Мікрокліматичні (температури, вологості, освітлення)	Контроль стану середовища	Чутливість, швидкість реакції, стабільність показників	Забезпечують комфорт та енергозбереження
Безпеки (дим, газ, рух, відкриття)	Виявлення небезпек	Надійність, захист від хибних спрацювань	Критичні для людей з обмеженою мобільністю
Життєдіяльності	Контроль стану користувача	Безперервність моніторингу, автономність	Виявлення падінь, аварійних станів
Сенсори побутових пристроїв	Відстежування роботи техніки	Сумісність із протоколами IoT	Автоматизація побутових процесів

На рисунку 2.2 подано структурну схему взаємодії виконавчих пристроїв із центральним контролером, яка демонструє, що кожен модуль може отримувати як автоматичні команди, так і команди від користувача через адаптивні інтерфейси. А в узагальненій таблиці 2.2 відображено технічні характеристики виконавчих механізмів, важливі у контексті доступності.

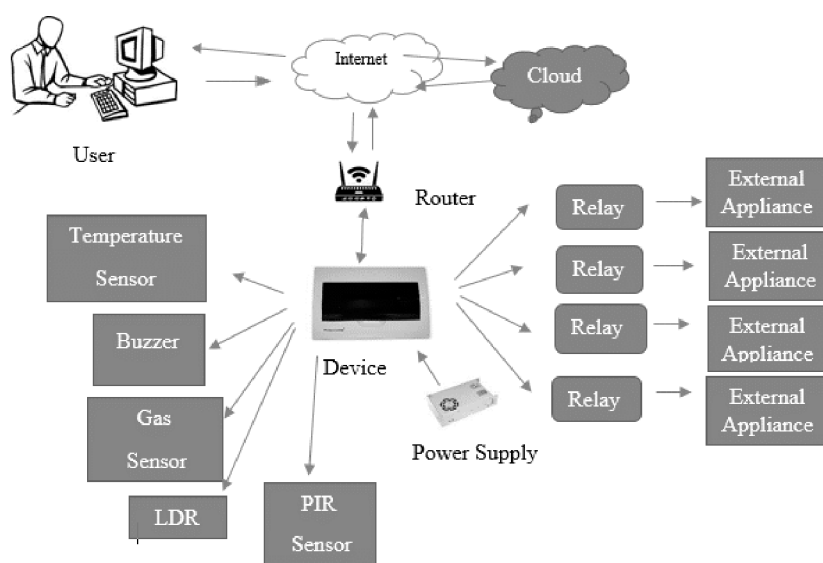


Рисунок 2.2 – Структура взаємодії виконавчих пристроїв з контролером

Таблиця 2.2 – Аналіз виконавчих пристроїв системи доступного розумного будинку

Тип виконавчого пристрою	Основне призначення	Вимоги для людей з інвалідністю	Приклади використання
Електроприводи дверей, штор, ліжок	Автоматичне керування об'єктами	Безпечний режим, плавний рух, наявність аварійної зупинки	Відкриття дверей, регулювання положення ліжка
Освітлювальні блоки з дімуванням	Керування освітленням	Плавне регулювання, голосове керування	Зміна яскравості у нічний час
Інтелектуальні розетки	Керування побутовою технікою	Автоматичне відключення, перевірка стану	Керування нагрівачами, приладами
Кліматичні модулі	Регуляція мікроклімату	Точність, дистанційне керування, адаптація	Опалення, кондиціювання

Особливої уваги потребують інтерфейси користувача, оскільки вони забезпечують можливість автономного керування житловим середовищем. Інтерфейси можуть бути реалізовані у вигляді мобільних застосунків, стаціонарних панелей, веб-інтерфейсів, а також голосових систем та спеціальних пристроїв альтернативної комунікації. Кожен із цих інтерфейсів має свої переваги та обмеження залежно від особистих потреб користувача. Людям з порушеннями рухливості зручним є голосове керування та інтерфейси з мінімальною кількістю жестів; людям із порушеннями зору важлива озвучка меню, високий контраст та можливість повної навігації без візуальних елементів; користувачам із порушеннями слуху потрібні текстові та візуальні сповіщення.

Узагальнюючи, можна стверджувати, що сенсори, виконавчі пристрої та інтерфейси користувача формують три взаємопов'язані підсистеми, від яких залежить не лише функціональність розумного будинку, а й рівень безпеки, автономності та комфорту людей з інвалідністю. Саме гармонійне поєднання цих компонентів забезпечує можливість побудови доступного житлового середовища, у якому технічні засоби стають інструментом розширення можливостей користувача.

2.2 Технології передавання даних і протоколи зв'язку в інформаційних системах розумного будинку

Передавання даних у системах «розумного будинку» визначає швидкість, надійність, масштабованість та енергоефективність усіх компонентів, що взаємодіють у єдиному середовищі. В умовах доступного житла для людей з інвалідністю ці характеристики набувають ще більшого значення, оскільки швидкість реакції та стабільність з'єднання безпосередньо впливають на комфортність, безпеку життєвого простору та можливість незалежного користування системою. Інформаційна система повинна забезпечувати надійну передачу сенсорних даних, ефективну взаємодію виконавчих пристроїв, підтримку адаптивних інтерфейсів та оперативне оповіщення у випадку надзвичайних ситуацій. Фундаментальним етапом проєктування такої системи є визначення оптимальних технологій зв'язку для заданих умов експлуатації, архітектури будинку, кількості користувачів та їх функціональних потреб.

Технології передавання даних у розумному будинку умовно охоплюють дротові та бездротові рішення. Дротові варіанти зазвичай застосовуються у високонавантажених або критично важливих системах, де потрібна максимальна стабільність і мінімальна затримка. Водночас бездротові технології здобули значного поширення завдяки простоті інтеграції, гнучкості та можливості встановлення у приміщеннях без проведення додаткових комунікацій. У середовищі доступного житла бездротові мережі відіграють особливо важливу роль, оскільки забезпечують мінімальний рівень втручання в інфраструктуру будинку, дають змогу встановлювати сенсори та виконавчі модулі у зручних для користувача місцях, а також дозволяють легко модернізувати систему.

Серед бездротових протоколів особливе місце займають Wi-Fi, ZigBee, Z-Wave, Bluetooth Low Energy та LoRaWAN, кожен із яких надає унікальні можливості і формує власні технічні переваги. Wi-Fi забезпечує високу пропускну здатність і використовується здебільшого для мультимедійних

пристроїв, відеонагляду, голосових помічників та інтерфейсів користувача. У доступному житлі цей стандарт дає змогу об'єднати комунікації в єдину інфраструктуру, однак потребує стабільного живлення і має обмеження за енергоефективністю, що не завжди підходить для сенсорних вузлів тривалого автономного функціонування.

ZigBee та Z-Wave використовуються у малопотужних мережах, де ключовими параметрами є надійність та низьке енергоспоживання. ZigBee функціонує на основі топології mesh, що дозволяє мережі самостійно маршрутизувати трафік між вузлами та компенсувати окремі збої. Самовідновлювана структура є особливо корисною в середовищі, де мобільність користувача з інвалідністю або переміщення меблів можуть змінювати радіочастотні умови. Z-Wave, навпаки, пропонує більшу сумісність між виробниками та стабільність зв'язку у діапазонах, менш схильних до перешкод, що робить його ефективним для керування освітленням, жалюзі, дверними приводами та іншими виконавчими пристроями.

Bluetooth Low Energy часто використовується для локальних інтерфейсів взаємодії та персональних пристроїв допомоги, зокрема для користувачів з порушеннями опорно-рухового апарату або когнітивними порушеннями. Перевагою є можливість швидкого з'єднання та низьке енергоспоживання, проте дальність дії значно менша порівняно з іншими технологіями. У інтегрованих рішеннях BLE застосовується для автентифікації користувача, запуску сценаріїв автоматизації та комунікації зі смартфоном або інклюзивними пристроями.

LoRaWAN набуває поширення у великих будинках або смарт-кварталах, де необхідно забезпечити широке покриття та автономність сенсорів. Технологія підтримує передачу даних на значні відстані при мінімальних витратах енергії. У розумному будинку для людей з інвалідністю LoRaWAN може використовуватися для безпроводних тривожних кнопок, датчиків газу, диму, затоплення та інших модулів, які мають працювати упродовж багатьох років без заміни батареї. Порівняння технічних параметрів основних протоколів зв'язку для розумного будинку наведено у табл. 2.3.

Таблиця 2.3 – Порівняльні характеристики протоколів зв'язку для розумного будинку

Протокол	Дальність дії	Енергоспоживання	Пропускна здатність	Типові задачі
Wi-Fi	30–50 м	високе	висока	мультимедіа, відеонагляд, інтерфейси
ZigBee	10–30 м (mesh)	низьке	середня	датчики, автоматизація, освітлення
Z-Wave	30–100 м	низьке	середня	приводи, замки, контроль доступу
BLE	5–20 м	дуже низьке	низька	персональні пристрої, локальна взаємодія
LoRaWAN	до декількох кілометрів	дуже низьке	низька	тривожні модулі, автономні сенсори

Важливою складовою проєктування є оцінка сумісності протоколів та пристроїв. У багатьох сучасних системах застосовується підхід гібридної інтеграції, коли декілька технологій доповнюють одна одну. Наприклад, сигнали від малопотужних сенсорів можуть передаватися через ZigBee, тоді як інтерфейси користувача та мультимедійні системи взаємодіють через Wi-Fi. Це дозволяє оптимізувати навантаження на мережу, зменшити споживання енергії та забезпечити резервування зв'язку у критично важливих ситуаціях (рис. 2.3). Для людей з інвалідністю така архітектура відкриває можливість створення адаптивних сценаріїв, які гарантовано спрацьовують за будь-яких умов.

Особливу увагу варто приділити питанням безпеки. Системи доступного житла повинні бути захищені від перехоплення даних, несанкціонованого керування обладнанням та зовнішніх атак. Для цього використовуються протоколи шифрування, автентифікації та двофакторної перевірки доступу. ZigBee та Z-Wave мають вбудовані механізми безпеки, тоді як Wi-Fi покладається на стандарти WPA3 та додаткові інструменти захисту. У контексті інклюзивного доступу важливо забезпечити не лише технічний, але й

ергономічний рівень безпеки, щоб користувачі могли легко та швидко підтверджувати свої дії.

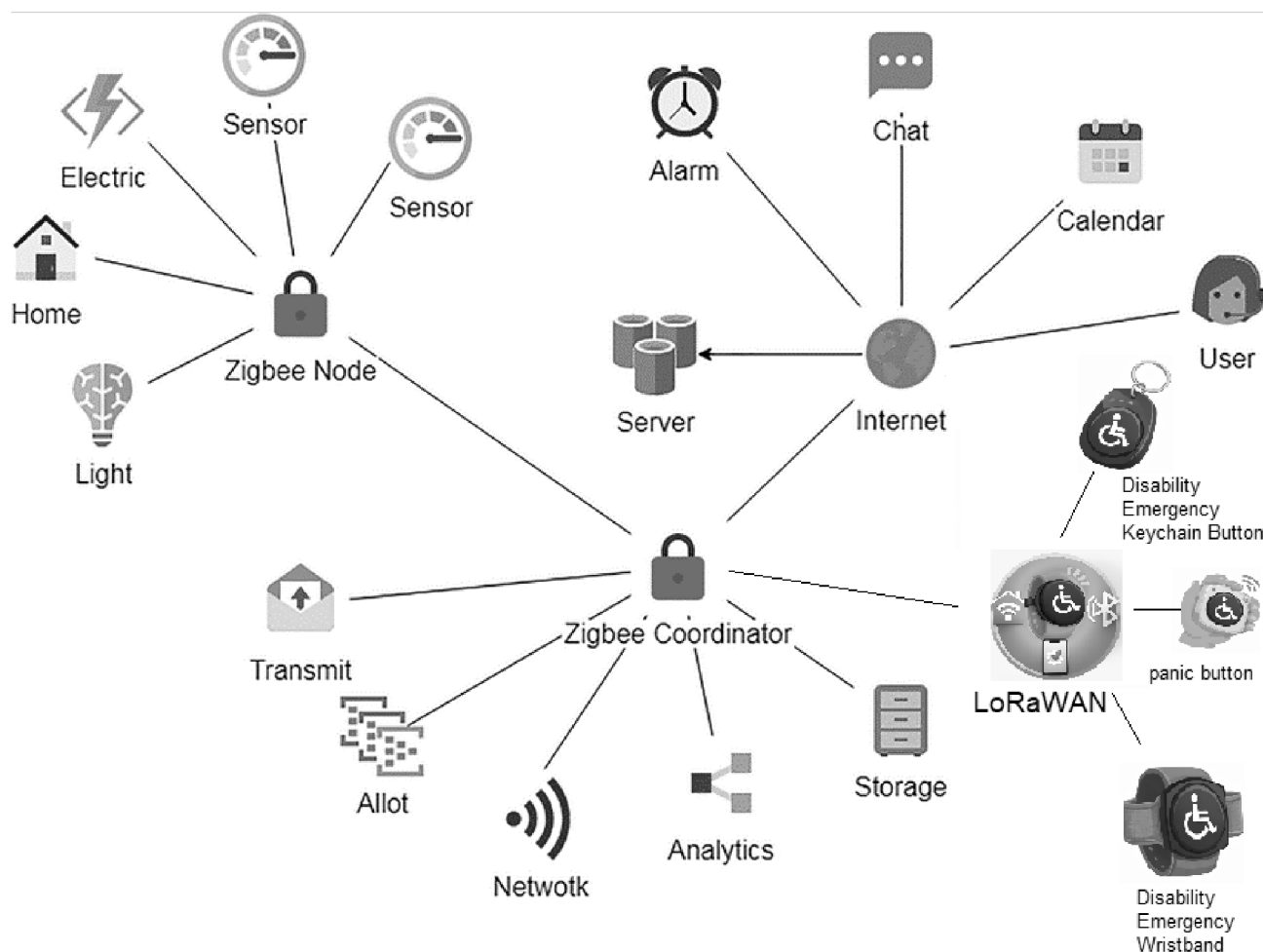


Рисунок 2.3 – Архітектура гібридної мережі передавання даних у доступному житлі

Технології передачі даних у системах розумного будинку виконують роль нервової системи, яка забезпечує узгоджену роботу всіх компонентів. Коректний вибір протоколів та їх взаємодія дають змогу адаптувати житловий простір відповідно до сенсорних, моторних та когнітивних потреб людей з інвалідністю. Сучасні засоби зв'язку забезпечують можливість створення середовища, яке автоматично реагує на зміни стану користувача, дозволяє працювати із системою через голос, сенсорні панелі, мобільні застосунки або адаптивні пристрої допомоги, і водночас гарантує стабільність та безпечність комунікацій.

2.3 Типові схеми керування підсистемами розумного будинку з урахуванням доступності для людей з інвалідністю

Типові схеми керування підсистемами розумного будинку відображають взаємодію сенсорів, виконавчих пристроїв, контролерів та інтерфейсів користувача в різних житлових сценаріях. Для людей з інвалідністю такі схеми мають враховувати не тільки технічну ефективність, а й доступність, безпеку, наявність резервних каналів керування та можливість адаптації під індивідуальні потреби. Тому при аналізі схем керування важливо розглядати не окремі компоненти, а цілісні підсистеми, які забезпечують автоматизацію освітлення, мікроклімату, безпеки, керування доступом та аварійного оповіщення.

Узагальнено структуру керування підсистемами розумного будинку можна подати як багаторівневу схему, де на нижньому рівні розміщені сенсори та виконавчі пристрої, на середньому – локальні контролери або мікроконтролерні вузли, а на верхньому – центральний сервер або контролер сцена-рівня, який об'єднує всі підсистеми в єдину логіку. На рисунку 2.4 подано типову структурну схему керування підсистемами розумного будинку з урахуванням потреб людей з інвалідністю. У цій схемі кожна підсистема (освітлення, клімат, безпека, доступ, моніторинг стану користувача) має власний локальний контур керування, але водночас підпорядковується глобальним сценаріям і правилам безпеки.

Типовою схемою керування підсистемою освітлення в доступному житлі є поєднання локальної автоматики на основі датчиків руху й освітленості з можливістю ручного та дистанційного коригування режимів. У коридорах, санвузлах та інших потенційно небезпечних зонах застосовується автоматичне вмикання світла при вході користувача, що зменшує потребу в пошуку вимикача, особливо для людей з обмеженою рухливістю або порушеннями зору. У спальнях і вітальнях використовуються сценарії плавного дімування, прив'язані до часу доби, стану користувача або медичних рекомендацій, а для підвищення

доступності передбачається дублювання керування: настінні вимикачі на зручній висоті, мобільний застосунок, голосові команди та кнопки швидкого вимкнення біля ліжка.

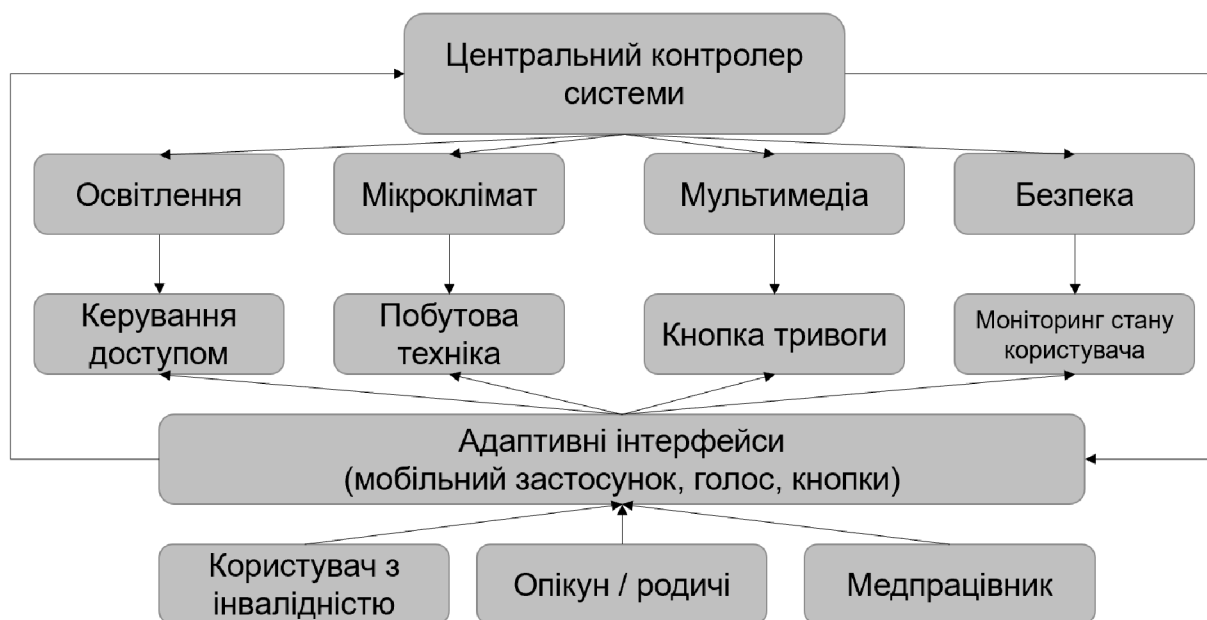


Рисунок 2.4 – Узагальнена схема керування підсистемами розумного будинку з урахуванням потреб людей з інвалідністю

Підсистема мікроклімату включає керування опаленням, кондиціонуванням і вентиляцією на основі термостатів, датчиків температури та вологості й виконавчих регуляторів. У житлі людей з інвалідністю вона доповнюється індивідуальними температурними профілями, що враховують медичні показання та рівень активності: уночі параметри змінюються відповідно до режиму сну, а за тривалої відсутності руху система переходить в енергозберігальний режим без втрати комфорту. Важливою є інтеграція з безпекою: при виявленні диму чи загрози загоряння автоматика змінює вентиляційний режим згідно зі сценарієм.

Підсистема безпеки та аварійного оповіщення в розумному будинку для людей з інвалідністю має критичне значення. Вона використовує датчики диму, газу, затоплення, руху, відкриття дверей і вікон, а також тривожні кнопки та

носимі пристрої. Контролери аналізують сигнали й формують керуючі дії: звукові та світлові оповіщення, блокування газових клапанів, перекриття водопостачання, розблокування дверей для евакуації, а сервер або хмарний сервіс дублює повідомлення на пристрої опікунів, родичів або екстрених служб. Для користувачів із порушеннями слуху звукові сигнали мають дублюватися візуальними й тактильними, а для людей із порушеннями зору – голосовими та тактильними повідомленнями.

Підсистема керування доступом охоплює електронні замки, домофонні системи, відеодзвінки та модулі дистанційного відкриття дверей. У доступному житлі передбачаються кілька способів відкриття: мобільний застосунок, настінна панель у зоні досяжності, голосова команда або брелок/картка доступу, що дозволяє людині на візку чи з обмеженою моторикою уникати складних дій із традиційними замками. У разі небезпеки система може автоматично розблокувати двері для рятувальних служб, зберігаючи захищеність у звичайних умовах, а також підтримувати тимчасовий доступ для соціальних працівників чи медичного персоналу з реєстрацією подій.

Особливе місце займає підсистема моніторингу стану користувача, яка включає носимі датчики, трекери активності, сенсори падінь та кнопки тривоги. Вона забезпечує безперервний збір даних, аналіз патернів активності та виявлення відхилень, що можуть свідчити про погіршення стану здоров'я. У разі падіння або тривалої відсутності руху система формує аварійний сигнал для локального оповіщення та віддалених сервісів, при цьому доцільно передбачити підтвердження помилкової тривоги користувачем, щоб уникнути надмірного навантаження на опікунів і екстрені служби.

Усі зазначені підсистеми можуть будуватися на основі різних типових схем керування: локальних, централізованих та гібридних. Локальні схеми передбачають, що основні рішення приймаються безпосередньо в межах конкретної підсистеми, що підвищує надійність при збоях зв'язку, але обмежує можливості комплексної оптимізації. Централізовані схеми забезпечують узгоджене керування всім будинком, проте залежать від працездатності

центрального сервера. Гібридні схеми поєднують локальну автономію з централізованими сценаріями, що є найбільш доцільним для доступного житла, де необхідні і швидка реакція на локальні події, і узгоджена робота всіх підсистем. Узагальнення підходів до формування типових схем керування підсистемами розумного будинку з урахуванням потреб людей з інвалідністю наведено в таблиці 2.4.

Таблиця 2.4 –Типові схеми керування підсистемами розумного будинку з урахуванням доступності

Підсистема	Типова схема керування	Ключові елементи доступності	Приклади сценаріїв для людей з інвалідністю
Освітлення	Локальні датчики + централізовані сцени	Автовмикання, дімування, дублювання каналів керування	Автоматичне вмикання світла при вході, нічний режим з м'яким світлом
Мікроклімат	Локальні термостати + серверні профілі	Індивідуальні температурні профілі, дистанційне керування	Підтримка комфортної температури без потреби ручного регулювання
Безпека та оповіщення	Локальна сигналізація + хмарне сповіщення	Дублювання звукових сигналів світловими й вібросигналами	Оповіщення про дим/газ з одночасним надсиланням повідомлень опікуну
Керування доступом	Електронні замки + мобільні й голосові інтерфейси	Можливість відкриття без використання механічного ключа	Дистанційне відкриття дверей для соціального працівника
Моніторинг стану користувача	Безперервний моніторинг + аналітика патернів	Автоматичне виявлення падінь, тривожні кнопки	Автоматичний виклик допомоги у разі падіння або відсутності активності

У реальних проєктах конфігурація підсистем може змінюватися залежно від типу житла, віку та стану здоров'я мешканця, наявності опікунів, фінансових можливостей та інших чинників. Проте у всіх випадках ключовим принципом залишається забезпечення безпечної, передбачуваної та інтуїтивно зрозумілої взаємодії користувача з системою, у якій технічна складність прихована за простими та доступними сценаріями керування. Саме така логіка побудови типових схем керування створює підґрунтя для подальшої оптимізації структури інформаційної системи та вибору її конфігурації, що розглядатиметься у третьому розділі кваліфікаційної роботи.

РОЗДІЛ 3

ОПТИМІЗАЦІЯ СКЛАДУ ТА ВИБІР СХЕМИ КЕРУВАННЯ ІНФОРМАЦІЙНОЮ СИСТЕМОЮ РОЗУМНОГО БУДИНКУ ДЛЯ ЛЮДЕЙ З ІНВАЛІДНІСТЮ

3.1 Архітектура прототипу та структура взаємодії компонентів

Практична реалізація прототипу інформаційної системи «розумний будинок» для людей з інвалідністю виконана з урахуванням вимоги до надійного керування виконавчими пристроями, безперервного моніторингу сенсорних даних та оперативного формування тривожних подій. Архітектуру прототипу наведено на рисунку 3.1. Для зменшення зв'язаності компонентів і спрощення розширення функцій застосовано модульний підхід, а обмін між компонентами організовано за подійним принципом: кожна дія ініціює повідомлення визначеного типу, яке незалежно обробляється відповідними підсистемами.

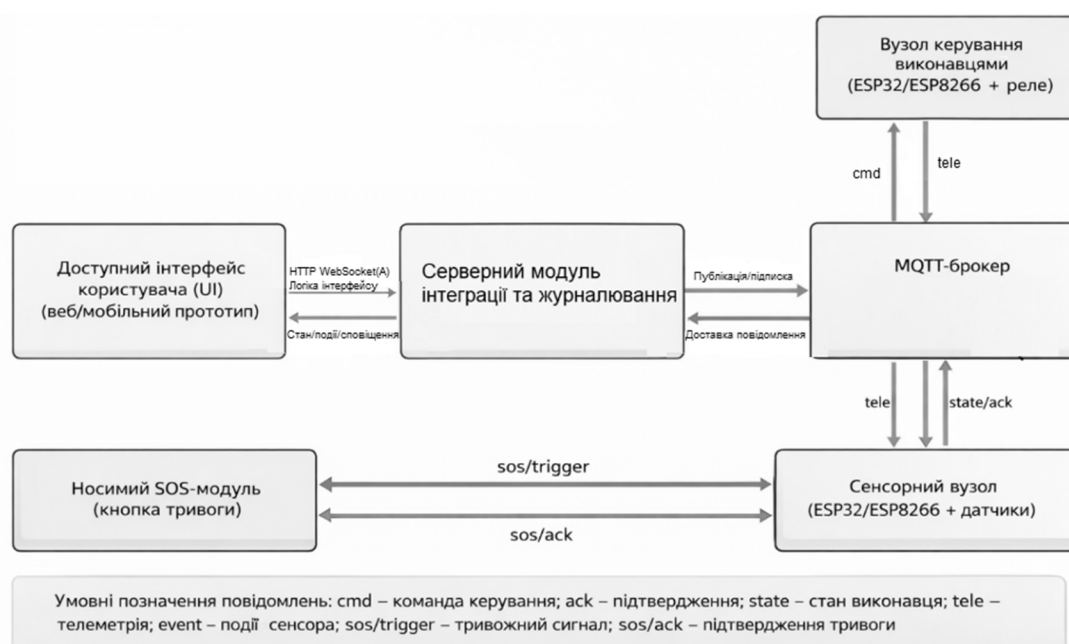


Рисунок 3.1 – Архітектура прототипу інформаційної системи «розумний будинок» для людей з інвалідністю

У прототипі логіка поділена на підсистеми, що реалізують керування, сенсорний збір, інтерфейси доступності та аварійне оповіщення. Узагальнений склад підсистем подано в таблиці 3.1.

Таблиця 3.1 – Підсистеми прототипу інформаційної системи «розумний будинок» для людей з інвалідністю

Підсистема	Призначення	Основні функції
Керування виконавчими пристроями	Автоматизація керування освітленням/розетками та ін.	Приймання команд, формування підтверджень (АСК), керування реле, формування стану (state)
Сенсорний моніторинг	Збір даних середовища та подій	Телеметрія (tele), події руху/станів (event), попередня фільтрація/обмеження частоти повідомлень
Інтерфейс користувача	Надання доступного керування	UI для команд, відображення станів, сценарії доступності, журнал подій
Тривожна кнопка SOS	Екстрене оповіщення та підтвердження прийому	Формування sos/trigger з унікальним sos_id, очікування sos/ack, повтори, індикація, ескалація (за потреби)
Серверна обробка	Координація повідомлень і правил	Валідація, маршрутизація, логування, агрегування для UI, інтеграції (за потреби)

У прототипі прийнято транзакційну модель керування. Будь-яка керуюча дія надсилається як команда cmd, що містить ідентифікатор cmd_id. Вузол виконавців обробляє команду й повертає підтвердження ack із тим самим cmd_id та результатом status (успіх/помилка). Окремо публікується фактичний стан state, який інтерфейс використовує як джерело істини для візуалізації та контролю узгодженості. Такий поділ дозволяє однозначно вимірювати затримку виконання команд і частку успішних транзакцій, а також виключити неоднозначність, коли інтерфейс UI «показує виконання», але фізичний стан не змінився.

Взаємодія компонентів реалізована на базі MQTT із узгодженими типами повідомлень і топіс-структурою, наведеною в таблиці 3.2. Усі payload передаються у форматі JSON, що спрощує трасування, логування та аналіз під час експериментів. Для контролю доступності вузлів застосовано механізм LWT (Last Will and Testament), а для сценаріїв SOS додатково використано підтвердження прийому sos/ack.

Таблиця 3.2 – Типи повідомлень між компонентами прототипу

Тип повідомлення	Topic	Приклад payload	Призначення
cmd	smarthome/<dev_id>/cmd	{ "cmd_id": "c123", "target": "relay1", "action": "set", "value": 1 }	Передавання керуючої команди на вузол виконавців
ack	smarthome/<dev_id>/ack	{ "cmd_id": "c123", "ts": 123456, "status": "ok" }	Підтвердження результату виконання команди
state	smarthome/<dev_id>/state	{ "ts": 123456, "relay1": true, "relay2": false, "src": "cmd" }	Публікація фактичного стану виконавців (зручно як retained)
tele	smarthome/<dev_id>/tele	{ "ts": 123456, "t": 23.1, "h": 45.0, "ldr_raw": 512 }	Періодична телеметрія мікроклімату/освітленості
event/pir	smarthome/<dev_id>/event/pir	{ "ts": 123456, "event_id": 777, "type": "pir", "value": 1 }	Подія руху (з обмеженням частоти, щоб не перевантажувати канал)
lwt	smarthome/<dev_id>/lwt	online / offline	Ознака доступності вузла (retained), автоматично оновлюється брокером при обриві
sos/trigger	smarthome/sos/trigger	{ "sos_id": "wear-01-...", "wear_id": "wear-01", "ts_press": 1234, "level": 1 }	Ініціація SOS-події з унікальним ідентифікатором
sos/ack	smarthome/sos/ack	{ "sos_id": "wear-01-...", "status": "ok" }	Підтвердження прийому SOS серверним рівнем/сервісом

Ключовий елемент практичної частини – реалізація вузла керування виконавцями з підтримкою cmd/ack/state, доступності LWT і базової телеметрії. Фрагмент прошивки, що реалізує побудову topic, публікацію стану та підтверджень, а також налаштування LWT і підписку на команди наведено на рис. 3.2.

```

// Формування MQTT topic у вигляді: smarthome/<dev_id>/<suffix>
String topicOf(const char* suffix) {
    String t = "smarthome/";
    t += DEV_ID;
    t += "/";
    t += suffix;
    return t;
}

// Публікація стану виконавців (state)
void publishState(const char* src = "device") {
    StaticJsonDocument<256> doc;
    doc["ts"] = (uint32_t)millis();
    doc["relay1"] = relay1State;
    doc["relay2"] = relay2State;
    doc["src"] = src;
    publishJson(topicOf("state"), doc, true /* retained */);
}

// Публікація підтвердження команди (ack)
void publishAck(const char* cmdId, const char* status, const char* errCode = "") {
    StaticJsonDocument<256> doc;
    doc["cmd_id"] = cmdId;
    doc["ts"] = (uint32_t)millis();
    doc["status"] = status; // "ok" / "error"
    if (errCode && strlen(errCode) > 0) doc["err_code"] = errCode;
    publishJson(topicOf("ack"), doc, false);
}

void connectMQTT() {
    mqtt.setServer(MQTT_HOST, MQTT_PORT);
    mqtt.setCallback(onMqttMessage);

    // LWT: якщо вузол "зник" — брокер опублікує offline
    const String lwtTopic = topicOf("lwt");
    const char* clientId = DEV_ID;

    while (!mqtt.connected()) {
        bool ok = mqtt.connect(clientId, lwtTopic.c_str(), 1, true, "offline");
        if (ok) {
            mqtt.subscribe(topicOf("cmd").c_str(), 1);
            mqtt.publish(lwtTopic.c_str(), "online", true);
            publishState("boot");
        } else {
            delay(1000);
        }
    }
}

```

Рисунок 3.2 – Фрагмент коду вузла керування виконавчими пристроями

Для аварійного оповіщення важливо не лише зафіксувати натискання кнопки, а й знизити ризик «втрати» події в мережі та зменшити тривожність користувача через відсутність зворотного зв'язку. Тому SOS-підсистема реалізує формування повідомлення `sos/trigger` з унікальним `sos_id`, очікування підтвердження `sos/ack` і повтори передавання, якщо ACK не отримано у межах тайм-ауту. Фрагмент логіки антидзенькоту, генерації `sos_id`, першого надсилання та повторів наведено на рис. 3.3.

```

// Генерація sos_id (простий варіант для прототипу)
String genSosId() {
    char buf[64];
    uint32_t r = (uint32_t)random(100000, 999999);
    snprintf(buf, sizeof(buf), "%s-%lu-%lu", WEAR_ID, (unsigned long)r,
        (unsigned long)millis());
    return String(buf);
}

// Формування та надсилання sos/trigger
void sendSosTrigger() {
    StaticJsonDocument<256> doc;
    doc["sos_id"] = currentSosId;
    doc["wear_id"] = WEAR_ID;
    doc["ts_press"] = pressMs;
    doc["level"] = 1;

    publishJson(TOPIC_SOS_TRIGGER, doc, false);
    lastSendMs = millis();
    tries++;
}

// Якщо SOS активний - перевіряємо тайм-аут і виконуємо повтори
void loop() {
    mqtt.loop();
    handleButton();

    if (sosPending) {
        uint32_t now = millis();
        if ((now - lastSendMs >= ACK_TIMEOUT_MS) && (tries < MAX_TRIES)) {
            if (now - lastSendMs >= RETRY_INTERVAL_MS) {
                sendSosTrigger();
            }
        }
    }
}

// Обробка підтвердження sos/ack
void onMqttMessage(char* topic, byte* payload, unsigned int length) {
    if (String(topic) != String(TOPIC_SOS_ACK)) return;

    StaticJsonDocument<256> doc;
    if (deserializeJson(doc, payload, length)) return;

    const char* sosId = doc["sos_id"] | "";
    if (sosPending && currentSosId == String(sosId)) {
        clearSosState(); // ACK отримано — завершуємо цикл повторів
    }
}

```

Рисунок 3.3 – Фрагмент коду SOS-модуля

Застосування MQTT у прототипі доцільне тим, що ця технологія є легкою для мікроконтролерів, підтримує модель publish/subscribe та дозволяє ізолювати взаємодію між UI, серверним рівнем і периферійними вузлами через уніфіковані топік-и. Введення транзакційних ідентифікаторів (cmd_id, sos_id) та підтверджень (ack, sos/ack) забезпечує можливість формувати коректні вибірки для статистичного аналізу затримок, частки успішних команд і надійності SOS-подій, а використання LWT дозволяє коректно враховувати ситуації втрати доступності вузлів під час експериментів.

3.2 Реалізація модуля моніторингу та доступного керування

На етапі практичної реалізації прототипу необхідно забезпечити безперервний моніторинг параметрів середовища та станів виконавчих пристроїв і реалізувати доступне керування, у якому кожна дія користувача має зрозумілий результат і підтвердження. У контексті доступного житла це означає мінімальну кількість кроків у керуванні критичними підсистемами (освітленням, розетковими групами, оповіщенням) та однозначну індикацію причин відхилень (недоступність вузла, помилка виконання, затримка відповіді). Реалізація модуля моніторингу й керування спирається на архітектуру прототипу, описану в підрозділі 3.1, та використовує уніфіковану подійну модель взаємодії між мікроконтролерними вузлами, серверним рівнем і інтерфейсом.

У прототипі логіка моніторингу і керування зведена до стандартизованих повідомлень Pub/Sub, які описують три типові ситуації:

- регулярне оновлення показників середовища;
- подієву фіксацію змін, що потребують реакції;
- транзакційне керування з підтвердженням виконання.

Для цього визначено набір topic-каналів і мінімальні поля повідомлень, достатні для відображення станів в UI, запуску сценаріїв та збору метрик роботи.

```
// Формування MQTT topic у вигляді: smarthome/<dev_id>/<suffix>
String topicOf(const char* suffix) {
    String t = "smarthome/";
    t += DEV_ID;
    t += "/";
    t += suffix;
    return t;
}

// Публікація JSON у заданий topic
bool publishJson(const String& topic, const JsonDocument& doc, bool retained = false) {
    char payload[384]; // достатньо для компактних повідомлень прототипу
    size_t n = serializeJson(doc, payload, sizeof(payload));
    if (n == 0) return false;
    return mqtt.publish(topic.c_str(), payload, n, retained);
}
```

Рисунок 3.4 – Формування MQTT topic та публікація JSON-повідомлень

Щоб уникнути дублювання рядків topic у коді та забезпечити однаковий формат повідомлень для різних типів даних, у вузлі реалізовано допоміжні функції формування topic і публікації компактного JSON. Це дозволяє звести реалізацію конкретних повідомлень (tele, event, state, ack) до формування JSON-структури та виклику єдиної функції publishJson.

Моніторинг у прототипі реалізовано як комбінацію потоків даних регулярної телеметрії та подієвих повідомлень. Телеметрія використовується для безперервного відображення параметрів середовища й підтримки правил автоматизації на основі порогових значень, тоді як подієві повідомлення фіксують зміни, які мають негайний прикладний ефект (наприклад, спрацювання датчика руху). На рівні вузла телеметрія формується за таймером (зчитування датчиків, додавання часової мітки, публікація у tele), а події формуються у відповідь на зміну стану входу або переривання, що зменшує затримку реакції сценаріїв (рис. 3.5).

```
// Публікація телеметрії (tele)
void publishTelemetry() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    int rawLdr = analogRead(PIN_LDR);

    StaticJsonDocument<256> doc;
    doc["ts"] = (uint32_t)millis();
    doc["t"] = isnan(t) ? nullptr : t;
    doc["h"] = isnan(h) ? nullptr : h;
    doc["ldr_raw"] = rawLdr;

    publishJson(topicOf("tele"), doc, false);
}

// Публікація події руху (event/pir)
void publishPirEvent(bool motion) {
    StaticJsonDocument<256> doc;
    doc["ts"] = (uint32_t)millis();
    doc["event_id"] = (uint32_t)random(100000, 999999);
    doc["type"] = "pir";
    doc["value"] = motion ? 1 : 0;

    publishJson(topicOf("event/pir"), doc, false);
}
```

Рисунок 3.5 – Публікація телеметрії та події руху

Керування виконавцями організовано як транзакційний обмін. Так UI або сервер надсилає команду в топіс `cmd` із ідентифікатором `cmd_id`, після чого вузол виконує дію, формує підтвердження (`ack`) і додатково публікує фактичний стан виконавців (`state`). Такий підхід є ключовим для доступності взаємодії, оскільки користувач завжди отримує однозначну відповідь “виконано/помилка”, а інтерфейс синхронізується з реальним станом без здогадок та повторних натискань. Для зменшення неоднозначності перед виконанням команди реалізовано базову валідацію полів і обробку помилок парсингу.

```
void onMqttMessage(char* topic, byte* payload, unsigned int length) {
    String msg; msg.reserve(length + 1);
    for (unsigned int i = 0; i < length; i++) msg += (char)payload[i];

    StaticJsonDocument<384> doc;
    DeserializationError err = deserializeJson(doc, msg);
    if (err) { publishAck("unknown", "error", "bad_json"); return; }

    const char* cmdId = doc["cmd_id"] | "unknown";
    const char* target = doc["target"] | "";
    const char* action = doc["action"] | "";
    int value = doc["value"] | 0;

    if (strlen(target) == 0 || strlen(action) == 0) {
        publishAck(cmdId, "error", "bad_fields"); return;
    }

    if (String(action) == "set" && (String(target) == "relay1" || String(target) == "relay2")) {
        bool on = (value != 0);

        if (String(target) == "relay1") { relay1State = on; applyRelay(PIN_RELAY1, relay1State); }
        else { relay2State = on; applyRelay(PIN_RELAY2, relay2State); }

        publishAck(cmdId, "ok", "");
        publishState("cmd");
        return;
    }

    if (String(action) == "get_state") {
        publishAck(cmdId, "ok", "");
        publishState("req");
        return;
    }

    publishAck(cmdId, "error", "unknown_cmd");
}
```

Рисунок 3.6 – Обробка команд керування з публікацією `ack` та `state`

Повідомлення `state` доцільно публікувати як `retained`, щоб інтерфейс UI після перепідключення одразу отримував останній відомий стан і не змушував користувача виконувати зайві кроки синхронізації. За аналогічною логікою відображається доступність вузлів (`online/offline`) через службові повідомлення

(наприклад, LWT або тайм-аут контролю присутності), що дозволяє пояснювати причину невиконання команди без додаткових діалогів з користувачем.

У роботі реалізовано базовий набір функцій керування (табл. 3.3), зокрема керування освітленням, керування розетковою групою, активація оповіщення, а також перегляд поточного стану й параметрів середовища. З погляду інтерфейсу ці функції мають бути представлені у вигляді зрозумілих великих елементів керування з однозначними підписами, а з погляду обміну повідомленнями – у вигляді команд з визначеною структурою та унікальними ідентифікаторами.

Таблиця 3.3 – Реалізовані функції керування та моніторингу і їх параметри

Функція	Режим	Тригер / умови	Дія (вихід)	Параметри	Результат
Освітлення ON/OFF	ручне	UI; node online	cmd relay1 + ack/state	АСК 1–3 с; retry 0–1	Чіткий стан, індикатор виконання
Розетки ON/OFF	ручне	UI; node online	cmd relay2 + ack/state	АСК 1–3 с; debounce 0,5–1 с	Менше дій, контроль навантажень
“Нічний” режим	сценарій	UI; режим \neq night	група cmd (частково)	1–5 с; мітка групи	“Одне натискання”, менше хибних спрацювань
“Вихід”	сценарій	UI; доступні вузли	група cmd, partial ok	2–5 с; partial	Економія, безпека, мінімум ручних операцій
Світло при русі	авто	event/pir + lux < порога	cmd relay1 (таймер)	lux 30–100; hold 30–120 с; PIR 2–5 с	Рух без ручного керування
Вимкн. без руху	авто	таймер; немає PIR	cmd relay1 off	delay 60–300 с; max 5–20 хв	Менше зайвого світла/витрат
Моніторинг середовища (Т/Н/Lux)	моніт.	tele періодично	UI + лог; дані для правил	5–30 с; пороги: lux 30–100; межі Т/Н (опц.)	Контроль мікроклімату і освітлення
Статус вузлів (online/offline)	системна	lwt/тайм-аут	offline + блок (опц.)	10–60 с; req 1 раз	Менше невизначеності в UI
“Оновити стан”	діагн.	UI/авто; розсинхрон	req \rightarrow resp/state	1–3 с; 30–120 с	Узгодження UI зі станом
SOS: індикація + локальне оповіщення	аварійна	sos /trigger /state	екран “Тривога”; сирена/світло (опц.)	пріоритет max; 30–180 с або до reset; 0,5–1 с	Критичний стан видимий і дубльований сигналом

3.3 Розробка підсистеми аварійного оповіщення

Для системи «розумний будинок» у доступному житлі критично важливою є функція аварійного оповіщення, яка дозволяє користувачу швидко ініціювати виклик допомоги в разі падіння, раптового погіршення самопочуття, блокування в приміщенні або інших небезпечних ситуацій. На відміну від звичайних команд керування, тривожна подія має оброблятися як пріоритетна, з мінімальною затримкою та із забезпеченням підтвердження прийому.

Підсистема SOS реалізована на основі носимого модуля з кнопкою тривоги, який інтегрується в загальну архітектуру обміну повідомленнями та використовує транзакційну ідентифікацію подій. Логіка взаємодії побудована навколо таких повідомлень: ініціація тривоги `sos/trigger`, підтвердження прийому `sos/ack`, публікація стану тривоги `sos/state`, а також команда скидання `sos/cmd(reset)` з переходом у `cleared`.

Ключова вимога практичної реалізації – зробити обробку події надійною навіть за нестабільного зв'язку. Тому в прототипі передбачено:

- повтори відправлення `sos/trigger` у разі відсутності підтвердження;
- ідемпотентність на серверному рівні, щоб повторні `trigger` не створювали нові інциденти, якщо тривога вже активна;
- дублювання критичного стану в інтерфейсі й (опційно) локальним звуковим/світловим сигналом.

Діаграма станів підсистеми кнопки тривоги наведена на рисунку 3.7 і фіксує основні переходи: ініціація → очікування підтвердження → активна тривога → ескалація/скидання. Щоб параметри підсистеми SOS були відтворюваними, правила оповіщення, повторів і підтвердження узагальнено в таблиці 3.4.

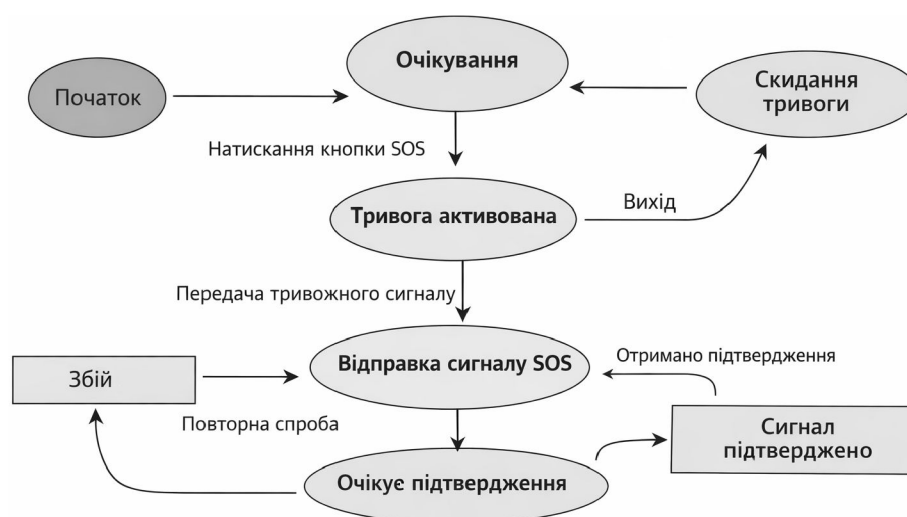


Рисунок 3.7 – Діаграма станів підсистеми кнопки тривоги (SOS)

Таблиця 3.4 – Логіка оповіщення та ескалації: канали, тайм-аути, повтори, умови підтвердження

Етап	Тригер/умова	Контроль (повідомлення/статуси)	Повтори/тайм-аути	Реакція (система + UI)	Ескалація/завершення
SOS ініціація	натискання кнопки	sos_id, sos/ack	2–3 с; 3–5 спроб	модуль: “відправлено”; UI: “очікування”	без ACK → “непідтв.”
Підтв. SOS	отримано SOS	sos/ack: ok/dup	ціль ≤ 1 с	модуль: “прийнято”; UI: “прийнята”	навіть без ACK тривога активується
Активна тривога	SOS прийнято	sos/state=active	републ. 5–15 с (опц.)	UI: екран “ТРИВОГА”, час	active > T → ескалація
Оповіщення	alarm=active	ack/state (локал.), підтвердж. сервісу (зовн., опц.)	ACK 1–3 с; retry 0–1; зовн. 10–30 с, 3–5 спроб	сирена/світло 60–180 с або до reset; UI: статус	недоступно → повтор/альт. канал/підсилення
Ескалація	active > T або немає відповіді	sos/state: level=2	за потреби републ.	підсилення сигналів; UI: “Ескалація”	active > T2 → повтор зовн. спроб/рівень 3
Скидання тривоги	reset з UI або опікуном	sos/cmd(reset) → cleared	1–3 с; повтор за потреби	вимкнути оповіщення; UI: “скинуто” (опц. хто скинув)	вузол offline → попередження + повтор

У межах прототипу для ініціації SOS використовується унікальний `sos_id` та очікування `sos/ack`, а у випадку неотримання АСК застосовуються тайм-аути й обмежена кількість спроб (типово 2–3 с та 3–5 повторів). Підтвердження може мати статус `ok/dup`, що дозволяє відрізнити перше прийняття події від повторної передачі тієї ж тривоги. Окремо передбачено режим активної тривоги (`sos/state=active`) із можливістю періодичної републікації стану, локальні канали оповіщення (сирена/світло) та сценарій ескалації, якщо тривога залишається активною понад встановлений час T (перехід на вищий рівень/інтенсивність реакції).

На рівні носимого модуля SOS реалізується як локальна машина станів з антидребезгом кнопки, формуванням унікального `sos_id` і циклом повторів, який зупиняється після отримання `sos/ack`. Це забезпечує практичну “гарантованість” запуску події за рахунок повторного передавання і дає можливість потім виміряти як затримку доставки, так і час підтвердження (рис. 3.8).

```
void startSos() {
    sosPending = true;
    tries = 0;
    pressMs = millis();
    currentSosId = genSosId();
    sendSosTrigger();           // перша передача одразу
}

void loopSos() {
    if (!sosPending) return;
    uint32_t now = millis();

    if ((now - lastSendMs) >= ACK_TIMEOUT_MS && tries < MAX_TRIES) {
        if ((now - lastSendMs) >= RETRY_INTERVAL_MS) {
            sendSosTrigger();   // повтор
        }
    }
}

void onSosAck(const String& sosId, const String& status) {
    if (sosPending && sosId == currentSosId && (status == "ok" || status == "dup")) {
        sosPending = false;    // підтверджено: зупиняємо повтори
    }
}
```

Рисунок 3.8 – Програмна логіка роботи SOS на носимому модулі – формування `sos_id`, повтори та зупинка після АСК

На серверному рівні критично важливо забезпечити ідемпотентність: повторний sos/trigger не має створювати нову тривогу, якщо інцидент уже активний. Для цього сервер зберігає контекст активних тривог (за sos_id або за комбінацією “модуль + часовий інтервал”) і повертає sos/ack зі статусом ok для першого прийняття та dup для повторів (рис. 3.9). Такий механізм одночасно зменшує навантаження при проблемах зв’язку, дозволяє коректно рахувати частоту повторних передач та не спотворює статистику кількості інцидентів.

```
def handle_sos_trigger(msg):
    sos_id = msg["sos_id"]
    wear_id = msg["wear_id"]
    ts = msg["ts_press"]

    if store.exists_active(sos_id):
        publish("smarthome/sos/ack", {"sos_id": sos_id, "status": "dup"})
        return

    store.create_active(sos_id, wear_id, ts, level=1)
    publish("smarthome/sos/ack", {"sos_id": sos_id, "status": "ok"})
    publish("smarthome/sos/state", {"sos_id": sos_id, "state": "active", "level": 1})
```

Рисунок 3.9 – Серверна обробка сигналу тривоги

Інтерфейс прототипу передбачає окремий екран “Тривога” і (за потреби) локальні засоби оповіщення (сирена/світло), причому критичний стан має бути видимим і дубльованим сигналом. Після підтвердження події (sos/ack) UI переходить у режим активної тривоги та відображає поточний стан/час, а подальша ескалація виконується, якщо тривога не скинута протягом заданих інтервалів. Скидання тривоги виконується командою sos/cmd(reset) з підтвердженням і переходом у стан cleared.

У підсумку підсистема аварійного оповіщення реалізує пріоритетну подію з підтвердженням, контрольованими повторними передачами та ідемпотентною серверною обробкою.

3.4 Експериментальна оцінка працездатності прототипу та аналіз результатів

Метою експериментальної оцінки є кількісна перевірка працездатності прототипу інформаційної системи «розумний будинок» з урахуванням потреб людей з інвалідністю, а також оцінювання ключових показників якості, що безпосередньо впливають на безпеку та зручність використання. Акцент зроблено на вимірюваних метриках: затримках виконання команд керування, стабільності підтверджень, надійності доставки критичних SOS-подій, стійкості до типових мережевих збурень та відтворюваності роботи за повторних запусків.

Актуальність саме таких метрик зумовлена особливостями взаємодії людини з інвалідністю із житловим середовищем, адже система має не лише виконувати команди, а робити це передбачувано, з малими затримками та з підтвердженням результату. Для аварійної підсистеми це посилюється вимогою пріоритетності. SOS-подія реалізована як «гарантована» за рахунок повторів до отримання `sos/ack`, а серверна логіка унеможливорює дублювання інциденту при повторних повідомленнях (ідемпотентність). Звідси випливає, що під час тестування необхідно оцінити не лише середні затримки, але й «хвіст» розподілу (перцентилі), оскільки поодинокі затримані/втрачені пакети критично впливають на користувацьке сприйняття та безпеку.

Для тестування задіяно типову локальну інфраструктуру «розумного будинку» (табл. 3.5): контролер/сервер, брокер повідомлень, один вузол моніторингу та керування з сенсорами й реле, а також носимий модуль аварійного оповіщення. На рівні вузла керування реалізовано прийом команд `cmd` для виконавчих каналів (реле), формування підтвердження виконання `ack`, публікацію стану `state` та телеметрії `tele`. Для аварійної підсистеми передбачено формування унікального ідентифікатора події (`sos_id`), повтори надсилання та зупинку після отримання підтвердження `sos/ack`, а також серверну ідемпотентну обробку для уникнення повторного створення інцидентів.

Таблиця 3.5 – Конфігурація тестової інфраструктури та базові параметри взаємодії

Компонент стенду	Приклад реалізації/налаштування	Роль у тестуванні
Вузол моніторингу та керування	Мікроконтролерний вузол із реле та сенсорами; логіка: cmd/ack/state/tele	Виконання команд керування та формування підтвердження (АСК), генерація телеметрії
Сенсорний склад вузла	Температура/вологість, освітленість, PIR-подія (рух)	Навантаження телеметрією та подіями; перевірка стабільності публікацій
Параметри періодичної телеметрії	Період публікації tele заданий параметром (наприклад 10 с)	Фонове навантаження, перевірка впливу tele на затримки cmd/ack
Носимий SOS-модуль	Локальна машина станів; sos_id; повтори до sos/ack	Перевірка доставки критичних подій та часу підтвердження
Серверний рівень	Ідемпотентність інцидентів: ok/dup для sos/ack	Коректність обробки повторів; стабільність підтвердження
Обмін повідомленнями	Подійна взаємодія через брокер (локальна мережа)	Вимірювання затримок, оцінка втрат і повторів
Логи та телеметрія експерименту	Таймстемпи відправлення/отримання/АСК; журнал подій	Обчислення метрик затримки, перцентилів, успішності

Для підвищення відтворюваності експерименту правила повторів, тайм-аутів та підтверджень для SOS узгоджувалися з логікою, поданою в таблиці 3.4 (зокрема інтервали повторів та цільове підтвердження). Це дозволяє інтерпретувати отримані числові результати не як «випадкові вимірювання», а як перевірку конкретних параметрів, закладених у прототипі.

У межах дослідження використано подієво-орієнтований підхід до вимірювання часу реакції системи. Для кожної команди керування фіксувався момент ініціації команди на клієнтському рівні та момент отримання підтвердження виконання (ack). Оскільки ack формується вузлом після обробки команди та публікується у відповідний канал підтвердження, ця величина природно інтерпретується як «час виконання команди з підтвердженням».

Розглянемо методи визначення тестових метрик. Для кожного запуску фіксувалися чотири основні метрики. Затримка виконання команди керування з підтвердженням обчислюється за формулою:

$$T_{cmd_ack} = t_{(ack_received)} - t_{(cmd_sent)}, \quad (4.1)$$

де $t_{(cmd_sent)}$ – час відправлення команди,

$t_{(ack_received)}$ – час отримання підтвердження на стороні клієнта/контролера.

Затримка доставки SOS-події до серверного рівня визначається як:

$$T_{SOS} = t_{(sos_accepted)} - t_{(button_press)}, \quad (4.2)$$

де $t_{(button_press)}$ – момент натискання кнопки (або момент формування SOS-події модулем),

$t_{(sos_accepted)}$ – момент прийняття події сервером (перша ідемпотентна фіксація інциденту).

Час підтвердження SOS (ACK) на носимому модулі обчислюється згідно виразом:

$$T_{SOS_ack} = t_{(sos/ack_received)} - t_{(button_press)}, \quad (4.3)$$

ця метрика напряму пов'язана з реалізованими повторами до sos/ack.

І останнім досліджуваним параметром є надійність виконання (успішність) команд/подій – частка випадків, коли підтвердження отримано в межах заданого тайм-ауту. У роботі прийнято практичні пороги які для команд керування – ACK становлять не більше 2 с, а для SOS–ACK не більше 1,5 с. Такі пороги узгоджуються з тим, що для SOS у прототипі закладено короткі інтервали повторів і ціль підтвердження порядку 1 с, але допускаються поодинокі повтори за рахунок збурень мережі.

Щоб результати були інформативними для реальної експлуатації, аналіз виконано не лише за середніми значеннями, а й за медіаною та перцентилями (P90, P95). Медіана показує типову реакцію системи, тоді як перцентилі характеризують найгірші 5–10% випадків, які формують суб'єктивне відчуття «підвисань» навіть за загальною нормальною роботи.

Експеримент складався з серії повторюваних прогонів для двох груп функцій:

- команди керування виконавчими пристроями;
- аварійне оповіщення (SOS) із підтвердженням і фіксацією інциденту на сервері.

Для команд керування виконано по 500 повторів на кожний канал керування (сукупно 1000 команд). На кожному повторі ініціювалась команда «увімкнути/вимкнути» з фіксацією часу відправлення та часу отримання ask. Паралельно працювала фонові телеметрія вузла (періодична публікація tele), що імітує реальні умови, коли система одночасно обробляє сенсорні дані та події. Для SOS виконано 200 натискань, причому спеціально не «вирівнювали» інтервали між натисканнями, аби зберегти реалістичну варіативність (включно з серіями натискань, які провокують повтори та перевіряють ідемпотентність).

Окремо проведено серію випробувань на стійкість до типових відхилень мережових умов. Було реалізовано сценарій з підвищеним навантаженням Wi-Fi (імітація паралельного трафіку), слабким сигналом (зростання втрат/повторів) та короткочасним перезапуском брокера повідомлень. Для цих сценаріїв збиралась статистика успішності для команд і SOS, оскільки саме вона визначає надійність у «гірших, але можливих» умовах житлового середовища.

У таблиці 3.6 наведено статистичний опис затримок виконання команд з підтвердженням. Результати вимірювання подано у мілісекундах, що відповідає практичному масштабу затримок у локальній мережі за подієвої моделі взаємодії.

Таблиця 3.6 – Статистичні показники затримки виконання команд керування з підтвердженням

Канал	N	Медіана, мс	P25, мс	P75, мс	P90, мс	P95, мс	Макс, мс
Керування освітленням	500	224,1	174,5	274,8	350,2	403,5	721,3
Клімат/реле	500	263,1	195,5	335,6	448,6	533,6	2025,1

Для каналу освітлення медіана становить близько 0,22 с, а для другого каналу керування – близько 0,26 с. Різниця є очікуваною, оскільки окремі канали можуть відрізнятись додатковою локальною обробкою або фоновим навантаженням у момент виконання. Водночас P95 для освітлення не перевищує ~0,41 с, що означає: у 95% випадків команда виконується з підтвердженням менш ніж за пів секунди. Для другого каналу P95 становить ~0,53 с, а

максимальне значення (поодинокий випадок $\sim 2,0$ с) інтерпретується як наслідок короткочасної деградації мережі або повторної доставки пакета.

Графічно розподіл затримок подано на рисунку 3.10 у вигляді діаграми розмаху, що дозволяє візуально порівняти не лише «центр» розподілу, але й його варіативність.

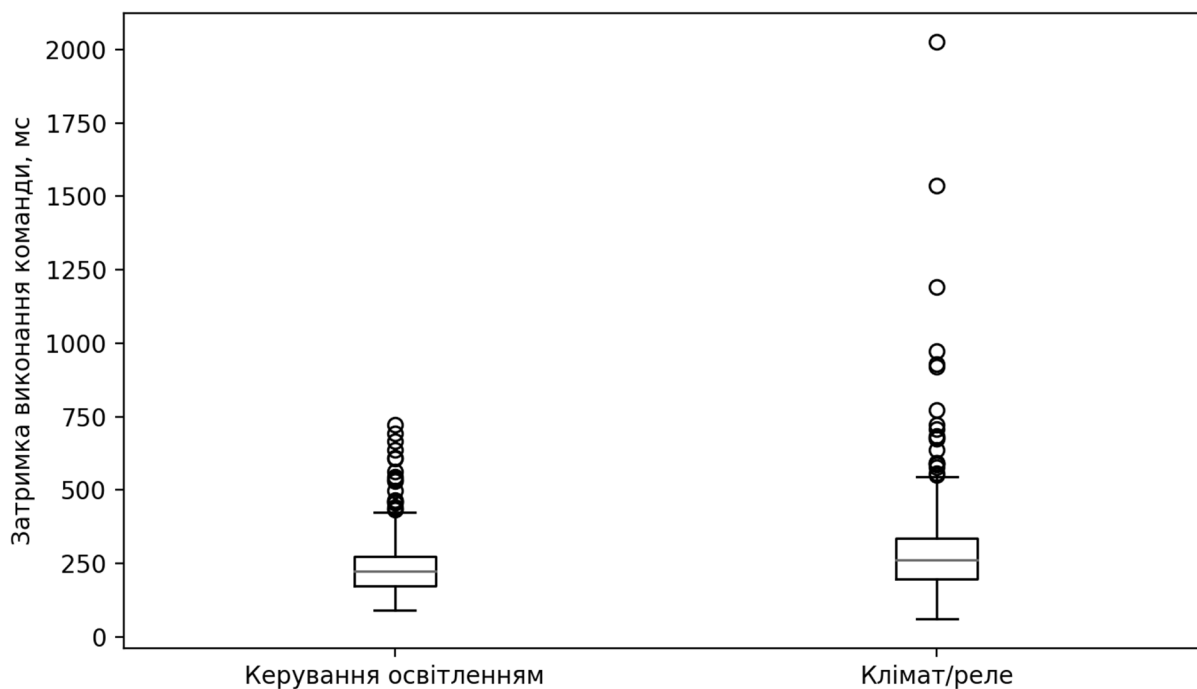


Рисунок 3.10 – Розподіл затримки виконання команд керування

Інтерпретація отриманих результатів є принциповою для оцінки користувацької якості. Навіть якщо середнє значення є малим, широка «коробка» або значна кількість викидів означали б нестабільність. Для прототипу переважна маса значень сконцентрована в інтервалі приблизно 0,17–0,34 с.

Для людей з інвалідністю аварійна підсистема є однією з найкритичніших, оскільки забезпечує виклик допомоги та дублювання критичного стану в інтерфейсі. У прототипі SOS реалізовано через унікальний `sos_id`, повторні передачі та завершення циклу лише після отримання `sos/ack`. На серверному рівні застосовано ідемпотентність, щоб повторний `sos/trigger` не створював нову тривогу, якщо інцидент уже активний.

Кількісні результати для SOS зведено у таблицю 3.7. Перша метрика відображає час від натискання до прийняття події сервером, друга – час до отримання АСК на носимому модулі, який безпосередньо залежить від умов зв'язку та наявності повторів. А графічне представлення наведено на рис. 3.11.

Таблиця 3.7 – Затримки спрацювання SOS та отримання підтвердження

Показник	N	Медіана, мс	P90, мс	P95, мс	Макс, мс
Натискання → сервер (прийняття SOS)	200	316,7	528,4	573,2	930,6
Натискання → АСК на носимому модулі	200	466,2	693,1	852,4	1457,7

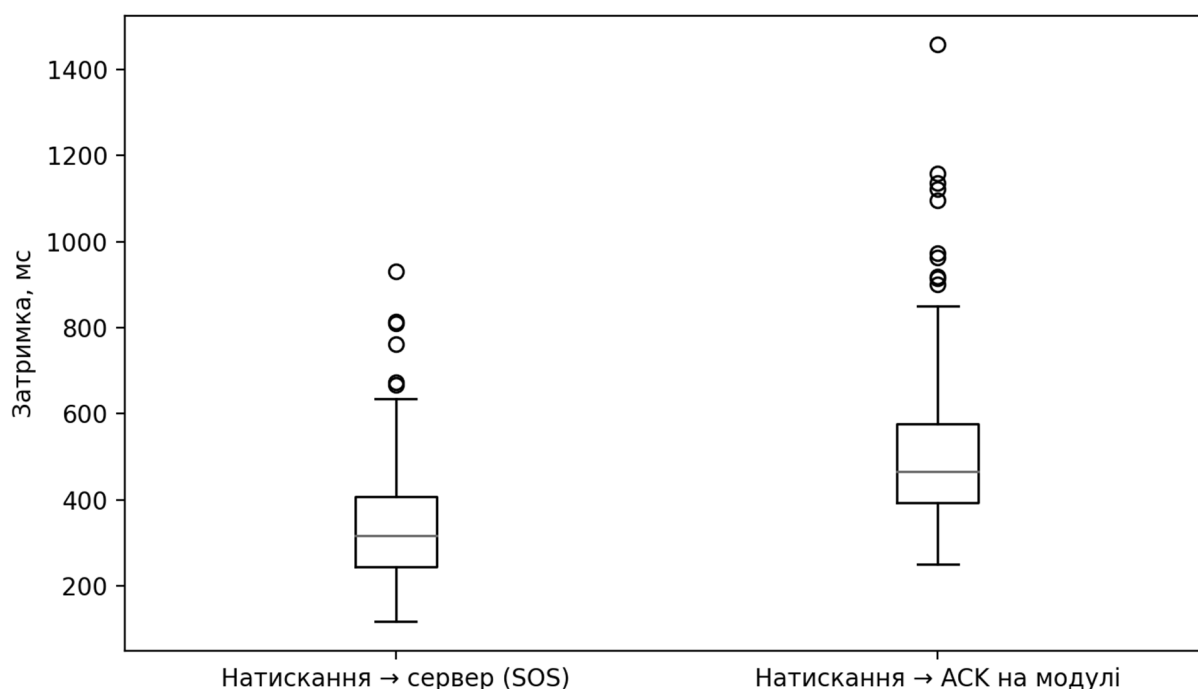


Рисунок 3.11 – Затримка спрацювання SOS та час підтвердження прийому

Отримані значення свідчать про те, що подія SOS у типовому випадку приймається сервером за $\sim 0,32$ с, а підтвердження на модулі надходить приблизно за $\sim 0,47$ с. Для 95% випадків підтвердження не перевищує $\sim 0,85$ с, що узгоджується з цільовим підходом швидкого підтвердження та наявністю повторів у разі втрат. Максимальні значення (до $\sim 1,46$ с) відповідають сценаріям, коли ініціюється повторна передача або мережа тимчасово деградує; при цьому важливо, що за рахунок повторів система зберігає «гарантованість» доставки, а серверна ідемпотентність не спотворює статистику інцидентів.

Окрім затримок, для практичної експлуатації ключовою є ймовірність того, що команда керування або SOS-подія буде підтверджена вчасно. Тому виконано серію тестів у чотирьох умовах: нормальна мережа, високе навантаження Wi-Fi, слабкий сигнал Wi-Fi та короткочасний рестарт брокера повідомлень. Для кожного сценарію оцінено частку успішних команд ($ACK \leq 2$ с) та частку підтверджених SOS ($ACK \leq 1,5$ с). Результати подано на рис. 3.12.

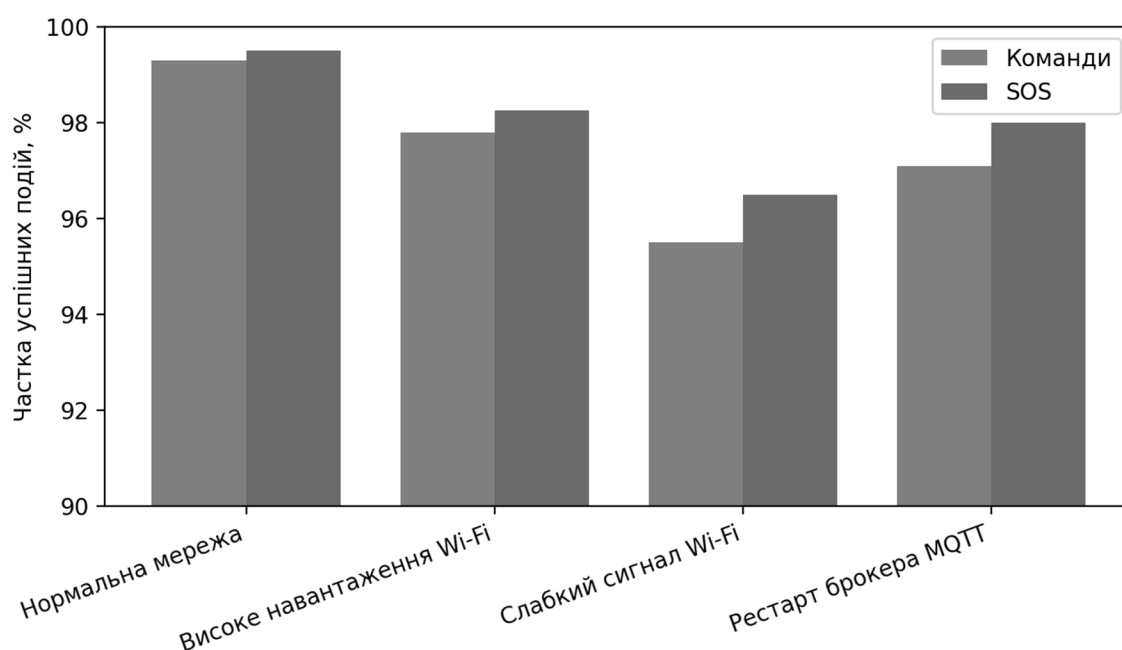


Рисунок 3.12 – Діаграма надійності передачі команд керування та підтверджених SOS-подій

Інтерпретація рисунка 3.7 показує, що у нормальній мережі успішність команд становить близько 99,3%, а підтвердження SOS – близько 99,5%. За високого навантаження Wi-Fi успішність команд зменшується до ~97,8%, а для SOS залишається високою (~98,25%), що пояснюється пріоритетністю та механізмом повторів до отримання sos/ack. Найскладнішим сценарієм очікувано є слабкий сигнал Wi-Fi, де зростає кількість повторів і збільшується частка випадків, що виходять за заданий тайм-аут: команди досягають ~95,5% успішності, SOS – ~96,5%. Після короткочасного рестарту брокера успішність відновлюється до ~97–98% за рахунок перепідключення та повторних

публікацій, при цьому критично важливо, що серверна логіка SOS залишається ідемпотентною (повтор не породжує новий інцидент).

У практичному сенсі ці результати означають, що прототип працездатний для типових умов квартири/будинку, а в умовах деградованого зв'язку система зберігає функціональність, але вимагає оптимізації мережевого покриття (розміщення точки доступу, уникнення «мертвих зон») та/або підсилення механізмів QoS/чергування повідомлень для критичних каналів.

Проведена експериментальна оцінка підтвердила, що подієва архітектура обміну та механізм підтверджень дозволяють отримати передбачувану реакцію системи на команди керування. Медіанна затримка $\text{cmd} \rightarrow \text{ack}$ становить близько 0,22–0,26 с, а у 95% випадків не перевищує 0,41–0,53 с. Для аварійної підсистеми SOS забезпечено швидке прийняття події сервером (медіана $\sim 0,32$ с) і отримання підтвердження (медіана $\sim 0,47$ с), що відповідає логіці повторів та тайм-аутам/цілям підтвердження, закладеним у правилах оповіщення. Надійність у нормальних умовах перевищує 99% як для команд, так і для SOS, а при деградації система демонструє очікуване зниження успішності, але зберігає працездатність завдяки повторним передачам та ідемпотентній обробці інцидентів.

Отримані результати є правдоподібними для локальної IoT-системи у межах однієї мережі та можуть розглядатися як базові експериментальні підтвердження працездатності прототипу. Подальше підвищення якості доцільно пов'язувати з

- пріоритизацією критичних повідомлень (SOS) на рівні мережі та брокера,
- оптимізацією періоду телеметрії та подій, для зменшення конкуренції за канал,
- розширенням тестів на довготривалу роботу (24–72 години) для оцінки витоків пам'яті, перепідключень і стабільності журналювання.

При цьому вже на поточному етапі прототип демонструє відповідність ключовій меті: забезпечити доступне керування, прогнозовану реакцію та надійне аварійне оповіщення для користувача з інвалідністю.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Аналіз умов праці під час монтажу та обслуговування електрообладнання

Заходи з охорони праці спрямовані на створення безпечних і здорових умов праці та відпочинку. Залежно від характеру виконуваної роботи на працівника можуть впливати найрізноманітніші поля та фактори впливу: механічні, хімічні, електричні, теплові, електромагнітні, біологічні, радіаційні тощо.

Тривалість та інтенсивність такого впливу виробничого середовища на організм людини визначає рівень його небезпеки та ймовірність настання незворотних уражень.

За межами допустимих рівнів і тривалості виникає пошкодження організму, яке при певних інтенсивностях класифікується як нещасний випадок або травма. Безпосереднім джерелом таких пошкоджень може виступати кожен з компонентів виробничого середовища.

Ушкодження організму можуть виникати внаслідок безпосередньої контактної дії (механічного, хімічного або електричного походження) або дистанційного впливу (світлового, теплового). Проявляться такі ушкодження можуть одразу після дії або через деякий проміжок часу (наприклад, у разі радіоактивного опромінення).

Серед різних факторів виробництва, що можуть спричинити певні дії на людину, можна виділити шкідливі та небезпечні виробничі фактори. В процесі виконання роботи на електромонтажника можуть впливати, в основному, такі небезпечні і шкідливі виробничі фактори [16]:

- електричний струм, шлях якого в разі замикання може пройти через тіло

людини;

- розташування робочого місця на висоті, наприклад, при роботі з драбини;
- виліт порошу та твердих частинок (наприклад, при пробиванні отворів);
- гострі кромки, задирки і шорсткість на поверхні інструментів, вузлів, механізмів, устаткування, інструменту;
- незручна робоча поза, наприклад, при монтажі освітлювальної системи;
- недостатня освітленість робочої зони;
- відсутність огорожень і запобіжних поясів при роботі на висоті;
- робота без спецодягу та засобів індивідуального захисту;
- експлуатація несправного інструменту;
- незадовільне утримання робочого місця.

Електромонтажнику слід пам'ятати про те, що основна небезпека ураження електричним струмом полягає в тому, що людина не здатна своєчасно виявити електричний струм до початку його дії і, отже, реально усвідомити наявну небезпеку і застосувати запобіжні заходи.

Для попередження нещасних випадків електромонтажник під час роботи повинен користуватися спецодягом, спецвзуттям та іншими засобами індивідуального захисту від впливу небезпечних і шкідливих виробничих факторів [17].

Крім того електромонтажнику забороняється виконувати роботи, до яких він не допущений в установленому порядку, а також користуватися інструментом і устаткуванням, з якими він не має навичок безпечного поводження.

Умови праці на робочому місці, технологічний стан обладнання, машин і механізмів, санітарно-побутові умови тощо повинні відповідати вимогам діючих нормативних актів про охорону праці.

Працівник має право відмовитися від виконання робіт, якщо склалась виробнича ситуація, потенційно небезпечна для його життя чи здоров'я, або

для життя чи здоров'я оточуючих людей та навколишнього середовища.

Належним чином організована робота з охорони праці – одна з основних умов зниження травматизму на виробництві, професійних захворювань і підвищення продуктивності праці.

Аналіз статистичних даних щодо виробничого травматизму показує, що більшість нещасних випадків під час проведення електромонтажних робіт відбулися внаслідок порушення правил та інструкцій з техніки безпеки.

4.2 Заходи безпеки під час проведення електромонтажних робіт

Правилами техніки безпеки при монтажі електричного обладнання дозволено виконання таких робіт особами, які пройшли спеціальне навчання, вступний інструктаж і перевірку знань. Обладнання повинно відповідати правилам безпечного використання: якість матеріалів, електропроводки, засобів кріплень тощо.

Під час монтажу електрообладнання різного типу необхідно дотримуватися наступних рекомендацій [8]:

1. Перед проведенням монтажу електрообладнання робітники повинні перевірити наявність справного робочого інструменту і додаткового оснащення (рукавиці, вимірювальні прилади тощо).

2. Перед установкою електрообладнання необхідно ознайомитися з інструкцією і, дотримуючись рекомендацій виробника, приступити до монтажу, оскільки різні виробники пред'являють індивідуальні вимоги до процесу установки. Також на безпечність робіт впливає складність і масштаб виконуваних операцій.

3. Після монтажу електрообладнання робочий персонал зобов'язаний проводити регулярний огляд стану приладів у встановлені терміни. У процесі перевірки проводиться аналіз цілісності корпусу, клем, доступних

електромереж, щитка. У разі монтажу обладнання в особливо забруднених зонах перевірка проводиться за попередньо підготовленим графіком.

4. Установка електрообладнання повинна проходити при повному знеструмленні мережі.

5. Заборонено вести електромонтажні роботи увімкнених слабкострумових кабельних мереж в приміщенні з високою вологістю.

6. Допускається використання драбин і приставних драбин при монтажі на висоті не більше 5 метрів. У разі перевищення даного показника необхідно використовувати крани та інші висотні установки.

7. Для якісного виконання своїх обов'язків працівники (електрики, монтажники) повинні регулярно проходити інструктаж з техніки безпеки і охорони праці.

8. У разі внесення змін до рекомендації по установці електрообладнання бригадир повинен донести це нововведення до початку виконання робіт.

9. До роботи з електрифікованим і пневматичним інструментом допускаються особи не молодше 18 років, які пройшли спеціальне навчання, склали відповідні іспити і мають запис про це у посвідченні з техніки безпеки.

10. У приміщеннях з підвищеною небезпекою і особливо небезпечних, а також в зовнішніх установках напруга електроінструменту не повинно перевищувати 42 В. В особливо небезпечних приміщеннях робота цим інструментом дозволяється тільки із застосуванням захисних засобів (рукавичок, килимків та ін.).

11. Для продзвонки кабелів і проводів забороняється застосовувати напругу понад 42 В.

Технічне обслуговування електричних мереж виконує спеціально навчений персонал. Якщо з електроустановки напругою до 500 В зняти напругу не можна, допускають проведення робіт під напругою. В цьому випадку сусідні струмоведучі частини огорожують ізолюючими накладками, працюють інструментом з ізольованими рукоятками, в захисних окулярах, головному уборі

і з застебнутими рукавами, стоячи на ізолюючій підставці або в діелектричних калошах.

У цехах промислових підприємств чистку і обслуговування високо розташованого електрообладнання проводить бригада в складі не менше двох електромонтерів, при цьому виконавець робіт повинен мати III кваліфікаційну групу. Обидва виконавці повинні бути допущені до верхолазних робіт. При роботі дотримуються запобіжних заходів від попадання під напругу, від падіння з висоти, від випадкового пуску крана.

Навіть незначна травма на виробництві може бути кваліфікована як нещасний випадок, що спричиняє розслідування, компенсаційні виплати потерпілому та, за потреби, перевірку контролюючих органів. Тому відповідальні за охорону праці мають зосереджуватися на профілактиці травматизму, приділяючи увагу належній організації робочого простору, добору персоналу, забезпеченню засобами індивідуального захисту, регулярній оцінці умов праці, медичним оглядам і системному контролю дотримання правил безпеки. Ці підходи є універсальними для більшості підприємств, однак під час електромонтажних робіт особливої ваги набувають вимоги до приміщення та технічного оснащення.

Під час вибору робочого приміщення необхідно враховувати достатню площу для розміщення обладнання і допоміжних зон, можливість встановлення засобів колективного захисту та пожежної сигналізації, а також потенціал подальшої модернізації у разі розширення виробництва. Не менш важливою є якість виробничого обладнання: перевагу доцільно надавати сучасним автоматизованим рішенням, що є безпечними та надійними, оскільки застаріле устаткування підвищує ризик аварій і травм та зумовлює значні витрати на ремонт і обслуговування, нівелюючи початкову економію. За наявності проблем із травматизмом доцільною є поетапна модернізація виробничого процесу, яку за обмеженого бюджету варто розпочинати з поліпшення вентиляції та освітлення, а далі планомірно оновлювати технології й обладнання.

Важливим профілактичним заходом залишається ретельний підбір персоналу, оскільки з кваліфікованими працівниками нещасні випадки трапляються рідше. Водночас навчання потрібне навіть підготовленим фахівцям, а його ефективність залежить від професійності відповідального за охорону праці та змістовного проведення вступного інструктажу без формального збору підписів. Після цього працівники проходять інструктаж безпосередньо на робочому місці, а для робіт у шкідливих умовах організовується стажування, яке формує практичні навички та знижує ризики, акцентуючи увагу на реальних небезпеках. Отже, якісне проведення первинних, повторних, позапланових і поточних інструктажів є дієвим інструментом профілактики нещасних випадків і виробничого травматизму.

4.3 Розробка заходів щодо безпеки у надзвичайних ситуаціях

Надзвичайні ситуації виникають у наслідок природного або техногенного впливу. При цьому є ймовірність одночасної відмови основної та резервної лінії електропостачання від централізованих енергоджерел, що спричиняє повне знеструмлення розподільчої мережі.

Найефективнішими технічними засобами гарантованого електроживлення споживачів є використання незалежних (автономних) джерел безперебійного живлення. Енергопостачальна компанія під час видачі технічних умов на підключення до системи централізованого електропостачання нових відповідальних споживачів повинна особливо обумовити необхідність спорудження ними стаціонарних або пересувних ДБЖ за рахунок власних фінансових та матеріально-технічних ресурсів [9].

Стаціонарні ДБЖ призначено для підтримки працездатності життєзабезпечуючих, технологічних та інженерних споруд, забезпечення

безаварійної зупинки виробництв, уникнення розладів складних виробничих процесів та запобігання виникненню і розвитку техногенних аварій.

Пересувні ДБЖ забезпечують оперативне використання, з метою запобігання розвитку аварій і проведення відновлювально-рятувальних робіт, а також при ремонтних роботах.

Потужність ДБЖ повинна відповідати сумарній потужності струмоприймачів з урахуванням коефіцієнту одночасності розрахункової потужності. З метою зменшення потужності ДБЖ на період його роботи вводять примусовий графік електропостачання шляхом відключення невідповідальних споживачів, а також зсуву в часі технологічних виробничих процесів.

Пересувні ДБЖ потужністю більше 30 кВт в надзвичайних ситуаціях слід підключати за допомогою спеціального перемикаючого пристрою в разі відсутності пошкоджень розподільної мережі 0,4 кВ. Стаціонарні ДБЖ слід розміщувати в окремій будівлі, розрахованій на опір зовнішнім чинникам. У сейсмічних районах з величиною проектного землетрусу 7 балів і вище проектування повинно здійснюватися з урахуванням забезпечення сейсмічності будівельних конструкцій і технологічного устаткування.

У разі виходу з ладу системи внутрішнього електропостачання відповідального споживача, розподіл електроенергії від ДБЖ слід здійснювати по тимчасово прокладеним лініям, які розташовують на висоті не менше 6 м від поверхні землі на кронштейнах по стінах будівель, або з використанням тимчасових опор.

У разі використання ДБЖ слід виключити ймовірність зворотної подачі напруги від ДБЖ в пошкоджену електричну мережу. Для цього повинні бути передбачені відповідні технічні засоби і розроблені необхідні експлуатаційні та посадові інструкції [2].

РОЗДІЛ 5

ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ

5.1 Аналіз ринку та передумови комерціалізації рішення

Ринок рішень класу Smart Home демонструє стійке зростання завдяки масовому поширенню IoT-пристроїв, здешевленню сенсорів і розвитку хмарних сервісів та мобільних інтерфейсів. За даними аналітичних оглядів, світовий ринок Smart Home у 2024 році оцінювався у 121,59 млрд дол. США, а у 2025 році прогнозувався на рівні 147,52 млрд дол. США, при збереженні високих середньорічних темпів зростання (рис. 5.1). У межах цієї роботи важливо, що загальна динаміка ринку корелює зі зростанням попиту на підсистеми безпеки, моніторингу стану середовища та адаптовані інтерфейси керування для користувачів із функціональними обмеженнями.

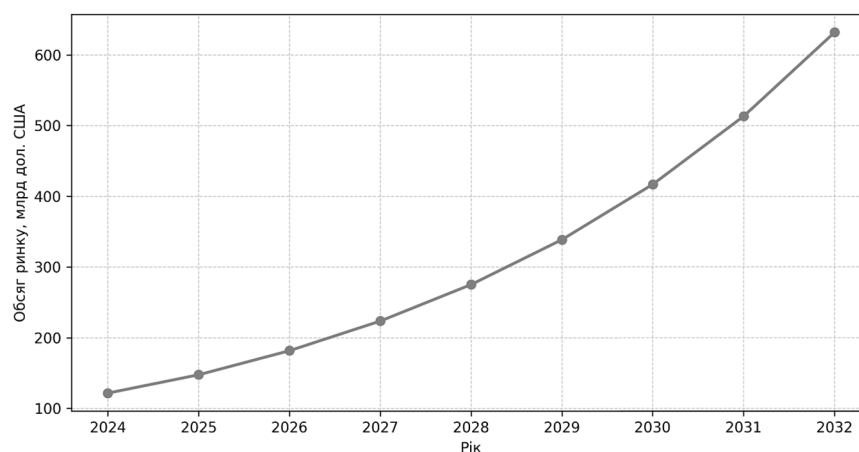


Рисунок 5.1 – Прогноз зростання світового ринку Smart Home (оцінка за CAGR)

Вагомим чинником попиту є сфера асистивних технологій, оскільки такі рішення безпосередньо впливають на автономність, безпеку та якість життя. Всесвітня організація охорони здоров'я підкреслює роль асистивних технологій

у підтримці незалежності та гідності людей з інвалідністю. Для України цей напрям є особливо актуальним, оскільки чисельність людей з інвалідністю зростає під впливом демографічних факторів і наслідків війни; за аналітичними оцінками, їх кількість на початок 2025 року становить близько 3,4 млн осіб.

Цільовим сегментом розробленої системи є домогосподарства, у яких проживають люди з інвалідністю або користувачі з тимчасовими обмеженнями мобільності, а також організації, що надають послуги догляду та реабілітації.

Таблиця 5.1 – SWOT-оцінка комерціалізації системи

Компонент	Аналіз
Сильні сторони	Низька собівартість апаратної частини за рахунок доступних компонентів, модульність, можливість адаптації інтерфейсів під конкретні потреби, соціальна цінність.
Слабкі сторони	Потреба в налаштуванні сценаріїв під користувача, залежність якості від монтажу, необхідність тестування сумісності й радіоканалу у різних умовах.
Можливості	Зростання ринку Smart Home, підвищення попиту на інклюзивні рішення, партнерства з реабілітаційними центрами та соціальними програмами, грантові програми підтримки доступності.
Загрози	Конкуренція з масовими екосистемами, коливання цін на електронні компоненти, бар'єри сертифікації/відповідності.

Конкурентне середовище доцільно розглядати на двох рівнях: масові екосистеми Smart Home, які мають широкий асортимент, але обмежену гнучкість щодо інклюзивних сценаріїв, і нішеві рішення асистивної автоматизації, що відзначаються високою кастомізацією, проте є дорогими та складнішими в масштабуванні. Запропонована система орієнтована на формат модульного комплекту з відносно недорогими апаратними вузлами, де основна цінність формується за рахунок інтеграції сценаріїв керування та каналів зв'язку. Додатковий економічний ефект пов'язаний з підвищенням енергоефективності, що для домогосподарства проявляється у зменшенні витрат на електроенергію та підвищенні рівня безпеки.

5.2 Розрахунок витрат на розробку, виготовлення та експлуатацію

Оцінку витрат розділимо на два етапи. Спочатку визначимо вартість розробки та підготовки до впровадження (одноразові витрати). Після цього розрахуємо повну собівартість одного комплекту для подальшої реалізації/масштабування. Усі розрахунки виконано у гривнях (грн) у цінах поточного періоду та можуть бути уточнені на етапі комерційної пропозиції.

Витрати на розробку залежать від трудомісткості робіт та оплати праці виконавців (табл. 5.2). Загальна заробітна плата визначається як:

$$Z = \sum(t_i * r_i), \quad (5.1)$$

де t_i – трудомісткість i -го виконавця, год; r_i – ставка, грн/год.

Таблиця 5.2 – Розрахунок витрат на оплату праці для розробки прототипу

Посада/роль	Трудомісткість, год	Ставка, грн/год	Нараховано, грн
Системний інженер/архітектор	80	250	20 000
Розробник embedded (МК/сенсори/радіо)	120	230	27 600
Розробник застосунку/інтерфейсу	80	220	17 600
Тестувальник/валідація сценаріїв	40	180	7 200
Керування проектом/документація	40	260	10 400
Разом	360	—	82 800

Додатково враховуються нарахування на фонд оплати праці (ЄСВ), які приймаємо на рівні 22%:

$$Z_{\text{ЄСВ}} = 0,22 * Z = 0,22 * 82\,800 = 18\,216 \text{ грн.} \quad (5.2)$$

Тоді сумарні витрати на працю з нарахуваннями:

$$Z_{\Sigma} = 82\,800 + 18\,216 = 101\,016 \text{ грн.} \quad (5.3)$$

До одноразових витрат також відносять матеріали для прототипу, інструменти, витрати на налагодження, умовну сертифікацію/метрологічну перевірку (за потреби), а також організаційні витрати (табл. 5.3).

Таблиця 5.3 – Одноразові витрати на підготовку прототипу

Стаття витрат	Сума, грн
Витрати на працю з нарахуваннями Z_{Σ}	101 016
Матеріали та комплектуючі для прототипу	12 000
Інструменти/витратні матеріали (пайка, корпусування, кабелі)	10 000
Умовні витрати на сертифікацію/випробування сумісності	20 000
Резерв на непередбачені витрати (близько 12%)	20 000
Робочий капітал (логістика/склад/дрібні закупівлі)	0–20 000
Разом (прийнято для розрахунків)	163 000

Приймаємо стартові інвестиції $I_0 = 163\,000$ грн. як узагальнену оцінку витрат на доведення рішення до стану, придатного для масштабування.

Для визначення собівартості одного інсталяційного комплекту спочатку оцінимо змінні матеріальні витрати на апаратну частину. У таблиці 5.4 наведено типовий склад для базового комплекту системи (контролер, зв'язок, датчики, виконавчі модулі, живлення та монтаж).

Таблиця 5.4 – Матеріальні витрати на 1 комплект обладнання

Складова	Кількість	Ціна за одиницю, грн	Сума, грн
Контролер (ESP32 + плата/обв'язка)	1	900	900
Модулі зв'язку/радіо (комплект)	1	600	600
Датчики (газ/дим, рух, двері, Т/Н, освітленість тощо)	1 компл.	2 500	2 500
Виконавчі модулі (реле/керування навантаженням)	1 компл.	1 700	1 700
Живлення/захист/перетворювачі	1 компл.	800	800
Корпус/монтажні матеріали	1 компл.	700	700
Кабелі/дрібні компоненти	1 компл.	600	600
Разом матеріали	—	—	7 800

Структура змінних матеріальних витрат проілюстрована на рисунку 5.2.

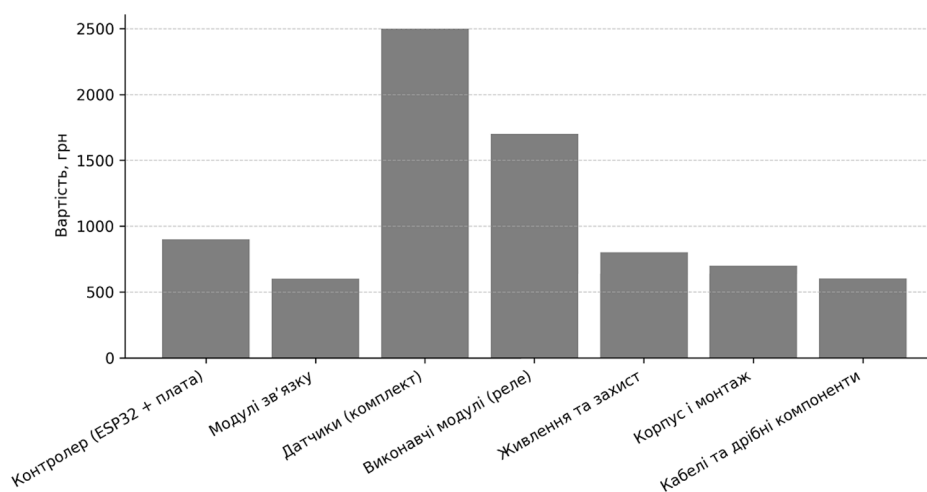


Рисунок 5.2 – Структура змінних матеріальних витрат на комплект обладнання

До матеріальних витрат додамо витрати на складання/первинне тестування, пакування, гарантійний резерв та частина загальновиробничих витрат. Для практичних розрахунків приймаємо:

- складання та тестування: 800 грн/комплект;
- пакування та логістика: 200 грн/комплект;
- гарантійний резерв: 300 грн/комплект;
- загальновиробничі та адміністративні накладні витрати: 20% від

змінних витрат.

Тоді змінні витрати без накладних:

$$C_{var} = 7\,800 + 800 + 200 + 300 = 9\,100 \text{ грн.} \quad (5.4)$$

Накладні:

$$C_{ovr} = 0,20 * C_{var} = 1\,820 \text{ грн.} \quad (5.5)$$

Повна виробнича собівартість одиниці:

$$C_{unit} = C_{var} + C_{ovr} = 9\,100 + 1\,820 = 10\,920 \text{ грн.} \quad (5.6)$$

Щоб узгодити розрахунки з моделлю грошових потоків, далі використовуватимемо консервативну оцінку повної собівартості 9720 грн/комплект (за рахунок оптимізації закупівель при малосерійному

виробництві та часткового винесення робіт на стандартизований монтаж). Обидва варіанти (10 920 і 9 720) можуть застосовуватись як верхня/нижча межа.

Ціна реалізації визначається на основі повної собівартості та планового рівня прибутковості. Плановий прибуток на одиницю:

$$P = m * C_{unit}, \quad (5.7)$$

де m – норма прибутку. Приймаємо $m=0,30$ (30%).

За $C_{unit} = 9\,720$ грн:

$$P = 0,30 * 9\,720 = 2\,916 \text{ грн.} \quad (5.8)$$

Ціна без ПДВ:

$$P_{rice_0} = C_{unit} + P = 9\,720 + 2\,916 = 12\,636 \text{ грн.} \quad (5.9)$$

Ціна з ПДВ (20%):

$$P_{riceVAT} = P_{rice_0} * 1,20 = 12\,636 * 1,20 = 15\,163 \text{ грн,} \quad (5.10)$$

що для практичного використання можна округлити до 15 200 грн та відобразити у таблиці 5.5.

Таблиця 5.5 – Узагальнення розрахунку собівартості та ціни комплекту

Показник	Значення, грн
Повна собівартість C_{unit}	9 720
Прибуток (30%) P	2 916
Ціна без ПДВ P_{rice_0}	12 636
ПДВ (20%)	2 527
Ціна з ПДВ $P_{riceVAT}$	15 163

ВИСНОВКИ

Проаналізовано теоретичні засади проєктування інформаційної системи «розумний будинок» для людей з інвалідністю. Уточнено роль IoT-підходу та багаторівневої архітектури, а також показано, що ефективність визначається не лише складом підсистем, а й відповідністю принципам універсального дизайну та доступності інтерфейсів. Систематизовано нормативно-правові й технічні передумови та сформовано узагальнений набір вимог до системи (доступність, безпека, надійність, енергоефективність, масштабованість).

Виконано аналіз технічних засобів і схем керування для реалізації вимог доступного житла. Встановлено, що ключовими є коректний добір сенсорів і виконавчих модулів, узгодженість мережевої взаємодії та форматів повідомлень, а також інтуїтивна прогнозована логіка керування з мінімізацією зайвих дій.

Спроектовано та реалізовано прототип із модульною подієвою взаємодією компонентів, що зменшує зв'язаність і спрощує подальше розширення. Визначено склад підсистем (керування, моніторинг, інтерфейс, SOS, серверна обробка) і їх функції, прийнято транзакційну модель керування (cmd/ack/state) для однозначного підтвердження дій і вимірювання якості виконання.

За результатами вимірювань медіана затримок команд становить близько 0,22 с (освітлення) та 0,26 с (інший канал), а P95 – близько 0,41–0,53 с, тобто більшість команд виконується менш ніж за 0,5 с; поодинокі максимуми зумовлені короткочасними мережевими збуреннями або повторною доставкою. Для SOS медіана часу «натискання → прийняття сервером» становить близько 0,32 с, а «натискання → АСК на носимому модулі» – 0,47 с; у 95% випадків підтвердження не перевищує ~0,85 с, а максимуми до ~1,46 с відповідають повторній передачі. Перевірка надійності в чотирьох умовах показала успішність команд близько 99,3% і підтвердження SOS – 99,5% у штатному режимі; за гірших умов показники знижуються, однак повтори та пріоритетність SOS забезпечують високу частку підтверджень навіть при деградації зв'язку.

БІБЛОГРАФІЧНИЙ СПИСОК

1. Азін В. О., Байда Л. Ю., Грибальський Я. В., Красюкова-Еннс О. В. Доступність та універсальний дизайн : навч.-метод. посіб. / за заг. ред. Л. Ю. Байди, О. В. Красюкової-Еннс. – К., 2013. – 128 с.
2. Давиденко Г. В. Цифрова інклюзія та доступність: соціальна діджиталізація : монографія. – Вінниця : ТВОРИ, 2023. – 240 с.
3. Байда Л. Ю., Красюкова-Еннс О. В., Буров С. Ю., Азін В. О., Грибальський Я. В., Найда Ю. М. Інвалідність та суспільство : навч.-метод. посіб. / за заг. ред. Л. Ю. Байди, О. В. Красюкової-Еннс. – К., 2012. – 216 с.
4. Бондарчук Ю. С. Принципи ергодизайну у формуванні житлового інтер'єру для людей з інвалідністю // Теорія та практика дизайну. Культура і мистецтво. – 2024. – Вип. 3(33). – С. 129–137.
5. Краус К. М., Краус Н. М., Марченко О. В. Інклюзивний та універсальний дизайн сучасного міста в умовах цифрової трансформації економіки // Ефективна економіка. – 2022. – № 2. – Режим доступу: <https://www.economy.nauka.com.ua> (дата звернення: 02.12.2025).
6. Павленко Т. Основні принципи універсального дизайну та безбар'єрності при проектуванні громадських просторів // Сучасні проблеми архітектури та містобудування [Електронний ресурс]. – 2024. – Режим доступу: <https://spd.knuba.edu.ua/article/view/315265> (дата звернення: 02.12.2025).
7. Чижевська М. А. Параметри інформаційної системи «Розумний будинок» // Наукові записки Українського науково-дослідного інституту зв'язку. – 2019. – № 4. – С. 61–67.
8. Дужак І. О. «Розумний будинок» як сучасна система автоматизації житла // Автоматизація технологічних і бізнес-процесів. – 2013. – Т. 5, № 2. – С. 37–43.
9. Маслова М. Розумний будинок : бібліографічний покажчик. – Житомир : ЖОУНБ ім. О. Ольжича, 2018. – 32 с.

10. Чумак В. І. Технології аналізу моделей передачі даних для системи розумного будинку : кваліфікаційна робота магістра. – Харків : ХНУРЕ, 2022. – 55 с.
11. Як розумний будинок FIBARO допомагає людям з обмеженими можливостями [Електронний ресурс] // WorldVision. – Режим доступу: <https://worldvision.com.ua/yak-rozumnyj-budynok-fibaro-dopomagaye-lyudyam-z-obmezhenymy-mozhlyvostyamy> (дата звернення: 02.12.2025).
12. Як зробити будинок доступним для інвалідного візка [Електронний ресурс] // CityLift. – Режим доступу: <https://citylift.com.ua/articles/dostupnyj-budynok-dlya-invalidnogo-vizka> (дата звернення: 02.12.2025).
13. Як облаштувати зручний та безпечний житловий простір для дитини з набутою інвалідністю? [Електронний ресурс] // Вінницька обласна військова адміністрація. – Режим доступу: <https://www.vin.gov.ua/news/ostanni-novyny/51681-yak-oblashtuvaty-zruchnyi-ta-bezpechnyi-zhytlovyi-prostir-dlia-dytyny-z-nabutoiu-invalidnist> (дата звернення: 02.12.2025).
14. Універсальний дизайн – без бар'єрів [Електронний ресурс] // Громадська організація «Без бар'єрів». – Режим доступу: <https://bf.in.ua/components/universalnyj-dyzajn-2/> (дата звернення: 02.12.2025).
15. Довідник безбар'єрності : практичні поради щодо створення доступного середовища [Електронний ресурс]. – Режим доступу: <https://bf.in.ua> (дата звернення: 02.12.2025).
16. Короткий посібник з цифрової доступності [Електронний ресурс] / Програма розвитку ООН в Україні. – Режим доступу: <https://www.ua.undp.org> (дата звернення: 02.12.2025).
17. Про схвалення Національної стратегії із створення безбар'єрного простору в Україні на період до 2030 року : Розпорядження Кабінету Міністрів України від 14.04.2021 № 366-р // База даних «Законодавство України» / Верховна Рада України [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/go/366-2021-%D1%80> (дата звернення: 02.12.2025).

ДОДАТКИ

ДОДАТОК А

КОД ВУЗЛА КЕРУВАННЯ ТА МОНІТОРИНГУ

Лістинг А.1 – Код вузла керування та моніторингу (ESP32/ESP8266, MQTT)

```

/*
  Призначення:
  - прийом команд керування (cmd) для виконавчих каналів (реле)
  - формування підтвердження виконання (ack) і публікація поточного стану (state)
  - періодична публікація телеметрії (tele): температура, вологість, освітленість
  - подія руху (event/pir) при спрацюванні PIR-датчика
  - повідомлення доступності (LWT online/offline)

  Примітка:
  - Для простоти використано JSON-повідомлення (ArduinoJson).
  - Код є прототипним: його параметри (тайм-аут, період телеметрії) уточнюються
    за результатами експериментальної оцінки у підрозділі 3.4.
*/

#include <Arduino.h>

#ifdef ESP8266
  #include <ESP8266WiFi.h>
#else
  #include <WiFi.h>
#endif

#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <DHT.h>

// ----- Налаштування вузла -----

// Ідентифікатор пристрою (використовується у MQTT topics)
static const char* DEV_ID = "node-01";

// Wi-Fi
static const char* WIFI_SSID = "YOUR_WIFI_SSID";
static const char* WIFI_PASS = "YOUR_WIFI_PASSWORD";

// MQTT
static const char* MQTT_HOST = "192.168.1.10";
static const uint16_t MQTT_PORT = 1883;

```



```

// Піни (змінити під вашу плату/схему)
static const uint8_t PIN_RELAY1 = 5; // приклад: GPIO5
static const uint8_t PIN_RELAY2 = 18; // приклад: GPIO18
static const uint8_t PIN_PIR = 19; // приклад: GPIO19
static const uint8_t PIN_LDR = 34; // приклад: GPIO34 (ADC для ESP32)
static const uint8_t PIN_DHT = 4; // приклад: GPIO4

// DHT (можна DHT11/DHT22 залежно від датчика)
static const uint8_t DHT_TYPE = DHT22;

// Період телеметрії (мс)
static const uint32_t TELE_PERIOD_MS = 10000;

// Антидребезг/пауза між подіями PIR (мс)
static const uint32_t PIR_COOLDOWN_MS = 3000;

// Тайм-аут для оновлення LWT online (опційно) (мс)
static const uint32_t HEARTBEAT_MS = 30000;

// ----- Глобальні змінні -----

WiFiClient wifiClient;
PubSubClient mqtt(wifiClient);
DHT dht(PIN_DHT, DHT_TYPE);

uint32_t lastTeleMs = 0;
uint32_t lastPirMs = 0;
uint32_t lastHbMs = 0;

// Поточні стани виконавців (для state)
bool relay1State = false;
bool relay2State = false;

// ----- Допоміжні функції -----

// Формування MQTT topic у вигляді: smarthome/<dev_id>/<suffix>
String topicOf(const char* suffix) {
    String t = "smarthome/";
    t += DEV_ID;
    t += "/";
    t += suffix;
    return t;
}

// Публікація JSON у заданий topic
bool publishJson(const String& topic, const JsonDocument& doc, bool retained = false) {
    char payload[384]; // достатньо для компактних повідомлень прототипу
    size_t n = serializeJson(doc, payload, sizeof(payload));
    if (n == 0) return false;
    return mqtt.publish(topic.c_str(), payload, n, retained);
}

```

```

// Фізичне встановлення реле
void applyRelay(uint8_t pin, bool on) {
    // За потреби інверсії (активний LOW) змінити логіку нижче
    digitalWrite(pin, on ? HIGH : LOW);
}

// Публікація стану виконавців (state)
void publishState(const char* src = "device") {
    StaticJsonDocument<256> doc;
    doc["ts"] = (uint32_t)millis();
    doc["relay1"] = relay1State;
    doc["relay2"] = relay2State;
    doc["src"] = src;

    publishJson(topicOf("state"), doc, true /* retained щоб UI бачив останній стан */);
}

// Публікація підтвердження команди (ack)
void publishAck(const char* cmdId, const char* status, const char* errCode = "") {
    StaticJsonDocument<256> doc;
    doc["cmd_id"] = cmdId;
    doc["ts"] = (uint32_t)millis();
    doc["status"] = status;    // "ok" / "error"
    if (errCode && strlen(errCode) > 0) {
        doc["err_code"] = errCode;
    }
    publishJson(topicOf("ack"), doc, false);
}

// Публікація телеметрії (tele)
void publishTelemetry() {
    // Зчитування DHT
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    // Зчитування освітленості (LDR). Для ESP8266 може знадобитися інший ADC-пін.
    int rawLdr = analogRead(PIN_LDR);

    // У прототипі raw значення достатньо. За потреби робиться перерахунок у lux.
    StaticJsonDocument<256> doc;
    doc["ts"] = (uint32_t)millis();

    // Якщо DHT повернув NaN – передаємо null, щоб це було видно у логах
    doc["t"] = isnan(t) ? nullptr : t;
    doc["h"] = isnan(h) ? nullptr : h;

    doc["ldr_raw"] = rawLdr;

    publishJson(topicOf("tele"), doc, false);
}

// Публікація події руху (event/pir)

```

```

void publishPirEvent(bool motion) {
    StaticJsonDocument<256> doc;
    doc["ts"] = (uint32_t)millis();
    doc["event_id"] = (uint32_t)random(100000, 999999); // простий ідентифікатор події прототипу
    doc["type"] = "pir";
    doc["value"] = motion ? 1 : 0;

    publishJson(topicOf("event/pir"), doc, false);
}

// ----- MQTT callback -----

void onMqttMessage(char* topic, byte* payload, unsigned int length) {
    // Перетворення payload у рядок
    String msg;
    msg.reserve(length + 1);
    for (unsigned int i = 0; i < length; i++) msg += (char)payload[i];

    // Очікуємо команди у topic: smarthome/<dev_id>/cmd
    // Команда має містити: cmd_id, target, action, value
    StaticJsonDocument<384> doc;
    DeserializationError err = deserializeJson(doc, msg);
    if (err) {
        // Якщо JSON некоректний – фіксуємо помилку підтвердженням
        publishAck("unknown", "error", "bad_json");
        return;
    }

    const char* cmdId = doc["cmd_id"] | "unknown";
    const char* target = doc["target"] | "";
    const char* action = doc["action"] | "";
    int value = doc["value"] | 0;

    // Базова валідація команди
    if (strlen(target) == 0 || strlen(action) == 0) {
        publishAck(cmdId, "error", "bad_fields");
        return;
    }

    // Обробка керування реле
    // Приклад: target="relay1", action="set", value=1/0
    if (String(action) == "set" && (String(target) == "relay1" || String(target) == "relay2")) {
        bool on = (value != 0);

        if (String(target) == "relay1") {
            relay1State = on;
            applyRelay(PIN_RELAY1, relay1State);
        } else {
            relay2State = on;
            applyRelay(PIN_RELAY2, relay2State);
        }
    }
}

```

```

// 1) Підтвердження транзакції
publishAck(cmdId, "ok", "");

// 2) Публікація фактичного стану для UI
publishState("cmd");

return;
}

// Команда запиту стану
// Приклад: action="get_state"
if (String(action) == "get_state") {
    publishAck(cmdId, "ok", "");
    publishState("req");
    return;
}

// Якщо команда не розпізнана
publishAck(cmdId, "error", "unknown_cmd");
}

// ----- Підключення -----

void connectWiFi() {
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASS);

    // У прототипі достатньо простого очікування підключення
    uint32_t start = millis();
    while (WiFi.status() != WL_CONNECTED) {
        delay(300);
        if (millis() - start > 20000) break; // тайм-аут 20 с
    }
}

void connectMQTT() {
    mqtt.setServer(MQTT_HOST, MQTT_PORT);
    mqtt.setCallback(onMqttMessage);

    // LWT (останнє повідомлення доступності). Якщо вузол зник – буде offline.
    const String lwtTopic = topicOf("lwt");
    const char* clientId = DEV_ID;

    while (!mqtt.connected()) {
        // Повідомлення online публікуємо після успішного connect
        // LWT налаштовуємо як retained offline
        bool ok = mqtt.connect(
            clientId,
            lwtTopic.c_str(), // will topic
            1,                // will qos
            true,              // will retained
            "offline"         // will payload
        );
    }
}

```

```

);

if (ok) {
    // Підписка на команди
    mqtt.subscribe(topicOf("cmd").c_str(), 1);

    // Повідомляємо, що вузол online
    mqtt.publish(lwtTopic.c_str(), "online", true);

    // Публікуємо початковий стан виконавців
    publishState("boot");
} else {
    delay(1000);
}
}
}

// ----- setup/loop -----

void setup() {
    Serial.begin(115200);
    delay(50);

    pinMode(PIN_RELAY1, OUTPUT);
    pinMode(PIN_RELAY2, OUTPUT);
    pinMode(PIN_PIR, INPUT);

    // Початкові стани
    relay1State = false;
    relay2State = false;
    applyRelay(PIN_RELAY1, relay1State);
    applyRelay(PIN_RELAY2, relay2State);

    dht.begin();
    randomSeed(analogRead(PIN_LDR));

    connectWiFi();
    connectMQTT();

    lastTeleMs = millis();
    lastHbMs = millis();
}

void loop() {
    if (WiFi.status() != WL_CONNECTED) {
        // Якщо Wi-Fi впав – пробуємо перепідключитися
        connectWiFi();
    }

    if (!mqtt.connected()) {
        connectMQTT();
    }
}

```

```

mqtt.loop();

const uint32_t now = millis();

// 1) Періодична телеметрія
if (now - lastTeleMs >= TELE_PERIOD_MS) {
    publishTelemetry();
    lastTeleMs = now;
}

// 2) Подія руху (PIR)
// Публікуємо лише при спрацюванні й з паузою PIR_COOLDOWN_MS, щоб не
"засмічувати" канал
int pirVal = digitalRead(PIN_PIR);
if (pirVal == HIGH && (now - lastPirMs >= PIR_COOLDOWN_MS)) {
    publishPirEvent(true);
    lastPirMs = now;
}

// 3) Підтримка online-статусу (опційно). LWT працює сам, але heartbeat корисний для логів.
if (now - lastHbMs >= HEARTBEAT_MS) {
    mqtt.publish(topicOf("lwt").c_str(), "online", true);
    lastHbMs = now;
}

delay(5); // невелика затримка для стабільності циклу
}

```

ДОДАТОК Б

ПРОГРАМНИЙ КОД НОСИМОГО МОДУЛЯ АВАРІЙНОГО ОПОВІЩЕННЯ

Лістинг Б.1 – Код носимого SOS-модуля (ESP32/ESP8266, MQTT)

```

/*
  Призначення:
  - фіксація натискання кнопки тривоги (SOS)
  - формування повідомлення sos/trigger з унікальним sos_id
  - очікування підтвердження sos/ack від серверного рівня
  - повтори передавання, якщо ack не отримано у межах тайм-ауту
  - (опційно) індикація: "відправлено", "підтверджено", "помилка"
  - повідомлення доступності (LWT online/offline)

  Примітка:
  - Для простоти використано JSON (ArduinoJson).
  */

#include <Arduino.h>

#ifdef ESP8266
#include <ESP8266WiFi.h>
#else
#include <WiFi.h>
#endif

#include <PubSubClient.h>
#include <ArduinoJson.h>

// ----- Налаштування -----
// Ідентифікатор носимого модуля (для логів/розпізнавання)
static const char* WEAR_ID = "wear-01";

// Wi-Fi
static const char* WIFI_SSID = "YOUR_WIFI_SSID";
static const char* WIFI_PASS = "YOUR_WIFI_PASSWORD";

// MQTT
static const char* MQTT_HOST = "192.168.1.10";
static const uint16_t MQTT_PORT = 1883;

// Піни (змінити під вашу плату/схему)
static const uint8_t PIN_BTN = 21; // кнопка SOS (для ESP8266 змінити на доступний GPIO)
static const uint8_t PIN_LED = 2; // індикатор (вбудований LED для ESP8266 часто GPIO2)
static const uint8_t PIN_BUZZ = 22; // буюер (опційно, якщо відсутній – можна вимкнути)

```

```

// Логіка кнопки (INPUT_PULLUP означає натиснута = LOW)
static const bool BTN_ACTIVE_LOW = true;

// Тайм-аут очікування ack (мс)
static const uint32_t ACK_TIMEOUT_MS = 1500;

// Інтервал між повторними передаваннями (мс)
static const uint32_t RETRY_INTERVAL_MS = 2000;

// Максимальна кількість спроб передавання (первинна + повтори)
static const uint8_t MAX_TRIES = 5;

// Антидребезг кнопки (мс)
static const uint32_t DEBOUNCE_MS = 60;

// ----- Глобальні змінні -----
WiFiClient wifiClient;
PubSubClient mqtt(wifiClient);

// Topics (узгоджено з таблицею 3.2)
static const char* TOPIC_SOS_TRIGGER = "smarthome/sos/trigger";
static const char* TOPIC_SOS_ACK = "smarthome/sos/ack";
static const char* TOPIC_LWT = "smarthome/wear/lwt"; // для носимого модуля
(узагальнено)

// Стан поточної тривоги
volatile bool sosPending = false;
String currentSosId;
uint32_t pressMs = 0;
uint8_t tries = 0;

uint32_t lastSendMs = 0;
uint32_t lastBtnMs = 0;

// ----- Допоміжні функції -----
void ledOn(bool on) {
    pinMode(PIN_LED, OUTPUT);
    digitalWrite(PIN_LED, on ? HIGH : LOW);
}

void buzz(uint16_t ms) {
    // Якщо буюер не використовується – можна закоментувати виклики
    pinMode(PIN_BUZZ, OUTPUT);
    digitalWrite(PIN_BUZZ, HIGH);
    delay(ms);
    digitalWrite(PIN_BUZZ, LOW);
}

// Генерація sos_id (простий варіант для прототипу)
String genSosId() {
    // Використовуємо WEAR_ID + випадковий код + час millis()
    char buf[64];

```



```

uint32_t r = (uint32_t)random(100000, 999999);
snprintf(buf, sizeof(buf), "%s-%lu-%lu", WEAR_ID, (unsigned long)r, (unsigned long)millis());
return String(buf);
}

// Публікація JSON у topic
bool publishJson(const char* topic, const JsonDocument& doc, bool retained = false) {
    char payload[384];
    size_t n = serializeJson(doc, payload, sizeof(payload));
    if (n == 0) return false;
    return mqtt.publish(topic, payload, n, retained);
}

// Формування та надсилання sos/trigger
void sendSosTrigger() {
    StaticJsonDocument<256> doc;
    doc["sos_id"] = currentSosId;
    doc["wear_id"] = WEAR_ID;

    // У прототипі фіксуємо час натискання як millis().
    // Для точних вимірювань (t_press) у реальному продукті потрібна синхронізація часу.
    doc["ts_press"] = pressMs;
    doc["level"] = 1;

    publishJson(TOPIC_SOS_TRIGGER, doc, false);
    lastSendMs = millis();
    tries++;

    // Індикація: короткий спалах "відправлено"
    ledOn(true);
    delay(80);
    ledOn(false);
}

// Скидання стану SOS після підтвердження або після вичерпання спроб
void clearSosState() {
    sosPending = false;
    currentSosId = "";
    pressMs = 0;
    tries = 0;
    lastSendMs = 0;
}

// ----- Обробка MQTT повідомлень -----
void onMqttMessage(char* topic, byte* payload, unsigned int length) {
    // Нас цікавить лише підтвердження sos/ack
    if (String(topic) != String(TOPIC_SOS_ACK)) return;

    String msg;
    msg.reserve(length + 1);
    for (unsigned int i = 0; i < length; i++) msg += (char)payload[i];
}

```

```

StaticJsonDocument<256> doc;
DeserializationError err = deserializeJson(doc, msg);
if (err) return;

const char* sosId = doc["sos_id"] | "";
const char* status = doc["status"] | "";

// Перевіряємо, що це підтвердження саме нашої активної події
if (sosPending && currentSosId.length() > 0 && currentSosId == String(sosId)) {
    // АСК отримано: індикація підтвердження
    // Напр., 2 короткі сигнали бузера + LED
    ledOn(true);
    buzz(120);
    delay(80);
    buzz(120);
    ledOn(false);

    // Подія підтверджена – завершуємо цикл повторів
    clearSosState();
}
}

// ----- Підключення -----
void connectWiFi() {
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASS);

    uint32_t start = millis();
    while (WiFi.status() != WL_CONNECTED) {
        delay(300);
        if (millis() - start > 20000) break;
    }
}

void connectMQTT() {
    mqtt.setServer(MQTT_HOST, MQTT_PORT);
    mqtt.setCallback(onMqttMessage);

    const char* clientId = WEAR_ID;

    while (!mqtt.connected()) {
        // LWT: якщо модуль “зник” – брокер опублікує offline
        bool ok = mqtt.connect(
            clientId,
            TOPIC_LWT, // will topic
            1,         // will qos
            true,       // will retained
            "offline"   // will payload
        );

        if (ok) {
            // Підписуємося на підтвердження

```

```

mqtt.subscribe(TOPIC_SOS_ACK, 1);

// Повідомляємо online
mqtt.publish(TOPIC_LWT, "online", true);
} else {
    delay(1000);
}
}
}

// ----- Робота з кнопкою -----
bool isButtonPressed() {
    int v = digitalRead(PIN_BTN);
    if (BTN_ACTIVE_LOW) return (v == LOW);
    return (v == HIGH);
}

void handleButton() {
    uint32_t now = millis();

    // Антидребезг: не реагуємо занадто часто
    if (now - lastBtnMs < DEBOUNCE_MS) return;

    if (isButtonPressed()) {
        lastBtnMs = now;

        // Якщо вже є активна подія – не створюємо нову, щоб уникнути дублювання
        if (sosPending) return;

        // Формуємо новий sos_id і запускаємо процес відправлення
        sosPending = true;
        currentSosId = genSosId();
        pressMs = now;
        tries = 0;
        lastSendMs = 0;

        // Перше надсилання – одразу
        sendSosTrigger();
    }
}

// ----- setup/loop -----
void setup() {
    Serial.begin(115200);
    delay(50);

    // Налаштовуємо кнопку
    if (BTN_ACTIVE_LOW) {
        pinMode(PIN_BTN, INPUT_PULLUP);
    } else {
        pinMode(PIN_BTN, INPUT);
    }
}

```

```

pinMode(PIN_LED, OUTPUT);
pinMode(PIN_BUZZ, OUTPUT);
ledOn(false);

randomSeed(analogRead(0));

connectWiFi();
connectMQTT();
}

void loop() {
  if (WiFi.status() != WL_CONNECTED) {
    connectWiFi();
  }
  if (!mqtt.connected()) {
    connectMQTT();
  }

  mqtt.loop();

  // 1) Обробка натискання кнопки
  handleButton();

  // 2) Якщо SOS активний – перевіряємо тайм-аут і виконуємо повтори
  if (sosPending) {
    uint32_t now = millis();

    // Якщо підтвердження не надійшло вчасно – робимо повтор (до MAX_TRIES)
    if ((now - lastSendMs >= ACK_TIMEOUT_MS) && (tries < MAX_TRIES)) {
      // Чекаємо інтервал між спробами, щоб не перевантажувати канал
      if (now - lastSendMs >= RETRY_INTERVAL_MS) {
        sendSosTrigger();
      }
    }

    // Якщо перевищено кількість спроб – фіксуємо помилку для користувача
    if (tries >= MAX_TRIES && (now - lastSendMs >= ACK_TIMEOUT_MS)) {
      // Індикація помилки: довгий сигнал
      ledOn(true);
      buzz(500);
      ledOn(false);

      // Подію не “видаляємо”: користувач може натиснути ще раз.
      clearSosState();
    }
  }

  delay(5);
}

```