

Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра автоматизації, комп'ютерних наук і технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти – бакалавр
за освітньо-професійною програмою
«Автоматизація та комп'ютерно-інтегровані технології»
зі спеціальності

151 – Автоматизація та комп'ютерно-інтегровані технології

Тема роботи:

**«Автоматизація систем контролю та керування розумним будинком з
розробкою програмного забезпечення»**

Виконала студентка гр. АКІТ-21ск _____ Рожкова Е. О.

Керівник _____ Тиханський М. П.

Нормоконтроль _____ Маринич І. А.

В.о.завідувача кафедри _____ Рубан С. А.

Кривий Ріг-2024

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: інформаційних технологій

Кафедра: автоматизації, комп'ютерних наук і технологій

Ступінь вищої освіти: Бакалавр

Спеціальність: 151 - Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ

Зав. кафедрою: к.т.н. Рубан С.А.

«25» березня 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

студентці групи АКІТ-21ск Рожковій Еліні Олегівні

1. Тема кваліфікаційної роботи: «Автоматизація систем контролю та керування розумним будинком з розробкою програмного забезпечення»

затверджено наказом по університету № 232с від 25.03.2024 р.

2. Термін здачі кваліфікаційної роботи: 03.06.2024 р.

3. Склад кваліфікаційної роботи: Пояснювальна записка, презентація у Microsoft PowerPoint в електронному та друкованому вигляді

4. Консультанти кваліфікаційної роботи:

Розділ 1-2

Тиханський М.П.

Нормоконтроль

Маринич І. А.

5. «Календарний план:

№	Етапи роботи	Термін виконання
1	<i>Вступ</i>	<i>05.04.24</i>
2	<i>Розділ 1</i>	<i>18.04.24</i>
3	<i>Розділ 2</i>	<i>02.05.24</i>
4	<i>Висновки</i>	<i>23.05.24</i>
5	<i>Оформлення кваліфікаційної роботи</i>	<i>01.06.24</i>
6	<i>Підготовка презентації та графічного матеріалу</i>	<i>10.05.24</i>
7	<i>Підготовка доповіді до захисту</i>	<i>07.06.24</i>

6. Дата видачі завдання: 25.03.2024р.

Керівник _____ / Тиханський М. П./

7. Запевнення: Я, Рожкова Еліна Олегівна, запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про академічну доброчесність Криворізького національного університету ознайомена.

Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі умисних порушень робота не допускається до захисту або оцінюється незадовільно.

Студентка _____ / Рожкова Е. О./

АНОТАЦІЯ

Рожкова Е. О. Автоматизація систем контролю та керування розумним будинком з розробкою програмного забезпечення

Кваліфікаційна робота на здобуття ступеня вищої освіти – бакалавр, за напрямом 151 – Автоматизація та комп’ютерно-інтегровані технології. – Криворізький національний університет, Кривий Ріг, 2024.

Робота складається з вступу, двох розділів, висновків, списку літератури з 37 позицій. Загальний обсяг роботи становить 70 сторінок, з яких основний зміст роботи викладено на 54 сторінках, робота включає 1 таблицю і 36 рисунків.

Об’єктом проектування є система моніторингу і керування та веб-інтерфейс для керування системою на базі платформи Raspberry.

Метою проекту є розробка системи моніторингу мікроклімату в приміщенні, системи детектування диму, керування освітленням та розробка веб-інтерфейсу для дистанційного керування системою «Розумний будинок» з будь-якого девайсу.

В першому розділі була розглянута концепція «Розумного будинку» та історія розвитку систем домашньої автоматизації, а також надана порівняльна характеристика існуючим технологіям та рішенням з автоматизації.

У другому розділі був здійснений вибір обладнання для проектування системи та описані їх характеристики, побудований макет, налаштований мікрокомп’ютер Raspberry, написана програма для мікроконтролера та розроблений веб-інтерфейс та потоки для його функціонування.

Ключові слова: АВТОМАТИЗАЦІЯ, РОЗУМНИЙ БУДИНОК, RASPBERRY PI, ESP8266, MQTT, MOSQUITTO, NODE-RED, МІКРОКЛІМАТ, СИГНАЛІЗАЦІЯ, УПРАВЛІННЯ ОСВІТЛЕННЯМ, ВЕБ-ІНТЕРФЕЙС

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Рожкова Е.О.</i>			<i>АНОТАЦІЯ</i>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Тиханський М.П.</i>						
<i>Н. Контр.</i>		<i>Маринич І.А.</i>				<i>КНУ АКІТ-21ск</i>		
<i>Затвердив</i>		<i>Тронь В.В.</i>						

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	9
1.1 Концепція «розумного» будинку.....	9
1.1.1 Історія розвитку «розумного будинку».....	9
1.1.2 Основні принципи розумного будинку.....	10
1.1.3 Ключові компоненти системи	12
1.1.4 Можливості автоматизації та управління	13
1.1.5 Сучасний погляд на системи розумного дому	15
1.2 Огляд існуючих технологій та способів реалізації.....	16
1.2.1 Дротові та бездротові технології для реалізації розумного будинку ...	16
1.2.2 Публікаційно-підписний протокол MQTT.....	19
1.2.3 Існуючі рішення з домашньої автоматизації та їх порівняння	20
Висновок до розділу 1	26
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОЇ ЧАСТИНИ.....	28
2.1 Особливості мікрокомп'ютера Raspberry Pi.....	28
2.2 Вибір та обґрунтування прийнятих компонентів системи	30
2.2.1 Raspberry Pi 3B+	30
2.2.2 ESP8266 CP2120.....	32
2.2.3 Датчики DHT11, MQ-2 та модуль активного зумера YL-44	33
2.2.4 Світлодіодна стрічка WS2812.....	37
2.3 Схема підключення компонентів	38
2.3.1 Функціональна схема системи.....	38
2.3.2 Схема підключення компонентів до мікроконтролера ESP8266 на макетній платі.....	41
2.4 Первинне налаштування Raspberry Pi 3B+	44
2.4.1 Установка ОС Raspbian та перший запуск	44
2.4.2 Встановлення статичного IP для плати Raspberry.....	46

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Рожкова Е.О.			<i>ЗМІСТ</i>	Лім.	Арк.	Аркушів
Перевір.		Тиханський М.П.						
Н. Контр.		Маринич І.А.				<i>КНУ АКІТ-21ск</i>		
Затвердив		Тронь В.В.						

2.4.3	Налаштування віддаленого доступу до плати	47
2.5	Налаштування та конфігурація MQTT брокера.....	48
2.5.1	Установка та налаштування Mosquitto	49
2.5.2	Налаштування Node-RED та Node-RED Dashboard	50
2.6	Розробка програмного забезпечення для ESP8266.....	51
2.7	Розробка потоків та інтерфейсів в Node-RED.....	55
2.7.1	Створення потоку «Моніторинг температури та вологості»	56
2.7.2	Створення потоку «Управління світлодіодною стрічкою»	60
2.7.3	Створення потоку «Сигналізація»	62
	Висновок до розділу 2	65
	ВИСНОВКИ	66
	СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	67
	ДОДАТОК А Код для NodeMCU ESP8266 CP2102	71

ВСТУП

Сучасний світ стрімко розвивається, приносячи у повсякденне життя нові технології та інноваційні рішення. Одним із найяскравіших проявів цього процесу є концепція «Розумного будинку», яка поєднує в собі автоматизацію, зручність та безпеку. Розумний будинок – це не просто житловий простір, а комплексна система, що включає безліч пристроїв і датчиків, які взаємодіють між собою, створюючи комфортні умови для проживання, економлячи енергію і забезпечуючи високий рівень безпеки.

Автоматизація систем контролю та управління розумним будинком відіграє ключову роль у цьому контексті. Вона дозволяє користувачам не лише віддалено керувати побутовими приладами та системами життєзабезпечення, а й отримувати оперативну інформацію про стан будинку, запобігати аварійним ситуаціям та оптимізувати споживання ресурсів. За допомогою сучасних технологій, таких як Інтернет речей (IoT), можливе створення інтегрованих систем, що адаптуються під індивідуальні потреби та уподобання кожного користувача.

Метою даного дипломного проекту є розробка програмного забезпечення для автоматизації систем контролю та управління розумним будинком та створення демонстративного стенду. У ході роботи будуть розглянуті сучасні підходи до побудови розумних будинків, проаналізовано існуючі рішення та технології, а також розроблено та протестовано власні алгоритми та програми для управління розумним будинком. Основну увагу буде приділено питанням комфорту та безпеки.

У процесі дослідження буде створено систему, здатну інтегрувати різні пристрої та датчики, забезпечувати їх взаємодію та управління в реальному часі. Особлива увага буде приділена розробці інтуїтивно зрозумілого

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Рожкова Е.О.</i>			<i>ВСТУП</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Архивів</i>
<i>Перевір.</i>		<i>Тиханський М.П.</i>						
<i>Н. Контр.</i>		<i>Маринич І.А.</i>			<i>КНУ АКИТ-21ск</i>			
<i>Затвердив</i>		<i>Тронь В.В.</i>						

інтерфейсу користувача, який дозволить легко та швидко налаштовувати та контролювати роботу всіх компонентів розумного будинку.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проведення аналізу існуючих рішень з домашньої автоматизації;
- вибір, знаходження та придбання необхідного обладнання;
- навчитися користуватися платою, працювати з її бібліотеками;
- спроектувати схему автоматизації
- створити макет системи та провести серію первинних випробувань з подальшим налагодженням роботи системи.
- обрати відповідні інструменти та розробити програмне забезпечення для контролю та керування розумним будинком;

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		8

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Концепція «розумного» будинку

У сучасних умовах стрімкого технологічного прогресу концепція «розумного будинку» набуває все більшої популярності. Ця область поєднує в собі автоматизацію різних аспектів повсякденного життя, тому розробка ефективних систем контролю та управління розумним будинком потребує глибокого розуміння предметної галузі та ретельного аналізу існуючих рішень.

Розумний дім, домашня автоматизація, домашня кібернетична система – це ідея об'єднання різних систем домашньої автоматизації в одну для забезпечення більшої безпеки, комфорту, економії енергії та можливості віддаленого моніторингу. Фундаментальна ідея розумного будинку полягає у використанні сучасних інформаційних технологій для централізованого керування та автоматизації різноманітних домашніх пристроїв, побутової техніки та інженерних систем будівлі. Основною метою розумного будинку є підвищення якості життя мешканців. [1, 2, 3]

Є багато визначень для «розумного» будинку, але всі виділяють, що суть такої системи полягає в підключенні приладів та пристроїв до мережі зв'язку для моніторингу, доступу та контролю показників житлового середовища. [2]

1.1.1 Історія розвитку «розумного будинку»

«Розумний будинок» - це ідея, яка має історію, починаючи з 60-х років, коли в житлові будинки почали проникати нові технології в автоматизації та

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Рожкова Е.О.</i>			<i>РОЗДІЛ 1</i>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Тиханський М.П.</i>						
<i>Н. Контр.</i>		<i>Маринич І.А.</i>			<i>КНУ АКІТ-21ск</i>			
<i>Затвердив</i>		<i>Тронь В.В.</i>						

електроніці. Основні періоди розвитку:

1960-1970-ті роки

EDIUS (Experimental Home Integrated Control System) була однією з перших систем, яка заклала основи для домашньої автоматизації. Вона була розроблена у Сполучених Штатах у 1966 році і мала центральну панель, яка контролювала освітлення, опалення, прилади та охорону.

У 1975 році була представлена система X10, яка дозволяла керувати освітленням і приладами за допомогою контрольних сигналів через наявну проводку будинку. Ця молода технологія бездротового керування широко використовувалась. [15]

1980-ті роки: Хоча офіційною датою народження «розумного будинку» прийнято вважати 1978 рік, сам термін був впроваджений трохи пізніше. Національна асоціація будівельників цивільних споруд США ввела цей термін у 1984 році, щоб описувати будинок, у якому є інтегровані системи управління, які підвищують енергоефективність, безпеку та функціональність.

1990-2000 роки

У 1990-х роках розвиток комп'ютерних мереж, Інтернету та бездротових технологій створив сильний поштовх для індустрії домашньої автоматизації. розроблено протоколи та стандарти для підключення LonWorks, SEBus та інших пристроїв та систем.

В 2000-х роках з'явилися перші комерційні системи «розумного будинку» від Control4, Creatron, AMX і Savant. Рішення на основі комп'ютерних технологій Microsoft, IBM, Cisco також набули широкого поширення. [15, 20]

1.1.2 Основні принципи розумного будинку

Першим основним принципом побудови системи розумного будинку є централізований контроль і моніторинг всіх підсистем через єдину точку доступу. Усі системи розумного будинку мусять бути підключені та

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

інтегровані до центрального контролера, який служить єдиною точкою керування в будинку. Це дозволяє використовувати один інтерфейс, щоб керувати та контролювати роботу всіх підсистем (освітлення, клімат, охорона тощо).

Другий принцип це автоматизація регулярних домашніх процесів та попередньо визначених сценаріїв. Завдяки цьому користувач може налаштувати автоматичне виконання певних дій, наприклад включення світла за визначеним часом. Крім того, можна створювати сценарії з кількома компонентами «в один клік». Наприклад, можна створити сценарій «Запуск», який вмикає світло, активує обладнання та активує систему безпеки.

Ще один важливий принцип це можливість дистанційного керування, хоча він присутній не у всіх системах. Так мешканці можуть переглядати інформацію про стан будинку та керувати системами, перебуваючи поза домом або у відрядженні, за допомогою мобільного додатку або веб-інтерфейсу.

Підвищення ефективності використання енергії та ресурсозбереження шляхом оптимізації роботи обладнання також є важливим аспектом при проектуванні систем домашньої автоматизації. Інтелектуальні алгоритми використовують дані датчиків і історію використання, щоб оптимізувати використання енергії систем опалення, охолодження та освітлення, одночасно знижуючи витрати на марнотрату. [4]

І останній принцип, це забезпечення високого рівня безпеки та максимальний комфорт для мешканців. Охоронні сигналізації, відеоспостереження та контроль доступу підвищують безпеку дому. Комфорт досягається за допомогою автоматизації сценаріїв освітлення, клімату та мультимедіа.

Отже, концептуальна основа розумного будинку базується на інтеграції різносторонніх домашніх систем в єдину об'єднану мережу з інтелектуальною системою управління.

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

1.1.3 Ключові компоненти системи

Найважливішою частиною в системі смарт-будинку є сама мережа, яка здійснює обмін інформацією в реальному часі між системою та користувачем для моніторингу та керуванням смарт-пристроями. Для безперебійного зв'язку між контролером і всіма компонентами системи, їх узгодження та встановлення в систему, потрібна надійна мережева інфраструктура, яка часто базується на Wi-Fi або інших спеціальних протоколах зв'язку. [2, 4]

Як правило, система мусить мати центральний вузол, який відповідає за мережу та контроль за всіма ресурсами. Шлюз, також відомий як центральний контролер, є «мозком» системи, який керує взаємодією всіх компонентів. Він об'єднує всі підсистеми розумного будинку в єдину мережу. Датчики передають дані контролеру, який потім їх аналізує та надсилає команди виконавчим механізмам відповідно до запрограмованих сценаріїв. [1, 2]

Зазвичай, розумний будинок як система автоматизації, включає в себе управління освітленням, опаленням, кондиціонуванням, вентиляцією, а також системами доступу та безпеки. Контроль та управління цими показниками здійснюється завдяки датчикам руху, диму, витоку, температури, світла та ін. «Очі та вуха» розумного дому можуть складатися з різних типів датчиків, які збиратимуть інформацію про навколишнє середовище, присутність людей і потенційні загрози, на основі яких система приймає рішення. [1]

Виконавчі пристрої – це приводи, реле, актуатори для управління побутовим обладнанням. Це «руки» розумного будинку, які можуть виконувати керування різними функціями, наприклад побутовою технікою, системою клімат-контролю та іншими функціями за командою контролера. [4]

Системи безпеки можуть включати в себе датчики руху, диму, води, вогню, дверних замків, розумні розетки, сигналізації та системи відеоспостереження. Це одна з найважливіших підсистем, яка контролює та захищає будинок від проникнення, пожеж, витоків та інших загроз. [2,4]

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

З часом питання енергоефективності в системах смарт-будівель також стало досить поширеним. Системи розумного енергоменеджменту використовуються для оптимізації споживання енергії. Ці алгоритмічні системи дозволяють знизити споживання енергії, автоматично регулюючи її використання в залежності від режиму життя жильців та їх потреб, аналізуючи споживання ресурсів і керуючи роботою комунального обладнання. [2, 4]

В сучасних системах впроваджуються інтелектуальні системи управління, такі як мобільні програми, веб-інтерфейси, сенсорні панелі та голосові помічники. Користувацькі інтерфейси дозволяють людям взаємодіяти з системою розумного дому, щоб дізнаватися про стан, налаштовувати параметри та виконувати сценарії. Це дозволяє більш зрозуміло, швидко та віддалено, керувати будинком.[15]

Звісно ж, ключовими компонентами системи є побутова техніка та обладнання, які відповідають концепції розумного дому. Пристрої, які сумісні та підтримують звичайні протоколи зв'язку, необхідні, щоб максимально інтегруватися в систему розумного будинку. [2]

Все ж таки, можна виділити, що найбільш критично важливими компонентами для таких систем являються центральний контролер як ядро системи, а також мережева інфраструктура та інтерфейси управління, які забезпечують зв'язність та можливість взаємодії зі всіма елементами розумного будинку.

1.1.4 Можливості автоматизації та управління

Система розумного дому представляє можливість централізованого інтелектуального управління та автоматизацію широкого спектру домашніх систем та пристроїв. [1]

Керування освітленням: регулювання яскравості та кольору, створення світлових сценаріїв, вмикання/вимикання світла за розкладом або при детектуванні руху в кімнаті. Автоматизоване керування освітленням дозволяє

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

адаптувати освітлення приміщення до часу доби, активності та уподобань мешканців. Це створює приємні умови та економить електроенергію. [4]

Кондиціонування, вентиляція та опалення забезпечують комфортний мікроклімат у приміщенні. Інтелектуальна система клімат-контролю автоматично регулює температуру та вологість, щоб забезпечити ідеальні умови. Це не тільки економить енергію, але й створює комфорт. [2, 4]

Мультимедійні системи та пристрої включають управління аудіо- та відеотехнікою, а також домашніми кінотеатрами. Якщо мультимедійні системи включені в розумний дім, це дає можливість керувати ними з одного місця, створювати інструкції для перегляду фільмів або прослуховування музики. [15]

«Розумна» побутова техніка може включати в себе пральні та сушильні машини, духовки, кавомашини та роботи-пилососи. Інтеграція побутової техніки в систему «розумний» дім дозволяє зручно керувати та програмувати її роботу віддалено.

Сонцезахисні функції включають автоматичне відкривання та закривання штор, жалюзі та рулонних штор. Автоматичне керування системами сонячного затінення покращує комфорт, регулюючи природне освітлення, затінення та приватність.

Комплексні системи безпеки включають сигнали охорони та пожежі, відеоспостереження та контроль доступу. Смарт-будинки мають інтегровані системи безпеки, щоб захистити майно від злому, пожеж, витоків та інших загроз, що гарантує збереження майна.

Смарт-системи обліку та управління енергією включають електроенергію, воду та газ. Якщо в системі є модулі для обліку та оптимізації споживання різних ресурсів, це дозволяє керувати споживанням ресурсів і економити гроші. [2]

Автоматичне зрошення саду та ландшафту полегшує догляд за садом, використовується для своєчасного та ефективного зволоження зелених посадок.

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Управління витокami води та інженерними системами будівлі. Функції моніторингу та попередження про витoki та несправності інженерних систем підвищують безпеку та дозволяють швидко реагувати на потенційні аварії.

Таким чином, розумний будинок являє собою масштабну концепцію інтеграції самих різноманітних домашніх систем та пристроїв в єдину мережу для спрощеного управління, суттєво збільшує рівень безпеки, комфорту та енергоефективності для жильців. [2]

1.1.5 Сучасний погляд на системи розумного дому

У сучасному світі концепція «розумного дому» міцно закріпилася як один з провідних напрямків впровадження інформаційних технологій у повсякденне життя людей. Розширення можливостей обробки великих даних, хмарних обчислень, IoT і AI відкривають нові перспективи для домашньої автоматизації. [4]

Упровадження екосистем IoT все більше інтегруються з системами смарт-дому. Підключені до Інтернету датчики, пристрої та побутова техніка дозволяють збирати дані, контролювати та керувати віддалено за допомогою хмарних сервісів. Це дозволяє створювати сценарії автоматизації, які є більш інтелектуальними, використовуючи алгоритми машинного навчання. [4, 15]

Розумні будинки отримують нові можливості завдяки впровадженню AI-технологій. Алгоритми машинного навчання можуть бути використані системами для вивчення моделей поведінки та вподобань мешканців і адаптації сценаріїв зайнятості до потреб кожного. На основі прогнозування аналітики AI також допомагає оптимізувати споживання енергії. [15]

Використовуючи віртуальних помічників, таких як Amazon Alexa, Google Assistant, та Apple Siri голосові інтерфейси стають все більш популярними. Розвиток технологій розпізнавання та синтезу мови робить взаємодію з домашньою системою більш природною та простою.

У рамках концепції «розумного» міста більш тісний зв'язок між розумними будинками та міською інфраструктурою є тенденцією сучасного

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

часу. Це дозволить синхронізувати роботу будинкових систем з комунальними службами, що призведе до більш ефективного використання ресурсів, доступних у місті.

Сучасною тенденцією є інтеграція розумних будинків із системами віддаленого моніторингу стану здоров'я. Дані з різних носимих пристроїв і побутових датчиків можна аналізувати, щоб відстежувати життєво важливі показники та надавати правильну медичну допомогу.

Отже, сучасний погляд на системи розумного будинку наголошує на тому, наскільки вони тісно пов'язані з технологіями IoT, AI, хмарними сервісами, голосовими інтерфейсами та міською інфраструктурою. Це створює нові можливості для підвищення продуктивності, зручності та безпеки сучасних будівель. [15]

1.2 Огляд існуючих технологій та способів реалізації

Концепція розумного будинку базується на використанні різних технологій для створення інтелектуального життєвого середовища, яке робить життя зручним, безпечним і економним. В даному підрозділі розглядаються існуючі технологічні рішення, які лежать в основі систем смарт-будівель, включаючи як поширені, так і менш відомі, але перспективні методи.

1.2.1 Дротові та бездротові технології для реалізації розумного будинку

Дротові технології:

Дротова система X10 є однією з традиційних технологій домашньої автоматизації, яка використовує наявну електропроводку, щоб передавати керуючі сигнали на різні пристрої та побутову техніку. Незважаючи на свій вік, X10 все ще є поширеним та доступним і бюджетним варіантом. [18]

Більш сучасна дротова технологія, KNX, яка раніше мала назву EIB, є відкритим стандартом для інтеграції всіх систем розумного будинку через

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

спеціалізовану шину. Рішення, які базуються на KNX, гарантують високу продуктивність і надійність. [20]

Також варто згадати LonWorks, запатентовану Echelon дротову технологію, яка використовує нейронні мережі для автоматизації будівель. LonWorks добре сприймається як у комерційних, так і промислових умовах завдяки потужній екосистемі сертифікованих пристроїв.

Хоча це не спеціалізована технологія розумного будинку, відкрити протокол BACnet часто використовується для інтеграції систем домашньої автоматизації з інженерною інфраструктурою будівлі, що дозволяє комбінувати обладнання від різних виробників.

Бездротові технології:

В останні роки в індустрії смарт-будівель бездротові протоколи стали більш поширеними, що полегшило впровадження систем і підключення пристроїв.

Найпоширенішою технологією є Z-wave, бездротова mesh-мережа, розроблена для низького споживання енергії та простої інтеграції між різними пристроями від різних розробників. [1, 5]

ZigBee є додатковим широко використовуваним стандартом бездротових мереж із самоорганізацією та маршрутизацією на основі сітчастої топології. Він дуже масштабований і є частиною багатьох комерційних рішень для розумного будинку, датчиків і освітлення. [1, 5]

Протоколи Bluetooth Low Energy і Bluetooth Mesh дозволяють створювати інтелектуальні мережі з мінімальним споживанням енергії за допомогою бездротового зв'язку пристроїв з низьким споживанням енергії на короткій відстані. Розширення мережі Bluetooth дозволяє масштабувати такі мережі. [5]

Не можна не згадати Wi-Fi, популярну бездротову технологію для організації локальних мереж, яка широко використовується в Інтернеті речей і рішеннях для розумного будинку завдяки високим швидкостям передачі даних. [1, 5]

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Стандарт Thread є останньою розробкою, розробленою спеціально для розумних будинків і міст. Це IP-сумісна сітчаста мережа з низьким енергоспоживанням і розширеними заходами безпеки. [5]

Незважаючи на всі переваги, існуючі дротові та бездротові технології розумного дому мають деякі недоліки.

X10, яка використовує електропроводку, є вже досить застарілою. До того ж вона чутлива до перешкод від інших електроприладів і має обмеження щодо відстані та кількості пристроїв. [18]

Сучасні дротові стандарти, такі як LonWorks і KNX/EIB, відрізняються високими витратами на розгортання дротової інфраструктури та обладнання, а також складністю монтажу та налаштування. KNX вимагає спеціальних підготовки, а LonWorks – це власна технологія одного постачальника, що робить інтеграцію з іншими системами складною. [18]

Хоча протокол BACnet дозволяє сумістити обладнання різних виробників, він був розроблений для масштабної автоматизації будівель, тому він може бути незручним для невеликих домашніх систем.

Переходячи до бездротових рішень, Z-Wave і ZigBee є чутливі до перешкод та використовують неліцензовані діапазони. Ці технології також мають обмежений діапазон і пропускну здатність каналу. Існують також проблеми з ZigBee, пов'язані з сумісністю пристроїв та масштабуванням великих сітчастих мереж. [1, 5]

Протоколи Bluetooth з низьким енергоспоживанням мають обмежений діапазон окремих вузлів, можуть мати проблеми з сумісністю та мають низьку пропускну здатність даних. [5]

Хоча бездротові мережі пропонують високі швидкості передачі даних, вони потребують більше енергії, ніж спеціальні протоколи Інтернету речей. Таким чином, для вибору технологічної платформи для системи розумного будинку, необхідно ретельно розглянути всі вимоги щодо продуктивності, масштабованості, енергоефективності, сумісності з існуючим обладнанням і екосистеми підтримуваних пристроїв і компонентів. [1]

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2.2 Публікаційно-підписний протокол MQTT

Завдяки своїм функціям та перевагам MQTT є одним із найкращих протоколів для систем розумного дому.

Перш за все, MQTT ідеально підходить для розподілених систем Інтернету речей і розумного дому, оскільки він є простим протоколом обміну повідомленнями для підписки та публікації. Його мінімалістичний дизайн дозволяє ефективно використовувати ресурси мережі та гарантує надійне передавання даних навіть при нестабільному з'єднанні. [21]

MQTT надає гнучку та масштабовану модель зв'язку між різними датчиками, шлюзами та пристроями розумного будинку завдяки архітектурі публікації/підписки. Пристрої можуть публікувати дані на певні теми, що дозволяє користувачам отримувати лише те, що їх цікавить. Це полегшує інтеграцію різноманітних компонентів, одночасно зменшуючи мережевий трафік. [21]

Механізми підтвердження доставки та повторної передачі входять до трьох рівнів якості обслуговування повідомлень, які підтримує протокол MQTT. Це необхідно для забезпечення надійності та цілісності даних систем житлової автоматизації.

Одною із переваг MQTT є його низьке споживання енергії та пропускна здатність, що робить його придатним для використання в пристроях з обмеженими ресурсами, таких як датчики з живленням від батареї.

Механізми безпеки, такі як автентифікація та шифрування трафіку, які підтримуються MQTT, також є корисними для захисту конфіденційних даних та запобігання несанкціонованому доступу.

Крім того, MQTT є відкритим стандартом, який має велику спільноту та різноманітні бібліотеки для різних платформ, що робить інтеграцію простішою.

Таким чином, MQTT є чудовим рішенням для організації обміну даними та взаємодії різних частинами систем розумного будинку, оскільки він є простим, економічним, надійним і масштабованим протоколом.

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2.3 Існуючі рішення з домашньої автоматизації та їх порівняння

При обиранні структури управління розумним будинком були розглянуті інші варіанти систем, кожна з котрих має свої переваги та недоліки.

Насамперед, на ринку існує багато компаній, які пропонують системи смарт-будинку, такі як: Google Nest, Amazon Alexa, Apple HomeKit, Samsung SmartThings, Hubitat Elevation. Але, нажаль, більшість із пропонованих систем на ринку автоматизують невелику низку завдань. Наприклад, система Ajax буде важко віднести до поняття «розумного будинку», бо то є просто дуже розвинена система безпеки. Тому, будуть розглянуті системи з самим широким функціоналом.

Безумовно, однією із найпопулярніших систем розумного будинку є Home Assistant. Це відкрите ПЗ, яке використовується для створення власної системи розумного будинку. Вона надає дуже велику гнучкість в налаштуванні, що дозволяє додавати різні пристрої автоматизації згідно з індивідуальними потребами. Важко знайти таку ж систему, котра підтримувала б таку велику кількість пристроїв для інтеграції та такий широкий список протоколів, що підтримуються. Також великою перевагою є те, що цей інструмент домашньої автоматизації має активну спільноту та постійно розвиває функціонал. [17]

Якщо казати за менш відомі розробки, то самим потужним представником буде Fibaro Home Center 3. Ця система надає широкий спектр можливостей автоматизації, що включає контроль освітлення, безпеки, опалення та багато іншого. Вона відома своєю простотою в установці та використанні. Також має високу надійність та функціональність. Має інтуїтивно зрозумілий інтерфейс користувача та має високу надійність та підвищену безпеку даних завдяки передовій технології шифрування. [16]

Loxone Smart Home System: це інтелектуальна система для збору і зберігання інформації, керування підключеними пристроями та контролю їх роботи, і все в єдиній платформі. До міні-серверу легко підключається до 30

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

додаткових розширень одночасно, що легко підвищує продуктивність та кількість поставлених задач системи. Все ще дозволяє здійснювати повний контроль та забезпечує оптимальний комфорт. [19]

Останнім прикладом готової системи автоматизації розумного будинку буде Control4 Smart Home System. Це система преміум-класу, що надає розширені можливості автоматизації та інтеграції з різними пристроями. Вона відома своєю потужністю та розширеними можливостями налаштування. Також відзначається своєю високою інтеграцією конкретно з такими розумними технологіями, як аудіо- та відеосистеми. [20]

Таблиця 1.1 - Порівняння названих існуючих систем

Назва	Простота встановлення	Простота налаштування	Бездротова мережа	Масштабованість	Наявність сценаріїв	Інтерфейс керування	Взаємодія з датчиками інших виробників	Голосове керування	Ціна
Home Assistant	-	-	+	+	+	веб-інтерфейс, мобільний додаток, спільнота	+	+	безкоштовно
Fibaro	+	+	+	+	+	мобільний додаток, веб-інтерфейс	+	+	від 300\$
Loxone	-	-	-	+	+	мобільний додаток, веб-інтерфейс, Loxone Miniserver	-	+	від 1500\$
Control4	-	-	+	+	+	додаток, пульт дис. управління, екранні панелі, веб-інтерфейс	-	+	від 1000\$

В рамках аналізу існуючих систем для автоматизації управління та контролю розумного будинку важливо розглянути патентні розробки в цій галузі. Патенти надають детальну інформацію про новітні технології та методи, що використовуються для покращення функціональності та

інтеграції розумних пристроїв. Було знайдено та запатентовано такі технології:

1. Патент U201212028 [6] описує метод і систему розумного будинку. Використання бездротових технологій дозволяє взаємодіяти з різними розумними пристроями в будинку, такими як термостати, системи безпеки та освітлення. Як сервер управління використаний принаймні один шлюз, що містить безпроводний приймально-передавальний елемент і забезпечує вихід у мережу Ethernet, а мережею Internet зв'язаний з хмаровим сервером, при цьому система виконана з можливістю реєстрації пристроїв і конфігурування алгоритмів їх взаємодії за допомогою WEB-сервісу у хмаровому сервері з подальшою передачею конфігурації у шлюз.



Рисунок 1.1 - Схематично зображена система управління інфраструктурою будинку "Розумний дім"

2. Патент US20240121133 [7] описує систему та метод керування пристроями розумного будинку за допомогою кількох інтерфейсів. Система

дозволяє користувачам взаємодіяти з різними розумними пристроями через єдиний інтерфейс, забезпечуючи інтеграцію та координацію їх роботи.

В патенті [7] надано системи та методи для Інтернету речей (IoT), системи клімат-контролю розумного будинку та системи зв'язку. IoT, система клімат-контролю та зв'язку в розумному домі включає в себе перший розумний домашній пристрій, який отримує джерела сигналу від глобальної мережі, передає сигнали, дані та команди на один або кілька розумних домашніх пристроїв у домі чи будівлі в локальній мережі IoT. Перший розумний домашній пристрій також отримує сигнали, дані та команди від одного або кількох розумних домашніх пристроїв у домі чи будівлі в локальній мережі IoT і передає сигнали, дані та/або команди в глобальну мережу. IoT LAN відрізняється від житлової бездротової локальної мережі.

3. Патент US201662359005P [8], зареєстрований під EP23178996, описує розширену систему керування для пристроїв розумного будинку. Метод, реалізований за допомогою комп'ютера, включає отримання аудіоданих, що представляють голосовий ввід; визначення першого користувача як постачальника голосового вводу, при цьому ідентифікація ґрунтується на порівнянні характеристик отриманих аудіоданих із даними профілю динаміка безлічі динаміків, включаючи першого користувача, для якого зберігаються дані профілю динаміка; визначення інструкції з отримання аудіоданих, при цьому інструкція посилається на другого користувача, крім першого користувача, і містить показник відносно мовця, а також визначення інструкції включає визначення посилення відносного мовця на основі другого збереженого профілю мовця, що містить дані, пов'язані із інформацією налаштування, пов'язаною з другим користувачем; та викликання дії, зазначеної інструкцією, яку потрібно виконати.

4. Патент EP20210150969 [9] пропонує спосіб керування декількома пристроями системи розумного будинку. Спосіб включає: перемикання системи розумного будинку між режимом «вдома» та режимом «поза домом»; і незалежно від того, що система розумного будинку знаходиться у

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

режимі відсутності, контролює налаштування кожного з безлічі пристроїв на основі шаблону використання відповідного пристрою; при цьому незалежне керування налаштуванням кожного з пристроїв системи включає: для конкретного пристрою визначення першого моменту часу для зміни налаштування пристрою на основі зразка використання пристрою, при цьому перший момент часу обмежений: будь-яким з набору передбачених моментів часу протягом дня, та моментом часу, коли система розумного будинку встановлена у режим «поза домом»; розрахунок другого моменту часу шляхом додавання випадкового зсуву в часі до першого моменту часу; змін налаштування пристрою у другому моменті часу; визначення періоду стабільності, протягом якого налаштування пристрою має бути збережено на основі зразка використання пристрою; та збереження налаштування пристрою протягом періоду стабільності. Також надається система розумного будинку.

Також, розглядаючи існуючі рішення з автоматизації, не можна не згадати вже спроектовану та встановлену систему розумного будинку, котра знаходиться в університеті. Ця система слугує демонстраційним стендом, який на практиці показує принципи та технології з домашньої автоматизації.



Рисунок 1.2 - Загальний вигляд розумного будинку

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		24

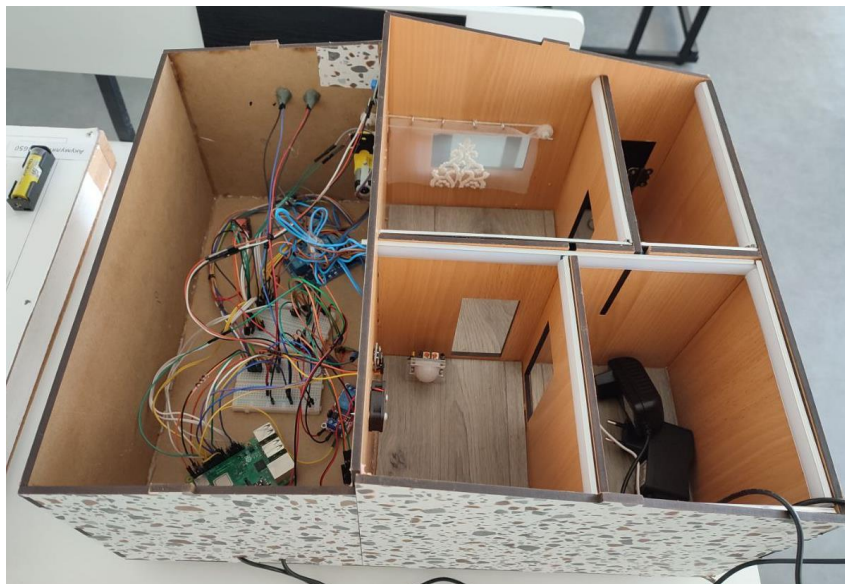


Рисунок 1.3 - Макет розумного будинку вид зсередини

Розроблена система включає різні елементи, такі як датчики, виконавчі механізми та альтернативні джерела живлення. Центральним контролером є плата Raspberry 3B+, яка відіграє ключову роль в управлінні та підключенні різних компонентів розумного будинку. Raspberry Pi взаємодіє зі всіма пристроями через свої контакти GPIO.

Функціонал системи включає управління освітленням, системами безпеки, клімат-контролем, жалюзі та резервним електроживленням. Ці функції реалізуються за допомогою таких пристроїв, як датчики руху, серводвигуни, реле, RFID-модулі та вентилятори. Програмування системи велось з використанням Python IDE.

Проект також включав створення функціональних та структурних схем автоматизації з використанням Eplan Education. Ці схеми докладно описують розташування та взаємозв'язки компонентів системи.

Мобільна програма для управління системою була розроблена з використанням платформи MIT App Inventor, що сприяло створенню інтерфейсу віддаленого керування «розумного дому». Також використовується технологія Bluetooth для програмування взаємодії елементів.

Недоліками цього проекту є складність у налаштуванні та

програмуванні Raspberry Pi як центра управління, а також масштабованість системи.

Для зменшення навантаження на центральний контролер системи, було б краще використати додатковий контролер для зв'язку його з елементами системи, а Raspberry використовувати у якості сервера. Це також сприятиме більш ефективному розподілу завдань, а ще підвищить загальну надійність системи.

Ще треба відзначити, що технологія Bluetooth є досить застарілою. Вона має менш надійну передачу даних, обмежена за дальністю, кількістю пристроїв та має гірше керування мережею, ніж той самий протокол MQTT. На відміну від Bluetooth, MQTT дозволяє будувати масштабовані системи з високою надійністю та стабільністю зв'язку.

Також, варто зазначити, що перед мобільним додатком, веб-інтерфейс має низку переваг, а саме: доступність з будь-якого пристрою з інтернет-браузеру, відсутність необхідності встановлення програм та більш легка масштабованість. Веб-інтерфейси суттєво легше оновлювати та розширювати функціонал системи, що робить його більш сучасним та гнучким рішенням у порівнянні з традиційними мобільними додатками.

Отже, виходячи з перерахованих систем, можна виділити такі важливі аспекти при проектуванні власної системи розумного будинку: функціональність, сумісність з пристроями та стандартами, простота встановлення та налаштування, безпека, вартість та масштабованість.

Висновок до розділу 1

У першому розділі детально розглядається концепція розумного будинку, включаючи її розвиток, основні принципи, основні частини, можливості автоматизації та сучасні тенденції. Згідно з дослідженнями, розумні системи домашньої автоматизації еволюціонували від звичайних систем до складних, що забезпечують комфорт, безпеку та

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

енергоефективність.

Розгляд основних елементів системи та основних принципів роботи дозволив визначити, які компоненти необхідні для створення ефективної системи розумного дому, підкресливши важливість сумісності, масштабованості та безпеки.

Пошук і аналіз існуючих технологій, таких як дротові та бездротові рішення, підтвердили, що використання протоколу MQTT, яку було обрано серед інших сучасних технологій і можливостей є одною з найактуальніших. Порівняння існуючих варіантів домашньої автоматизації допоможе в подальшому створити та запустити власну систему, використовуючи найкращі практики та сучасні технології.

Таким чином, аналіз предметної області та пошук існуючих рішень, створили основу для вибору найкращих методів і технологій для впровадження системи «розумний дім». Це також створює добру основу для наступних етапів розробки системи та програмного забезпечення.

					<i>КНУ КРБ.151.24.02.01.ПЗ</i>	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОЇ ЧАСТИНИ

2.1 Особливості мікрокомп'ютера Raspberry Pi

Після проведення аналізу предметної галузі та існуючих рішень настає ключовий етап розробки системи розумного будинку – проектування та реалізація власної апаратно-програмної частини. Цей розділ присвячений детальному опису процесу створення інтегрованої системи, яка включає як апаратні компоненти, так і програмне забезпечення для їх управління.

На етапі проектування важливо визначити архітектуру системи, обрати відповідні пристрої та технології, а також розробити схеми їхньої взаємодії. Увага буде приділятися вибору датчиків, контролерів та виконавчих пристроїв, які забезпечують надійну та ефективну роботу системи. Також розглядається питання безпеки та масштабованості системи.

Реалізація програмної частини включає розробку алгоритмів управління, інтеграцію та створення інтерфейсу користувача. Особливо розглядається забезпечення стабільної роботи системи, автозапуску всіх необхідних процесів при ввімкненні системи та обробці даних у реальному часі та можливості віддаленого керування. Програмне забезпечення повинно бути інтуїтивно зрозумілим, гнучким і легко налаштованим під потреби користувача.

Raspberry Pi це одноплатні комп'ютери, які розробляються фондом Raspberry Pi Foundation із Британії в асоціації із Broadcom. Першочергово, метою цього проекту було створення зручної та маленької плати, на якій учні могли просто вивчати базис комп'ютерних наук у школах. Але оригінальна

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Рожкова Е.О.</i>			<i>РОЗДІЛ 2</i>	<i>Лім.</i>	<i>Арк.</i>	<i>Архивів</i>
<i>Перевір.</i>		<i>Тиханський М.П.</i>						
<i>Н. Контр.</i>		<i>Маринич І.А.</i>			<i>КНУ АКІТ-21ск</i>			
<i>Затвердив</i>		<i>Тронь В.В.</i>						

модель стала такою відомою, і знайшла своє місце на великому ринку. На сьогодні мікрокомп'ютери Raspberry широко використовуються у самих різноманітних сферах, такі як автоматизація, систем моніторингу та передачі показників, робототехніці. Користувачам вона полюбилась завдяки відносно низьку ціну, дизайн, та модульність. Також, завдяки наявності інтерфейсів USB та HDMI, вона є популярною серед комп'ютерних та електротехнічних хобістів. [10, 11, 12]

Існує безліч моделей плат малини, але в кожній з них можна виділити, базування на «системі на чипі» Broadcom та відсутність енергозалежної та вбудованої пам'яті. Замість цього плати оснащують роз'ємом для карти пам'яті.

На момент написання роботи існує 16 моделей плат: Model B, Model A, Model B+, Model A+, Model 2B, Zero, Model 3B, Zero W, Model 3B+, Model 3A+, Model 4B, Compute Module 4, 400 Unit, Pico, Zero 2 W, Model 5. [12]

В цілому Raspberry Pi це потужний одноплатний комп'ютер, який використовують для величезного спектру задач, починаючи навчання програмуванню, та закінчуючи створенням власних проектів IoT. Ось декілька особливостей, які роблять його хорошим вибором для розумного будинку:

- Плата має доступну ціну, тому для таких домашніх проектів є досить привабливим вибором.
- Ця платформа має компактний розмір, це дозволяє без зайвих зусиль вбудувати її в пристрої або системи.
- Raspberry має широкий вибір інтерфейсів та моделей, а також є можливість підключення до плати самих різних датчиків, модулів розширення та пристроїв.
- Raspberry підтримує багато різних операційних систем, таких як: Raspbian (Raspberry Pi OS), Ubuntu, Pidora, RISC OS, Windows 10 IoT, це дозволяє використовувати його у різноманітних сценаріях.
- Плата може використовуватись для контролю освітленням,

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

термостатами, системами безпеки, автоматизації побутових пристроїв та ін.

2.2 Вибір та обґрунтування прийнятих компонентів системи

2.2.1 Raspberry Pi 3B+

З кількох причин Raspberry Pi 3B+ буде чудовим вибором конкретно для роботи брокера MQTT у системі розумного будинку. Перш за все, його продуктивність достатня для обробки великої кількості повідомлень у режимі реального часу з різних пристроїв. RPі з вказаними вище характеристиками гарантує швидку та стабільну роботу брокера Mosquitto, що дозволяє ефективно керувати обміном даними між компонентами системи. [13]

Наявність вбудованих модулів Wi-Fi та Ethernet є ще одним важливим елементом, який забезпечує гнучкість підключення до мережі. Це особливо важливо для безперебійної роботи розумних будинків, де безперебійне з'єднання між пристроями є критично важливим. З'єднання Wi-Fi дозволить розмістити RPі у зручному місці без використання мережевих кабелів, а Gigabit Ethernet забезпечить високу швидкість передачі даних у ситуаціях, коли потрібне найшвидше та стабільне з'єднання. [13, 14]

Крім того, підтримка багатьох операційних систем дозволяє отримати доступ до різноманітного програмного забезпечення та інструментів. Це полегшує встановлення та налаштування важливих служб і гарантує інтеграцію всіх компонентів системи. Процес налаштування та налагодження системи значно полегшується великою кількістю ресурсів та документації, які надає велика спільнота розробників і користувачів малини.

Оскільки RPі забезпечує надійність, високу продуктивність та економічну ефективність, він є ідеальним вибором для центрального вузла керування в системі розумного будинку.

Основні характеристики:

Процесор: Cortex-A53, 4 ядра, 64-розрядний, з тактовою частотою

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

1.4GHz; [12]

Оперативна пам'ять: 1GB LPDDR2 SDRAM;

Мережеві можливості: гігабітний Ethernet (через USB 2.0), 2.4GHz та 5GHz бездротові мережі стандарту IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE; [12]

Порти вводу/виводу: 40-контактна GPIO головка, 4 USB 2.0 ports, HDMI, 3.5mm audio jack, MIPI DSI display port, MIPI CSI camera port; [12]

Сховище: слот для карти microSD;

Живлення: 5V/2.5D DC через micro-USB.



Рисунок 2.1 - Raspberry Pi 3B+ вид зверху

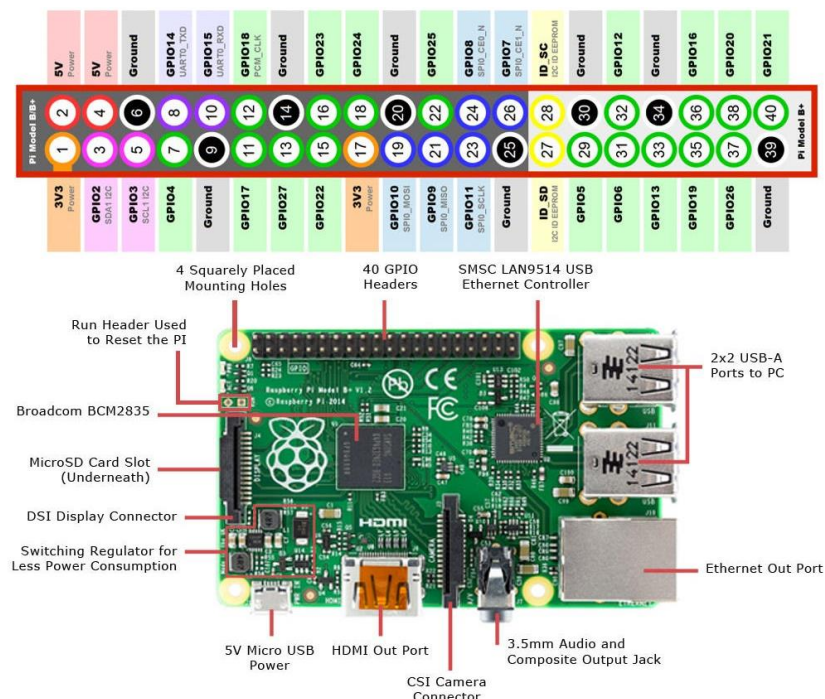


Рисунок 2.2 - Розташування елементів на платі та призначення кожного порта GPIO

2.2.2 ESP8266 CP2120

ESP8266 CP2120 – мікроконтролер від Silicon Labs, який має вбудований модуль Wi-Fi в перетворювач USB-Serial CP2120. Цей модуль використовується для підключення мікроконтролера до комп'ютера для програмування та налагодження, що робить роботу з пристроєм легкою та зручною. [30]

ESP8266 має багато переваг, які роблять його чудовим вибором для клієнтських пристроїв у системі розумного будинку.

По-перше, він ідеально підходить для бездротових мереж завдяки вбудованому модулю Wi-Fi, який дозволяє легко підключати пристрій до мережі та обмінюватися даними з центральним вузлом (в даному випадку це Raspberry Pi 3B+). Оскільки пристрої не потребують додаткових кабелів, це полегшує установку та налаштування пристроїв. [30]

По-друге завдяки підтримці ESP8266 протоколу MQTT він може ефективно спілкуватися з брокером Mosquitto, встановленим на Raspberry Pi. Підтримуючи MQTT, ESP8266 може надійно передавати дані датчиків і отримувати команди для керування приводами, що робить рішення розумного будинку гнучким і масштабованим.

Крім того, ESP8266 споживає мало енергії; це особливо важливо для пристроїв, що працюють, наприклад, від батарейок. У режимі глибокого сну споживання струму становить лише 10мкА, що значно збільшує термін служби батареї. Таким чином мікроконтролер працює достатньо добре, щоб виконувати всі завдання, які йому потрібно виконувати, наприклад обробку даних від датчиків і керування виконавчими механізмами.

ESP8266 легко інтегрувати в різні пристрої та конструкції, не займаючи багато місця та не збільшуючи загальну вагу системи завдяки своєму невеликому розміру та вазі. Це особливо важливо при побудові компактних та мобільних систем.

Отже, завдяки своїм технічним характеристикам і можливостям ESP8266 є чудовим варіантом для впровадження клієнтських пристроїв у

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

систему розумного будинку, оскільки він гарантує бездротове з'єднання, підтримку MQTT і має низьке використання енергії.

Основні характеристики:

Мікропроцесор: Tensilica L106 32-бітний RISC; [22]

Частота процесору: 80MHz (можливий розгін до 160MHz); [22]

Пам'ять: 32KB інструкції + 80KB SRAM; [22]

Вбудована флеш-пам'ять: 4 MB;

Wi-Fi: 802.11 b/g/n, підтримка Wi-Fi Direct (P2P), Soft-AP; [22]

Робоча напруга: 3.3V;

GPIO-піни: до 17;

Аналоговий вхід (ADC): один 10-бітний вхід;

USB-to-Serial конвертер: CP2120, підтримка USB 2.0;

Підтримка струму: до 170mA при передаванні даних, 10mA в режимі глибокого сну; [22]

Розміри: 24мм x 16мм;

Вага: приблизно 2г.



Рисунок 2.3 - ESP8266 CP2120 вид зверху

2.2.3 Датчики DHT11, MQ-2 та модуль активного зумера YL-44

Модуль активного зумера YL-44:

Зумер – це електроний пристрій, який генерує аудіосигнали в діапазоні слухового сприйняття. Активний зумер YL-44 може виробляти звукові сигнали, якщо йому піддається напруга. Активний зумер, на відміну від пасивних зумерів, має вбудований генератор, що полегшує використання,

оскільки він не потребує зовнішніх сигналів для роботи. [25]

В системах домашньої автоматизації може бути корисним для сповіщення подій, таких як відтворення мелодії при відкриванні входних дверей або проникнення в будівлю. Але конкретно для цього проекту він буде використаний як сигналізація про небезпеку: витік чадного газу або детектування диму. Це дозволить мешканцям оперативно відреагувати на загрозу.

Характеристики:

Тип зумера: активний;

Тип звуку: аналогічний динаміку для ініціалізації системного блоку комп'ютера; [25]

Напруга: 3.3 – 5V;

Максимальний струм: 20мА;

Частота звукового сигналу: 2.5кГц (± 0.5 кГц); [25]

Рівень звуку: ≥ 85 дБ на відстані 10см; [25]

Інтерфейс: Цифровий сигнал (включення/виключення) [25]

Розміри: 27 x 13 x 8 мм;

Довжина петлі: 250 мм;

Вага: 3 г.



Рисунок 2.4 - Модуль п'єзодинаміка(зумер) YL-44

Модуль датчика температури та вологості DHT11:

Модуль DHT11 це датчик температури та вологості та може бути корисним компонентом для моніторингу і контролю кліматичних умов в приміщенні. Він забезпечує стабільні показання та легко інтегрується з

					КНУ КРБ.151.24.02.02.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

мікроконтролерами, такими як ESP8266. [23]

У серії датчиків DHT є більш нові та досконалі моделі, але їх доцільно використовувати в системах, де є дійсними потреби з великої точності вимірювання. Для домашньої автоматизації він цілком підходить, оскільки тут велика точність вимірювань не відіграє важливої ролі.

Характеристики:

Робоча напруга: 3.3 - 5V;

Діапазон температур: 0 – 50C°;

Похибка показань температури: ± 2C°;

Діапазон вологості: 20 – 90%;

Похибка показань вологості: ± 5%;

Інтерфейс: однопровідний цифровий сигнал; [23]

Споживання струму: 0.3mA при вимірах, до 1mA в піковому режимі;
[23]

Розміри: 15.5 x 12 x 5.5мм;

Вага: 2г.



Рисунок 2.5 - Модуль датчику температури та вологості DHT11

Модуль датчика диму MQ-2:

MQ-2 – це твердотільний датчик газу, призначений для виявлення легкозаймистих парів, таких як метан, пропан, метан, водні пари та дим. Він працює, коли внутрішня провідність датчика змінюється в присутності цільових газів. [26]

Модуль датчика диму є популярним і надійним рішенням для детектування диму та легкозаймистих газів. Його ідеально використовувати в

проектах розумного дому з моніторингом якості повітря та безпеки, оскільки він чутливий до багатьох газів.

Характеристики:

Робоча напруга: 5V;

Діапазон виявлення: 300 до 10000ppm (часток на мільйон); [26]

Чутливість: Дим: 100 - 10000ppm; Пропан: 200 - 5000ppm; Метан: 500 - 10000ppm; Водень: 100 - 2000ppm; Бутан: 300 - 5000ppm; [26]

Час відгуку: <10 секунд;

Робоча температура: -20 до 50С°;

Споживаний струм: 150mA (максимум); [26]

Інтерфейс: аналоговий вхід, цифровий вихід через компаратор; [26]

Розміри: 32 x 22 x 27мм;

Вага: приблизно 5г.



Рисунок 2.6 - Модуль датчику диму MQ-2 вид знизу



Рисунок 2.7 - Модуль датчику диму MQ-2 вид зверху

2.2.4 Світлодіодна стрічка WS2812

WS2812 – це адресна світлодіодна стрічка з контролером у кожному світлодіоді. Це дозволяє використовувати один керуючий сигнал для керування яскравістю та кольором кожного світлодіода. Вона надає яскраві, насичені кольори з високою точністю кольору. [28]

Світлодіодна стрічка WS2812 добре підходить для візуалізації інтенсивності освітлення та контролю кольору в системах розумного будинку. Оскільки ця світлодіодна стрічка надає можливість керувати кожним світлодіодом, вона відрізняється від інших освітлювальних приладів, таких як звичайні світлодіодні лампи або неадресні RGB-стрічки. Це дає більшу гнучкість і дозволяє створювати різноманітні світлові ефекти, складні анімації, плавні переходи та динамічні світлові ефекти. Цього неможливо досягти звичайними освітлювальними пристроями. [28]

Стрічка WS2812 має високу гнучкість керування, оскільки вона має цифровий інтерфейс. Це полегшує програмування та інтеграцію з мікроконтролером, таким як ESP8266. Це робить його більш зручним та універсальним для створення розумних систем освітлення.

Компактний розмір і проста установка дозволяють стрічці бути легко встановленою в важкодоступних місцях. За потреби її можна розрізати та з'єднати, що забезпечить гнучкість встановлення та налаштування.

Також перевагою цієї стрічки є те, що світлодіоди споживають менше енергії, ніж традиційні освітлювальні прилади, що робить її також економічно ефективною.

Характеристики:

Робоча напруга: 5V;

Кількість світлодіодів: 60 на метр (в даній стрічці - 23);]

Максимальний споживаний струм: приблизно 60mA на один світлодіод при максимальній яскравості (всі три кольори ввімкнені на 100%); [27]

Кольори: RGB (16 мільйонів кольорів);

Управління: однопровідний цифровий інтерфейс;

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Розміри: ширина стрічки – 10мм, висота – 3мм;

Кут огляду: 120 °;

Вологозахист: IP20 (внутрішнє використання).



Рисунок 2.8 - Світлодіодна стрічка WS2812

2.3Схема підключення компонентів

Перш ніж створювати прототип системи «розумний дім», необхідно ретельно розробити та реалізувати схему підключення всіх частин. Далі будуть розглянуті всі схеми, необхідні для правильного функціонування системи та інтеграції всіх її компонентів.

Основні компоненти системи включають звукову сигналізацію, світлодіодну стрічку, датчики, мікроконтролер ESP8266. Для забезпечення надійної роботи та взаємодії з центральним брокером MQTT, встановленим на Raspberry Pi 3B+, кожен із цих компонентів потребує відповідного підключення.

2.3.1 Функціональна схема системи

Функціональна схема системи показує, як усі компоненти системи

розумного дома взаємодіють між собою. Це допомагає зрозуміти загальну структуру системи, визначити основні компоненти системи та їх функціональні ролі, а також переконатися, що всі частини системи взаємодіють одна з одною правильно.

Основні елементи функціональної схеми системи:

1. Raspberry Pi 3B+ - центральний брокер.

Raspberry Pi 3B+ налаштований як брокер, і він має встановлені Mosquitto, Node-RED і Node-RED Dashboard. Ця важлива частина системи відповідає за отримання, обробку та передачу даних між користувачами та компонентами системи. Raspberry Pi служить координаційною точкою, що дозволяє всім компонентам взаємодіяти через протокол MQTT.

2. Мікроконтролер ESP8266.

На макетній платі кожен компонент безпосередньо підключений до мікроконтролера ESP8266. Це важливий вузол, який працює з актуаторами та датчиками, збирає дані та передає їх через протокол MQTT брокеру.

3. Підключені компоненти.

Датчик температури та вологості DHT11 – це модуль, який вимірює температуру та вологість навколишнього середовища, а потім передає ці дані на ESP8266.

Модуль датчика диму MQ-2: датчик виявляє наявність диму та чадного газу та передає інформацію на ESP8266. Сигналізація спрацьовує, коли датчик спрацьовує.

Активний модуль зумера YL-44: коли датчик диму MQ-2 спрацьовує, зумер подає сигнал тривоги.

Світлодіодна стрічка WS2812 працює за допомогою мікроконтролера ESP8266 і дозволяє змінювати кольори та інтенсивність світла через веб-інтерфейс.

4. Обмін даними через MQTT.

Протокол MQTT за допомогою Mosquitto дозволяє передавати команди управління та дані з усіх датчиків між ESP8266 і Raspberry Pi. Це гарантує

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

швидкий і надійний зв'язок усередині системи.

5. Node-Red і Node-RED Dashboard.

Node-RED обробляє та маршрутизує дані на Raspberry Pi. Користувачі можуть відстежувати показання датчиків і керувати параметрами системи, такими як колір і інтенсивність світлодіодної стрічки, використовуючи веб-інтерфейс інформаційної панелі Node-RED.

6. Підключення через Wi-Fi.

Через маршрутизатор усі елементи системи, включаючи Raspberry Pi, мікроконтролер ESP8266 і пристрої користувача, такі як комп'ютери та смартфони, підключені до однієї мережі Wi-Fi. Це дозволяє підключитися до системи з будь-якого місця в мережі.

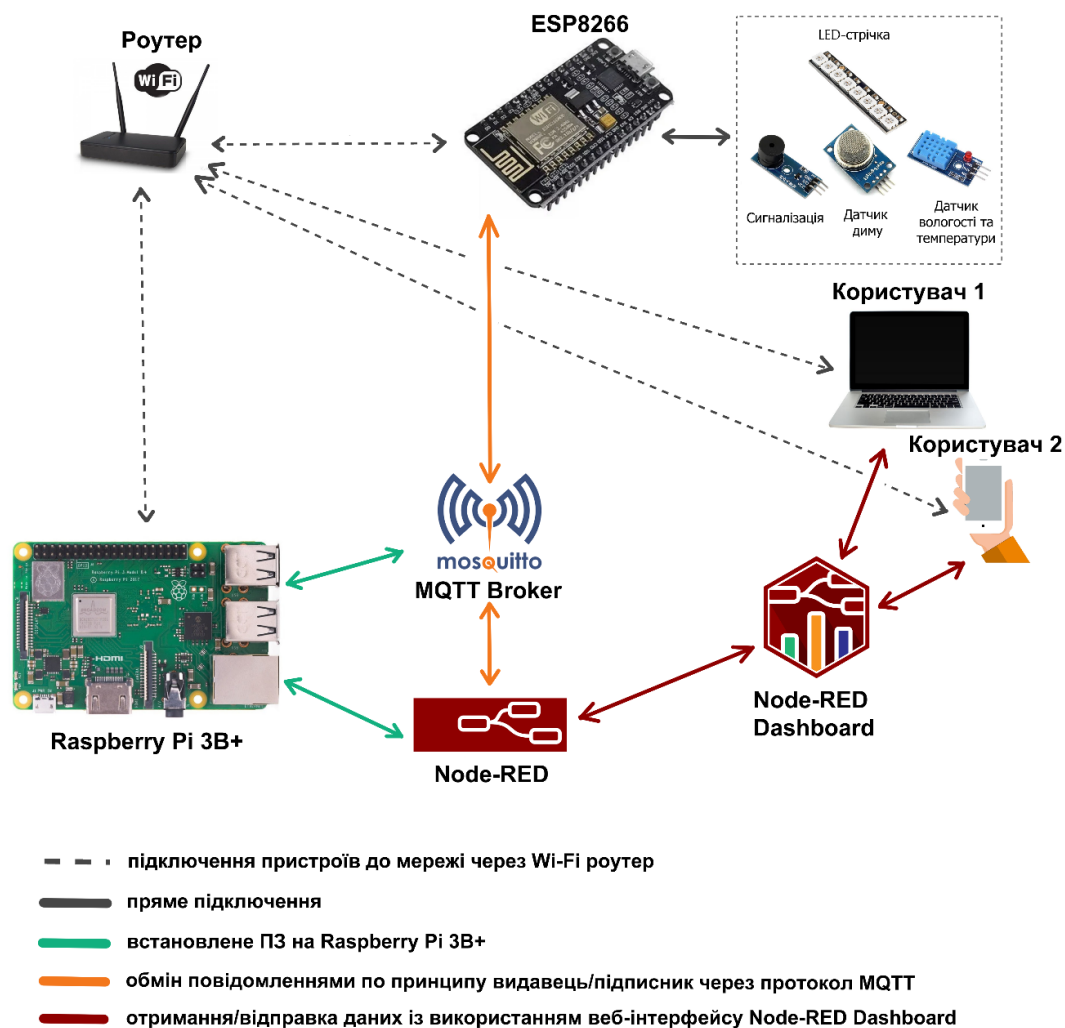


Рисунок 2.9 - Функціональна схема системи

2.3.2 Схема підключення компонентів до мікроконтролера ESP8266 на макетній платі

Перед побудовою схеми на макетній платі було вирішено спроектувати схему в програмі Fritzing. Цей підхід є дуже корисним, оскільки надає безліч переваг на етапі розробки. Насамперед, це дозволяє візуалізувати схему, що робить її легко зрозумілою та доступною для аналізу. Це допомагає бачити загальну картину проекту та заздалегідь враховувати, як будуть з'єднані всі компоненти.

Крім того, за допомогою попереднього проектування можна заздалегідь планувати розташування кожного компонента на макетній платі. Це особливо важливо, коли компоненти мають особливі розміри або коли простір на платі обмежений. Fritzing полегшує процес складання схеми, дозволяючи заздалегідь планувати розташування компонентів. З готовим віртуальним проектом легко зрозуміти, куди і як підключати кожен компонент, що скорочує час монтажу та знижує ймовірність помилок.

Крім того, це чудова документація для проекту. Це корисно для вивчення, а також для допрацювань.

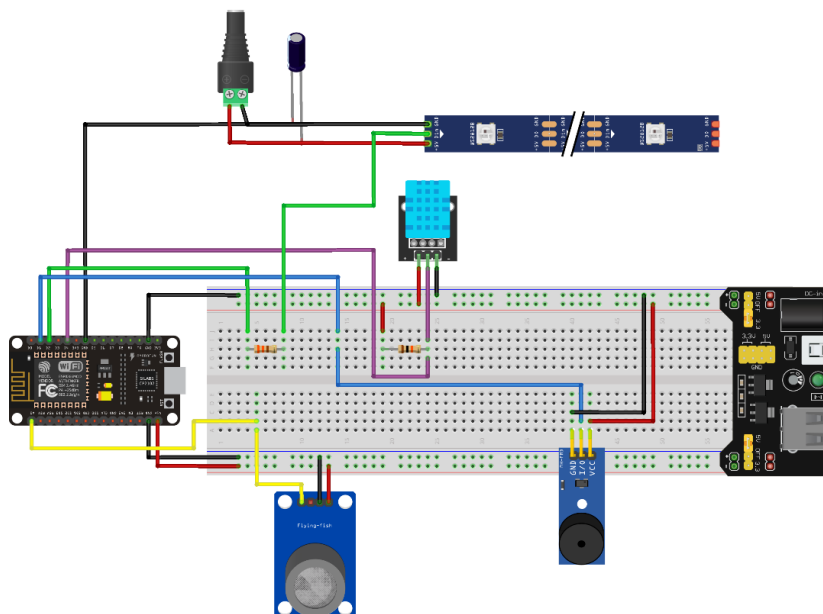


Рисунок 2.10 - Схема підключення компонентів на макетній платі в програмному застосунку Fritzing

Приступимо до безпосереднього з'єднання елементів на макетній платі. Ця система використовує макетну плату, яка отримує енергію від модуля живлення MB-102. Цей модуль живлення призначений для забезпечення постійної напруги та струму, необхідних підключеним електронним компонентам. Важливо зауважити, що на одній стороні лінії живлення макетної плати встановлена перемичка на 3.3В, тоді як на іншій стороні перемичка встановлена на 5В. Це дозволяє одночасно використовувати компоненти, які мають різні вимоги до напруги. При вхідній напрузі від 7В до 12В модуль MB-102 може видавати струм до 700мА. [24]

NodeMCU ESP8266 CP2102 — це основний мікроконтролер системи. Два його контакти GND підключені до мінусів з обох боків макетної плати, що допомагає забезпечити загальне заземлення. Вивід Vin підключається до лінії живлення 5В, яка надає мікроконтролеру напругу, яку він потребує для роботи. Також можна для живлення мікроконтролера підключити його пін 3.3В до лінії живлення відповідно на 3.3В, але використання піна Vin для живлення є більш переважним. До нього можна підключати джерела живлення до 20В, оскільки плата має вбудований перетворювач.

Модуль DHT11, який використовується для вимірювання температури та вологості, підключається за наступною схемою. Підключення: джерело живлення модуля (VCC) підключається до лінії 3.3В, а контакт GND підключається до мінуса на макетній платі. Контакт DATA підключений до цифрового контакту D4 NodeMCU (GPIO2). Також був використаний pull-up резистор на 10 кОм між лінією живлення та виводом DATA для більш коректної роботи модуля та захисту схеми.

Датчик газу MQ-2 підключається до мережі живлення 5В для забезпечення необхідної напруги. Вивід GND модуля підключається до мінуса. Оскільки даний модуль використовує аналоговий сигнал при спрацьовуванні датчику, аналоговий вихід модуля підключений до входу A0 на NodeMCU. Дискретний вихід модуля не використовується в цій конфігурації.

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

Активний модуль зумера YL-44 підключається до наступної лінії живлення: джерело живлення (VCC) модуля підключається до лінії 3.3В, а контакт GND підключається до мінусу цієї лінії. До NodeMCU вихід вводу/виводу модуля підключено до цифрового контакту D1 (GPIO5). Це дозволяє мікроконтролеру активувати зумер, якщо це відповідає умові.

Адресна світлодіодна стрічка WS8212 живиться від альтернативного джерела живлення - блоку живлення 5В з максимальним струмом 2А. До цього джерела живлення підключені штирі живлення та заземлення стрічки. [27]

Щоб забезпечити загальне заземлення для всієї схеми, додатковий провід заземлення під'єднується до одного з контактів GND на NodeMCU. Вивід даних стрічки підключений до цифрового контакту D2 (GPIO4) на NodeMCU через резистор 330 Ом, який допомагає захистити вхід стрічки від стрибків напруги. Між лініями живлення стрічки встановлений конденсатор ємністю 1000 мкФ і напругою 6,3 В, що сприяє згладжуванню можливих пульсацій напруги та забезпечує стабільну роботу стрічки.

Основна проблема при проектуванні цієї схеми полягає в тому, щоб забезпечити стабільне живлення всіх компонентів. Модуль живлення MB-102 забезпечує роздільне живлення для компонентів, які вимагають різних рівнів напруги (3.3В і 5В). Використання додаткового джерела живлення для світлодіодної стрічки WS8212 допомагає розвантажити основний модуль живлення та запобігти можливим перевантаженням.

Для забезпечення надійних з'єднань всі дроти і контакти необхідно ретельно перевірити на цілісність і якість з'єднання. Це передбачає перевірку з'єднувальних проводів за допомогою мультиметра, щоб переконатися у відсутності обривів або поганих з'єднань.

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

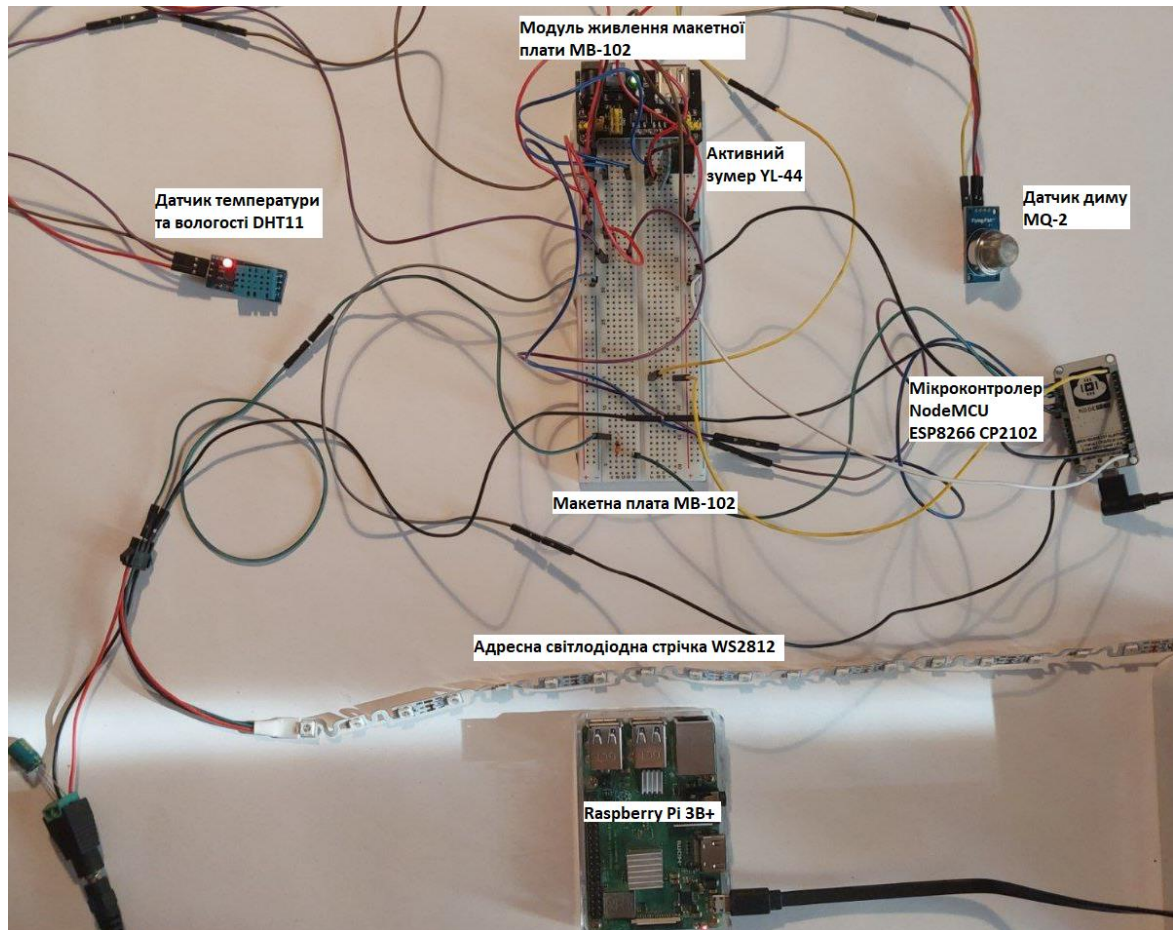


Рисунок 2.11 - Створений макет системи

2.4 Первинне налаштування Raspberry Pi 3B+

2.4.1 Установка ОС Raspbian та перший запуск

Процес встановлення операційної системи Raspbian на мікрокомп'ютер Raspberry Pi 3B+ починається з підготовки необхідних компонентів і програмного забезпечення. У цьому випадку використовується карта пам'яті SanDisk Ultra microSDHC UHS-I Card 16GB, кардрідер, джерело живлення 5V і 3A, кабель microUSB, клавіатура, миша, монітор і конвертер HDMI to VGA.

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

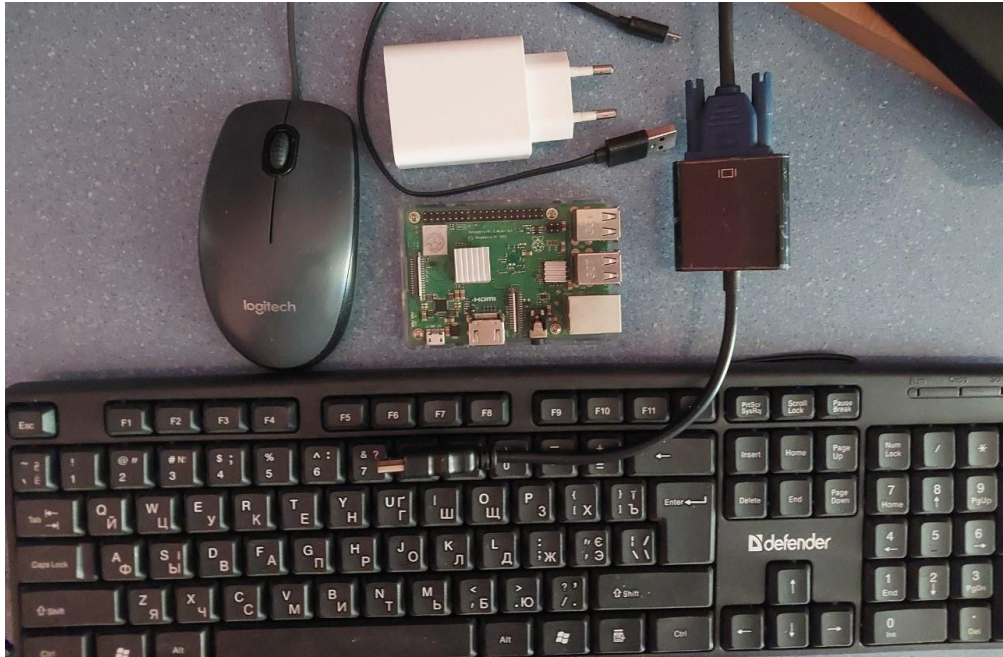


Рисунок 2.12 - Необхідні компоненти для первинного запуску Raspberry Pi 3B+

Спочатку потрібно підготувати карту пам'яті. Картка SanDisk Ultra microSDHC UHS-I 16 ГБ за допомогою кардрідера підключається до комп'ютера. Після завантажується остання версія Raspbian з офіційного веб-сайту Raspberry Pi Foundation. Важливо переконатися в завантаженні саме того образу Raspbian, який сумісний із Raspberry Pi 3B+. Після завантаження образу необхідна програма для запису образу на карту пам'яті. Однією з найпопулярніших і зручних програм для цього є Balena Etcher.



Рисунок 2.13 - Картка SanDisk Ultra microSDHC UHS-I 16 ГБ

Після успішного запису зображення на карту пам'яті необхідно вставити її в слот microSD на Raspberry Pi 3B+. Після всіх підключень треба

ввімкнути живлення. Raspberry Pi почне завантажуватися з карти пам'яті. На екрані монітора з'явиться процес завантаження операційної системи Raspbian. [11]

Коли завантаження завершиться, з'явиться робочий стіл Raspbian. У цей час як раз можна виконати початкове налаштування системи, включаючи вибір мови, налаштування мережі та оновлення системи. [11]

У результаті операційна система Raspbian буде успішно встановлена та готова до використання на Raspberry Pi 3B+. Процес інсталяції, починаючи з підготовки карти пам'яті і закінчуючи первинним налаштуванням системи, забезпечує готовність пристрою до подальшої експлуатації та виконання поставлених завдань.

2.4.2 Встановлення статичного IP для плати Raspberry

Налаштування статичної IP-адреси може бути корисним у випадках, коли потрібна постійна та прогнозована мережна адреса для доступу до Raspberry Pi. Програми, які працюють на серверах, мережевих пристроях або інших завданнях, які потребують постійної адреси мережі, потребують цього особливо.

Спочатку потрібно відкрити термінал Raspberry Pi, щоб визначити поточну мережеву конфігурацію. Це можна зробити за допомогою застосування «Термінал», де потрібно ввести команду «hostname -I». Це виведе на термінал поточну IP-адресу пристрою.

Далі наберіть команду «ip r». Для налаштування статичного IP вона покаже таблицю маршрутизації, з якої можна дізнатися поточний шлюз. Для безпосереднього налаштування статичної адреси потрібно відредагувати конфігураційний файл мережі. Для редагування з правами суперкористувача із використанням текстового редактору, треба ввести команду «sudo nano /etc/dhcpd.conf».

В даному файлі були внесені такі зміни:

«interface wlan0» - був вказаний мережевий інтерфейс для бездротового

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

з'єднання;

«static ip_address» - тут необхідно вказати статичну IP-адресу та маску підмережі, було обрано 192.168.0.102/24;

«static routers» - в даній строчці було вказано IP-адрес маршрутизатора, а саме 192.168.0.1;

«static domain_name_servers» - був вказаний DNS-сервер Google.

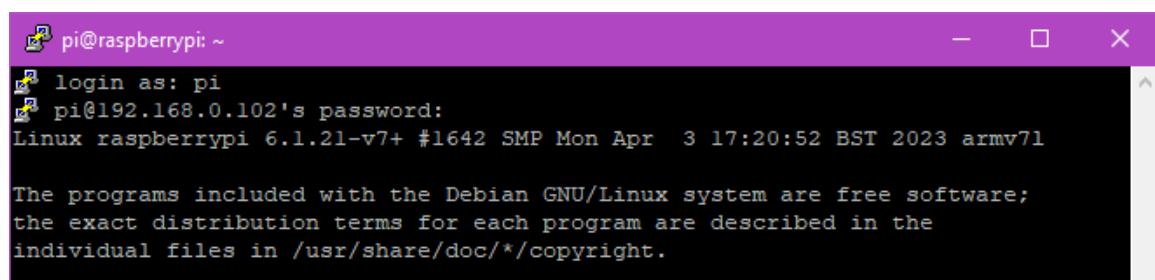
Після всіх змін та збереження файлу треба перезавантажити пристрій, використовуючи команду «sudo reboot». Це застосує внесені зміни.

2.4.3 Налаштування віддаленого доступу до плати

Налаштування віддаленого доступу до Raspberry Pi дозволяє зручно управляти пристроєм через командну строку з іншого пристрою, використовуючи протокол SSH.

Спочатку необхідно переконатися, що малинка під'єднана до мережі Інтернет. Далі потрібно ввімкнути SSH на Raspberry. Досить часто, SSH за замовчуванням вимкнений, з питань безпеки. Для того, щоб його ввімкнути, потрібно в терміналі Raspberry виконати команду «sudo raspi-config» та активувати SSH в файлі конфігурації, а потім перезавантажити пристрій. Після цих дій можна запускати PuTTY на комп'ютері.

Для підключення вводиться встановлена статична IP-адреса мікрокомп'ютера, вказується порт 22, який є стандартним для SSH, та вказується саме цей протокол для з'єднання. Також сесія була збережена, завдяки чому всі ці параметри не треба буде вводити кожен раз при підключенні.



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.0.102's password:  
Linux raspberrypi 6.1.21-v7+ #1642 SMP Mon Apr 3 17:20:52 BST 2023 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

Рисунок 2.14 - Вдале підключення до мікрокомп'ютера по SSH

2.5 Налаштування та конфігурація MQTT брокера

MQTT-брокер виконує роль централізованого посередника в системі обміну повідомленнями між різними пристроями та додатками. Після запуску брокера, він встановлює параметри підключення, включаючи порти, протоколи та налаштування безпеки. Далі брокер приймає підключення від клієнтів, які можуть бути як видавцями, так і передплатниками повідомлень. Клієнти, такі як датчики, мікроконтролери або програмні програми, підключаються до брокера і підписуються на певні топіки або публікують у них повідомлення. [29]

Коли клієнт публікує повідомлення на конкретний топик, брокер приймає це повідомлення і пересилає його всім клієнтам, підписаним на цей топик. Брокер обробляє безліч паралельних з'єднань, забезпечуючи надійну доставку повідомлень навіть за високого навантаження. Якщо повідомлення не може бути доставлене негайно, брокер може зберігати його доти, доки підписник не стане доступним (якщо використовується політика якості обслуговування QoS). [29]

Брокер також управляє політиками безпеки, включаючи автентифікацію клієнтів та контроль доступу до топиків, що запобігає несанкціонованому доступу до даних.

Взаємодія між клієнтами через брокер дозволяє створювати масштабовані та гнучкі системи обміну даними, де пристрої та програми можуть спілкуватися один з одним незалежно від фізичного розташування та мережових умов. Брокер забезпечує централізоване управління трафіком повідомлень та логікою маршрутизації, що спрощує інтеграцію різних компонентів у єдину систему.

На основі цього була складена блок-схема алгоритму для MQTT-брокера.

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		48

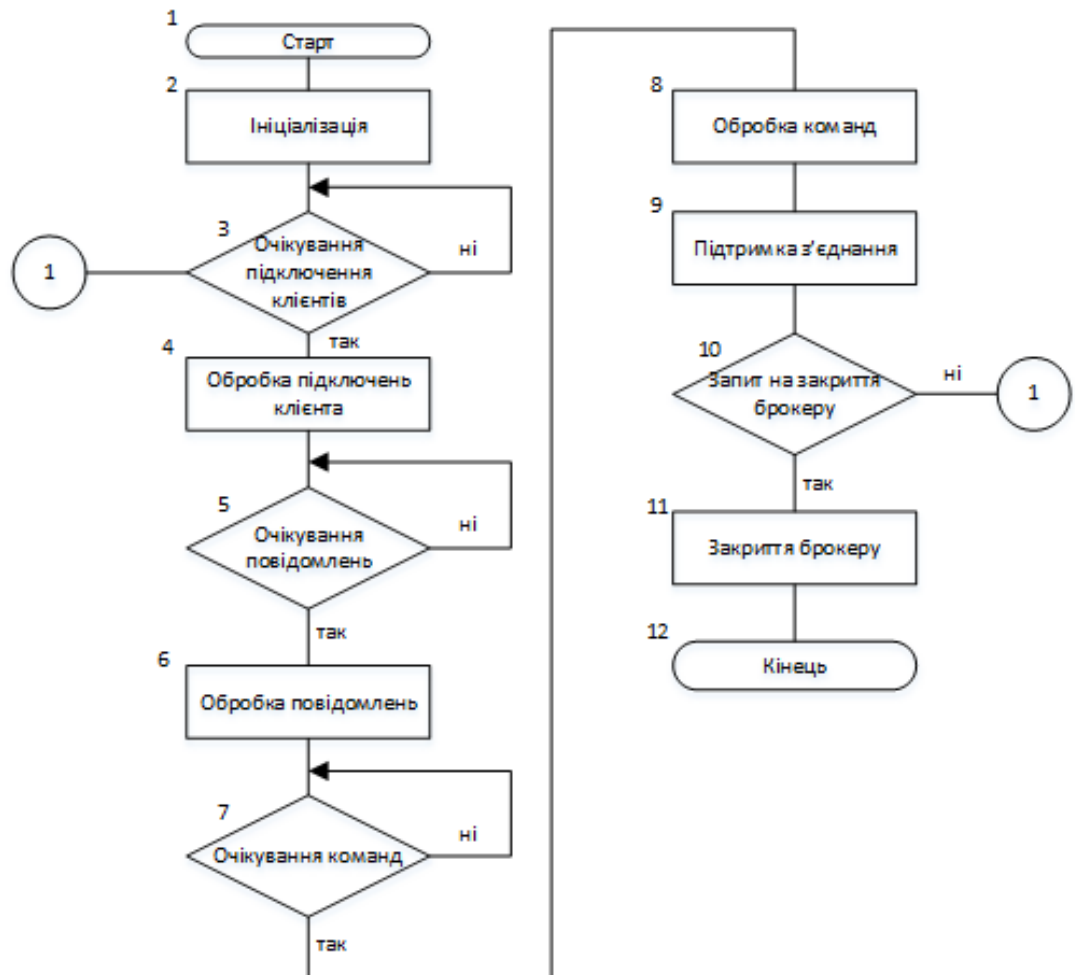
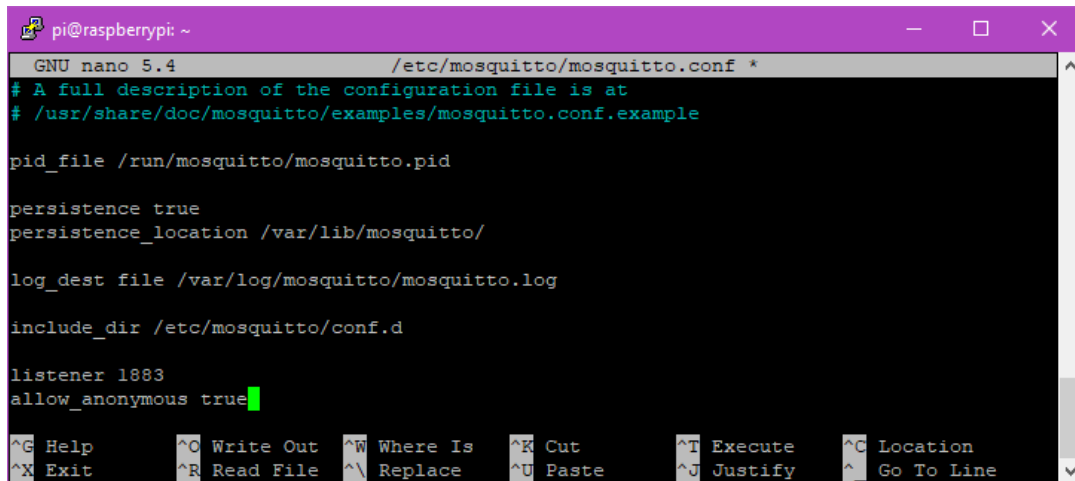


Рисунок 2.15 - Блок-схема алгоритму MQTT-брокера

2.5.1 Установка та налаштування Mosquitto

Щоб встановити сам Mosquitto та клієнтські утиліти, які дозволяють тестувати з'єднання та обмінюватися повідомленнями, треба виконати команду «`sudo apt-get install mosquitto mosquitto-clients`». По завершенню встановлення треба буде налаштувати конфігураційний файл Mosquitto «`/etc/mosquitto/mosquitto.conf`». Оскільки конфігурація буде спрощеною, треба переконатися, що рядки `listener 1883` та `allow_anonymous true` присутні в файлі. Це надає можливість клієнтам підключатися без автентифікації. [29]



```
pi@raspberrypi: ~
GNU nano 5.4 /etc/mosquitto/mosquitto.conf *
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

listener 1883
allow_anonymous true
```

Рисунок 2.16 - Зміст конфігураційного файлу /etc/mosquitto/mosquitto.conf

Після налаштування конфігураційного файлу необхідно запустити сервіс Mosquitto та налаштувати його на автозапуск під час завантаження системи. Для цього виконуються команди «`sudo systemctl start mosquitto`», щоб запустити брокер, та «`sudo systemctl enable mosquitto`», щоб увімкнути автозапуск Mosquitto під час старту системи.

2.5.2 Налаштування Node-RED та Node-RED Dashboard

Для установки Node-Red спочатку встановлюється Node.js із використанням команди «`sudo apt install -y nodejs npm`», та по установці виконується команда для завантаження самого Node-RED «`sudo npm install -g --unsafe-perm node-red`». Також, напряму через термінал був встановлений пакет Node-RED Dashboard, для цього в терміналі вводилася команда «`sudo npm install -g node-red-dashboard`». [31]

Для налаштування автозапуску при завантаженні системи використовується служба `systemd`. В конкретному випадку необхідно створити новий файл служби наступною командою «`sudo nano /lib/systemd/system/node-red.service`» та в нього вписується відповідний текст. [31]

Після збереження змін редактор можна закрити, а служба оновлюється такою послідовністю команд: «`sudo systemctl daemon-reload`», «`sudo systemctl`

enable node-red», «sudo systemctl start node-red».

Відтепер Node-RED буде автоматично запускатися при завантаженні системи та його роботу можна буде перевірити, переходячи за адресою «<http://192.168.0.102:1880>» в браузері.

2.6 Розробка програмного забезпечення для ESP8266

Програмування ESP8266 виконується мовою Python, яка забезпечує зручність і гнучкість у розробці додатків, що вбудовуються. Одним із найпоширеніших фреймворків для програмування ESP8266 на Python є MicroPython. Середовище розробки MicroPython є інтерпретатором Python, оптимізованим для роботи на мікроконтролерах і вбудовуваних системах. Вона надає зручний спосіб створення програмного забезпечення для таких пристроїв, дозволяючи використовувати знайомий синтаксис мови Python та стандартну бібліотеку мови. [32]

ESP8266 може бути використаний для безлічі завдань, таких як збір даних від датчиків, керування пристроями, зв'язок із зовнішніми сервісами через інтернет та багато іншого. Його вбудований модуль Wi-Fi дозволяє підключатися до бездротових мереж, що робить його ідеальним для створення IoT пристроїв, які можуть взаємодіяти з інтернетом.

Вбудовані апаратні інтерфейси ESP8266 дозволяють підключати до нього різні периферійні пристрої, такі як датчики температури, вологості, диму та інші. Це забезпечує широкі можливості для моніторингу навколишнього середовища та створення систем розумного будинку та безпеки.

Важливим аспектом роботи з ESP8266 є взаємодія з брокером MQTT. ESP8266 може бути легко налаштований для підключення до брокера MQTT, що дозволяє обмінюватися даними з іншими пристроями та хмарними сервісами.

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

інтерфейси, керування живленням і т.д.;

dht - бібліотека для роботи з датчиками температури та вологості DHT. Вона забезпечує функції для зчитування даних з датчика та представлення їх у зручній формі для подальшої обробки; [32]

network – бібліотека для роботи з мережею, включаючи підключення до Wi-Fi, встановлення IP-адреси та інші параметри мережі;

neopixel – бібліотека для роботи з адресними світлодіодами Neopixel, такими як WS2812;

MQTTClient (з umqtt.simple) - це бібліотека для роботи з протоколом MQTT. Вона дозволяє встановлювати з'єднання з MQTT брокером, підписуватися на топики, надсилати та приймати повідомлення.

Далі слід визначити параметри підключення, такі як SSID та пароль для Wi-Fi, а також адресу та порт MQTT брокера. Ці параметри необхідні для успішного підключення до відповідних мереж та серверів.

Ініціалізація апаратних пристроїв здійснюється шляхом визначення відповідних пінів та налаштувань для кожного пристрою. Наприклад, для датчика DHT11 вказується пін, до якого він підключений, а для світлодіодної стрічки визначається кількість світлодіодів і пін, до якого вона підключена.

Для взаємодії з MQTT брокером визначаються функції підключення до брокера, підписки на топики та обробки отриманих повідомлень. Це дозволяє керувати пристроями на основі отриманих команд і надсилати дані на брокер для подальшої обробки.

```
1 #  
2 #Параметри підключення к Wi-Fi  
3 SSID = 'morojenko' #Назва Wi-Fi мережі  
4 PASSWORD = r'morojenko))' #Пароль від Wi-Fi мережі  
5  
6 #Параметри підключення к MQTT брокеру  
7 MQTT_BROKER = '192.168.0.102' #IP-адреса MQTT брокера  
8 MQTT_PORT = 1883 #Порт для підключення до MQTT брокера  
9 MQTT_TOPIC_DHT11 = 'home/sensors/dht11' #Топік для відправки даних з DHT11  
10 MQTT_TOPIC_ALARM = 'home/sensors/alarm' #Топік для відправки даних об активації сигналізації  
11 MQTT_TOPIC_LED_BRIGHTNESS = 'home/led/brightness' #Топік для управління яскравістю світлодіодної стрічки  
12 MQTT_TOPIC_LED_COLOUR = 'home/led/colour' #Топік для управління кольором світлодіодної стрічки  
13 MQTT_TOPIC_LED_POWER = 'home/led/power' #Топік для вмикання/вимкання світлодіодної стрічки
```

Рисунок 2.18 - Лістинг ініціалізації підключення до Wi-Fi та MQTT-брокеру на microPython

```

21 #Налаштування підключення до Wi-Fi
22 def connect_wifi():
23     wifi = network.WLAN(network.STA_IF) #Створення об'єкта WLAN в режимі Station
24     wifi.active(True) #Активізація Wi-Fi
25     wifi.connect(SSID, PASSWORD) #Підключення до Wi-Fi мережі
26     while not wifi.isconnected(): #Очікування підключення до мережі
27         pass
28     print('Connected to WiFi:', wifi.ifconfig()) #Виведення інформації о підключенні
29
30 #Налаштування MQTT клієнта
31 def connect_mqtt():
32     client = MQTTClient('esp8266', MQTT_BROKER, port=MQTT_PORT) #Створення об'єкта MQTTClient
33     client.set_callback(mqtt_callback) #Встановлення функції обратного зв'язку
34     client.connect() #Підключення до MQTT брокеру
35     client.subscribe(MQTT_TOPIC_LED_BRIGHTNESS) #Підписка на топик управління яскравістю
36     client.subscribe(MQTT_TOPIC_LED_COLOUR) #Підписка на топик управління кольором
37     client.subscribe(MQTT_TOPIC_LED_POWER) #Підписка на топик управління вмиканням/вимиканням
38     print('Connected to MQTT broker') #Виведення повідомлення про вдале підключення
39     return client #Повернення об'єкта клієнту

```

Рисунок 2.19 - Лістинг з налаштуванням підключення до Wi-Fi та MQTT-брокеру

```

62 #Зворотній виклик для обробки повідомлень MQTT
63 def mqtt_callback(topic, msg):
64     global led_on, led_color, led_brightness #Об'явлення глобальної змінної
65     if topic == MQTT_TOPIC_LED_POWER: #Перевірка повідомлення топіку
66         if msg == b'ON': #Якщо повідомлення "ON"
67             led_on = True #Вмикання світлодіодної стрічки
68         elif msg == b'OFF': #Якщо повідомлення "OFF"
69             led_on = False #Вимикання світлодіодної стрічки
70     elif topic == MQTT_TOPIC_LED_BRIGHTNESS: #Якщо топик для управління яскравістю
71         try:
72             led_brightness = int(msg.decode('utf-8')) #Отримання значення яскравості
73             if led_brightness < 0:
74                 led_brightness = 0
75             elif led_brightness > 255:
76                 led_brightness = 255
77         except ValueError:
78             print("Invalid brightness value") #Виведення повідомлення про помилку
79     elif topic == MQTT_TOPIC_LED_COLOUR: #Якщо топик для управління кольором
80         try:
81             parts = msg.decode('utf-8').split(',') #Розбивання повідомлення на частини
82             r = int(parts[0]) #Отримання значення червоного кольору
83             g = int(parts[1]) #Отримання значення зеленого кольору
84             b = int(parts[2]) #Отримання значення блакитного кольору
85             led_color = (r, g, b) #Встановлення кольору
86         except:
87             print("Invalid color value") #Виведення повідомлення про помилку

```

Рисунок 2.20 - Лістинг встановлення зворотнього зв'язку для обробки повідомлень по MQTT

Основна функція програми виконує читання даних із датчиків, оновлення стану світлодіодної стрічки та відправлення даних на MQTT брокер. При цьому передбачено обробку можливих помилок, таких як недоступність датчиків або втрата з'єднання з брокером.

```

102 #Основна функція
103 def main():
104     connect_wifi() #Підключення до Wi-Fi
105     client = connect_mqtt() #Підключення до MQTT
106
107     while True:
108         try:
109             #Зчитування даних з DHT11
110             dht_sensor.measure() #Запуск виміру
111             temp = dht_sensor.temperature() #Зчитування температури
112             hum = dht_sensor.humidity() #Зчитування вологості
113             print('Temperature:', temp, 'C') #Виведення температури
114             print('Humidity:', hum, '%') #Виведення вологості
115
116             #Формування повідомлення для відправки
117             payload = '{ "temperature": %d, "humidity": %d }' % (temp, hum) #Формування строки з даними
118             client.publish(MQTT_TOPIC_DHT11, payload) #Відправлення даних на брокер
119             print('Data sent to MQTT broker') #Виведення повідомлення про відправку
120
121             #Зчитування даних з MQ-2
122             smoke_level = mq2_pin.read() #Зчитування рівня диму
123             print('Smoke level:', smoke_level) #Виведення рівня диму
124
125             #Перевірка рівня диму з пороговим значенням
126             if smoke_level > ALARM_THRESHOLD and not alarm_triggered: #Якщо рівень диму перевищує порогове значення
127                 buzzer_pin.on() #Ввімкнення зумеру
128                 alarm_triggered = True #Встановлення флагу активності тривоги
129                 alarm_payload = '{ "alarm": "Smoke detected!" }' #Формування повідомлення про активацію сигналізації
130                 client.publish(MQTT_TOPIC_ALARM, alarm_payload) #Відправка повідомлення на брокер
131                 print('Alarm triggered!') #Виведення повідомлення про активацію сигналізації
132             elif smoke_level <= ALARM_THRESHOLD and alarm_triggered: #Якщо рівень диму не перевищує порогове значення
133                 buzzer_pin.off() #Вимкнення зумеру
134                 alarm_triggered = False #Вимкнення флагу активності тривоги
135
136             #Оновлення світлодіодної стрічки
137             update_leds() #Оновлення світлодіодної стрічки
138
139             #Пауза перед наступною відправкою
140             time.sleep(60) #Затримка 1хв
141         except OSError as e:
142             print('Failed to read sensor.') #Виведення повідомлення про помилку
143             time.sleep(2) #Затримка 2 секунди

```

Рисунок 2.21 - Лістинг основної функції

Нарешті важливо переконаватися, що основна функція запускається при старті пристрою, щоб код автоматично починав роботу після перезавантаження.

2.7 Розробка потоків та інтерфейсів в Node-RED

Node-RED – це потокове середовище розробки, створене для підключення програмного забезпечення та апаратного забезпечення до Інтернету речей. Він пропонує простий графічний інтерфейс, за допомогою якого користувачі можуть створювати та змінювати потоки даних, перетягуючи блоки або вузли та з'єднуючи їх лініями. Node-RED Працює з багатьма протоколами та пристроями, що робить його універсальною платформою для інтеграції різних компонентів у системи Інтернету речей.

Node-RED Dashboard – це розширення Node-RED, яке дозволяє створювати інтерфейси користувача для моніторингу та керування пристроями системи. Він може використовуватися для створення візуальних інформаційних панелей, які включають графіки, кнопки, індикатори та інші

						КНУ КРБ.151.24.02.02.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			55

елементи керування, які відображають дані в реальному часі. В цілому, це чудовий інструмент для візуалізації даних і керування системами, а його інтерфейс для кінцевих користувачів простий та зрозумілий.

З кількох причин використання цих двох рішень є відмінним вибором для роботи з протоколом MQTT. По-перше, підтримка протоколу MQTT «з коробки», полегшує інтеграцію протоколу. По-друге, налаштування потоків даних, підключення та обробка повідомлень MQTT є простими завдяки візуальному інтерфейсу Node-RED, що значно спрощує та прискорює розробку та налагодження систем. По-третє, Node-RED Dashboard пропонує інструменти для створення користувацьких інтерфейсів, які дуже легко та зрозуміло налаштовуються, та надають користувачам зрозумілі візуальні дані, якими, за протоколом MQTT, можна не тільки слідкувати, а й керувати. Таким чином, поєднання Node-Red та Node-Red Dashboard надає потужне та гнучке рішення для розробки, інтеграції та керування спроектованою системою.

На Рисунок 2.22 зображений алгоритм взаємодії середовища Node-RED з брокером.



Рисунок 2.22 - Блок-схема взаємодії Node-RED та MQTT-брокера

2.7.1 Створення потоку «Моніторинг температури та вологості»

Даний потік повинен отримувати дані про температуру та вологість з

MQTT-брокера, після чого обробляти ці дані, відсортовувати їх, тільки після чого відображати їх в інтерфейсі користувача. Також, потік мусить записувати дані до бази даних, після чого візуалізувати їх на графіках для забезпечення наочності та аналізу тренду зміни показань датчика вологості та температури.

На Рисунок 2.23 зображений алгоритм роботи потоку «Моніторинг температури та вологості».



Рисунок 2.23 - Блок-схема роботи потоку "Моніторинг температури та вологості"

Перед створенням потоку було прийнято рішення встановити на брокер базу даних для запису значень температури і вологості та виведення цих даних у вигляді графіку. Для цієї задачі була обрана InfluxDB.

Для її встановлення в терміналі Raspberry необхідно виконати наступну команду: «sudo apt install influxdb». По завершенню встановлення програми треба запустити InfluxDB командами «sudo systemctl unmask/start/status influxdb».

```
pi@raspberrypi: ~
Processing triggers for man-db (2.9.4-2) ...
pi@raspberrypi:~ $ sudo systemctl unmask influxdb
pi@raspberrypi:~ $ sudo systemctl start influxdb
pi@raspberrypi:~ $ sudo systemctl enable influxdb
Synchronizing state of influxdb.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable influxdb
pi@raspberrypi:~ $ sudo systemctl status influxdb
● influxdb.service - InfluxDB is an open-source, distributed, time series database
   Loaded: loaded (/lib/systemd/system/influxdb.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-05-28 18:24:15 EEST; lmin 32s ago
     Docs: man:influxd(1)
  Main PID: 2488 (influxd)
    Tasks: 9 (limit: 1595)
      CPU: 232ms
   CGroup: /system.slice/influxdb.service
           └─2488 /usr/bin/influxd -config /etc/influxdb/influxdb.conf
```

Рисунок 2.24 - Активація служби автозапуску InfluxDB

Після установки та активації служби автозапуску треба виконати наступні кроки для створення бази даних, до котрої будуть надходити показання температури та вологості:

- Відкрити термінал та запустити InfluxDb командою «influx»;
- Створити базу даних командою «CREATE DATABASE dht11»;
- Обрати базу даних для роботи командою «USE dht11»;
- Створити вимір «dht11» з полями «temperature» та «humidity», записавши текстові дані до бази даних командою «INSERT dht11 temperature=25,humidity=50»;

Далі можна буде перевірити записані дані, виконав запит «SELECT * FROM dht11».

Тепер можна приступати до створення самого потоку в Node-RED. Спочатку треба встановити відповідний пакет, який додає ноди для взаємодією з InfluxDB.

Створений потік для моніторингу температури та вологості наведений на Рисунок 2.25.

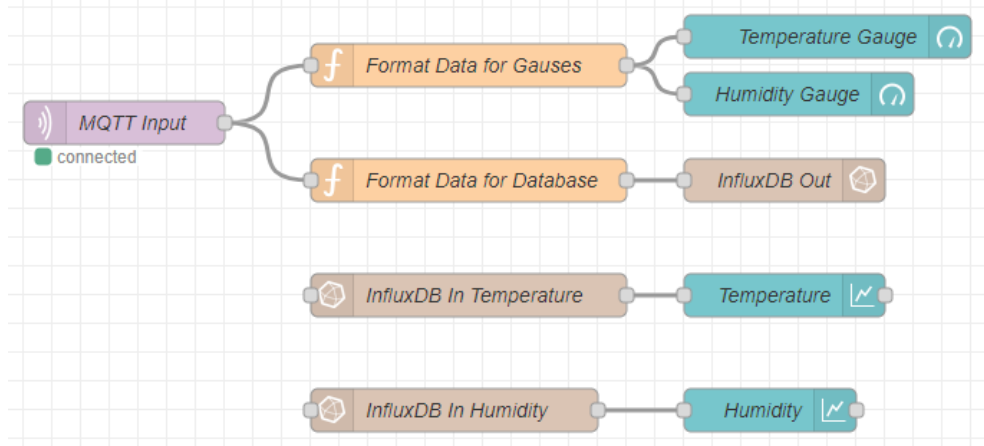


Рисунок 2.25 - Вигляд потоку моніторингу температури та вологості в середовищі Node-RED

У потоці Node-RED для моніторингу температури та вологості дані з датчика DHT11 надходять через нод «mqtt_in». Ці дані обробляються двома «function» нодами. Перший нод розбиває отримані параметри на температуру та вологість та відправляє їх на відповідні елементи «gauge» для відображення. Другий нод розбиває дані для запису в базу даних і відправляє їх через «influxdb out». У потоці також є два елементи «influxdb in», які підключені до нодів графіків. Ці графіки отримують дані з бази даних та візуалізують зміни температури та вологості.

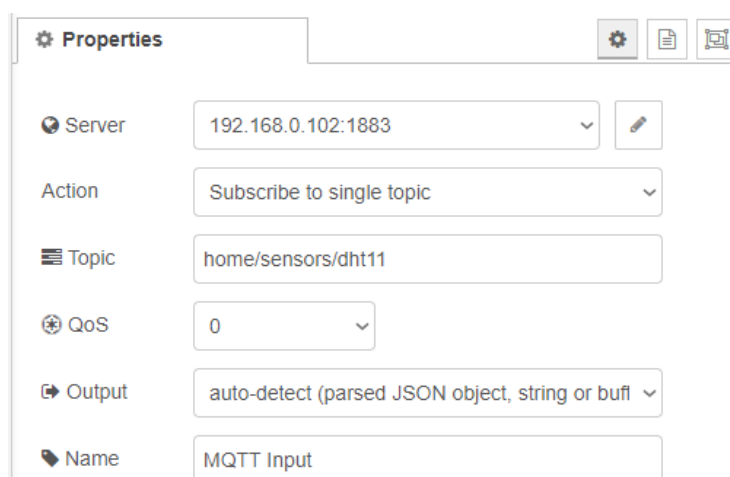


Рисунок 2.26 - Налаштування вузла MQTT Input

```

1  var temperature = msg.payload.temperature;
2  var humidity = msg.payload.humidity;
3
4  msg.payload = {
5      temperature: temperature,
6      humidity: humidity
7  };
8
9  return msg;

```

Рисунок 2.27 - Код блоку function "Format Data for Gauges"

Наведений код отримує значення температури та вологості з повідомлення, перетворює їх на об'єкт і повертає для відображення на елементах «gauge».

```

1  var temperature = msg.payload.temperature;
2  var humidity = msg.payload.humidity;
3
4  var temperatureQuery = "INSERT INTO dht11 (temperature) VALUE
5  node.send({payload: temperatureQuery});
6
7  var humidityQuery = "INSERT INTO dht11 (humidity) VALUES (" +
8  node.send({payload: humidityQuery});
9
10 return msg;

```

Рисунок 2.28 - Код нода function "Format Data for Database"

Цей код отримує значення температури та вологості з повідомлення, формує два окремі SQL-запити для вставки даних у базу даних і відправляє їх. Ці запити записують значення температури і вологості вимірювання dht11 в базі даних InfluxDB.

2.7.2 Створення потоку «Управління світлодіодною стрічкою»

Потік «Керування світлодіодною стрічкою» в Node-RED дозволяє користувачеві налаштувати яскравість, колір та увімкнення/вимкнення світлодіодної стрічки через інтерфейс користувача. Процес починається з отримання даних від користувача через інтерфейс, де він визначає параметри світлодіодної стрічки. Ці дані включають рівень яскравості, вибраний колір

та стан (увімкнено або вимкнено). Потім дані обробляються, щоб перевірити їхню коректність і перетворити на формат, придатний для відправки. Після обробки встановлені параметри відправляються на MQTT-брокер, який передає команди світлодіодної стрічки. Таким чином, потік управляє світлодіодною стрічкою, забезпечуючи можливість гнучкого контролю її параметрів через зручний інтерфейс.



Рисунок 2.29 - Блок-схема роботи потоку "Управління світлодіодною стрічкою"

Потік «Управління світлодіодною стрічкою» складається з декількох елементів інтерфейсу користувача, які дозволяють вибирати колір, яскравість і стан світлодіодної стрічки. Для вибору кольору використовуються елементи «colour picker» та 16 кнопок, кожна з яких задає параметри популярних кольорів. Елемент «colour picker» призначений для вибору незвичного відтінку. Параметри яскравості задаються за допомогою нода «slider», а увімкнення/вимкнення стрічки здійснюється через нод «switch».

У потоці встановлено чотири ноди «change» для зміни топиків, на які будуть надсилатися відповідні параметри. Перший «change node» підключений до «colour picker» і змінює топик цього елемента на «home/led/colour». Другий «change node» підключений до «slider» і змінює

топiк цього елемента на «home/led/brightness». Третiй «change node» пiдключений до «switch» i змiнює топiк цього елемента на «home/led/power». Четвертий «change node» пiдключений до всiх 16 кнопок та змiнює iх топiк на «home/led/colour».

Усi чотири «change node» пiдключенi до «mqtt_out», який надсилає необхіднi параметри на MQTT-брокер. Таким чином, змiни в iнтерфейсi користувача передаються на брокер, який управляє свiтлодiдною стрiчкою, забезпечуючи її коректне вклучення, регулювання яскравостi i змiна кольору.

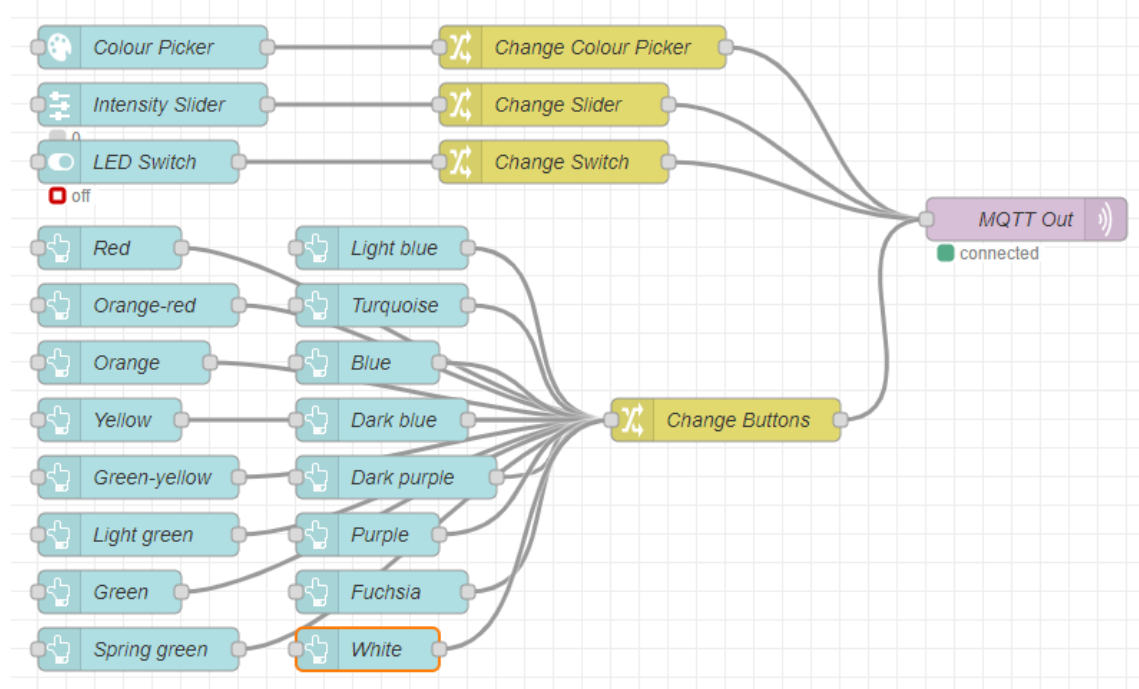


Рисунок 2.30 - Загальний вигляд потоку для управління свiтлодiдною стрiчкою в середовищi Node-RED

2.7.3 Створення потоку «Сигналізація»

Потiк «Сигналізація» починається з отримання повідомлення вiд MQTT-брокера про подiю, яка потребує уваги користувача. Потiм відбувається перевiрка отриманого повідомлення щодо необхідностi прийняття подальших дiй. Якщо подiя вимагає реакцiї, формується вiдповiдне повідомлення для користувача. Повiдомлення надсилається до Node-RED для обробки. Пiсля цього повідомлення вiдправляється на

Telegram-бота для повідомлення користувача про подію, що відбулася.

Таким чином, потік «Сигналізація» автоматизує процес моніторингу та сповіщення про події, дозволяючи користувачам оперативно реагувати на важливі ситуації.



Рисунок 2.31 - Блок-схема роботи потоку "Сигналізація"

Для створення телеграм-бота в цій системі було використано Telegram API. Спочатку було створено бот через Telegram BotFather - офіційного бота Telegram для створення та налаштування нових ботів. BotFather надав унікальний токен, який ідентифікує наш бот.

Потім було встановлено розширення для Node-RED, яке додає до нього ноди для взаємодії з Telegram. Це розширення забезпечує зручний спосіб інтеграції Telegram у процеси автоматизації, що надаються Node-RED. Після встановлення розширення в Node-RED з'явилися нові ноди, такі як «telegram sender» і «telegram receiver», які дозволяють відправляти і приймати повідомлення через Telegram-бота.

Потік «Сигналізація» починається з елемента «mqtt_in», який отримує оповіщення про виявлення диму MQTT-брокера. Отримане повідомлення

потім направляється в нод «change» для формування зрозумілого користувача повідомлення. Нод «change» виконує необхідну обробку даних, щоб подати інформацію про виявлений дим у зрозумілому вигляді.

Вихід нода «change» підключений до двох нодів: «text» та «telegram sender». Нод «text» використовується для формування текстового повідомлення, яке буде передано користувачеві. Нод «telegram sender» відповідає за відправку сформованого повідомлення на Telegram-бота, який повідомить користувача про виявлений дим.

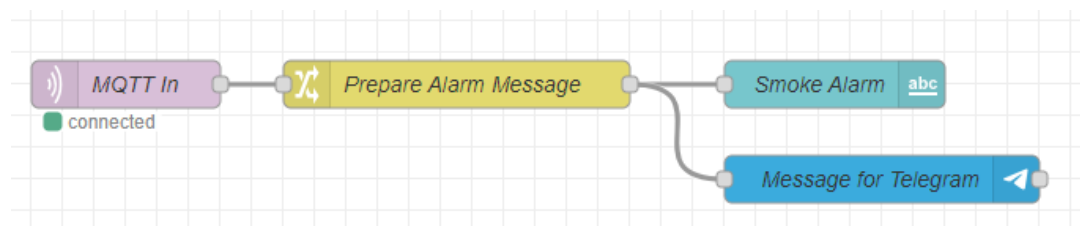


Рисунок 2.32 - Загальний вигляд потоку сигналізації в середовищі Node-RED

Після створення всіх потоків та налаштування нодів, можна спробувати запуснути користувацький веб-інтерфейс, вбивши в строку браузера «<http://192.168.0.102:1880/ui/>».

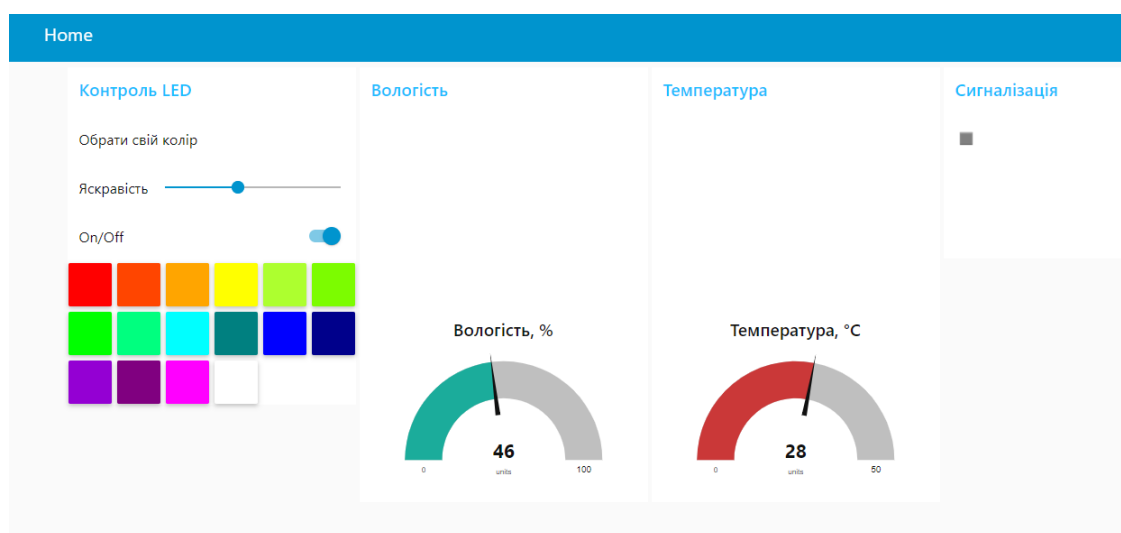


Рисунок 2.33 - Вид створеного користувацького інтерфейсу

Висновок до розділу 2

В результаті проведеної роботи з проектування та реалізації апаратно-програмної частини системи управління розумним будинком, було створено інтегровану систему, що поєднує різні пристрої та технології. Використання Raspberry Pi як MQTT-брокера забезпечило надійну та ефективну передачу даних між компонентами системи. Програма, написана для NodeMCU ESP8266 на MicroPython, успішно керує датчиками та виконавчими пристроями, дозволяючи здійснювати моніторинг та керування в реальному часі.

Інтерфейс, реалізований в Node-RED, забезпечує зручний та інтуїтивно зрозумілий спосіб взаємодії користувача із системою розумного будинку. Він дозволяє легко налаштовувати параметри, отримувати дані від датчиків та керувати пристроями через веб-інтерфейс.

Досягнуті результати підтверджують ефективність обраного підходу та демонструють високий рівень інтеграції апаратних та програмних компонентів. Створена система відповідає сучасним вимогам до функціональності, безпеки та зручності використання, що робить її перспективним рішенням для подальшого розвитку та впровадження у розумні будинки.

					<i>КНУ КРБ.151.24.02.02.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

ВИСНОВКИ

У ході виконання дипломного проекту "Автоматизація систем контролю та управління розумним будинком з розробкою програмного забезпечення" було проведено комплексну роботу, що включає аналіз предметної галузі, проектування та реалізацію апаратно-програмної частини системи.

На етапі аналізу предметної галузі та існуючих рішень були вивчені сучасні технології та підходи до створення розумних будинків, що дозволило визначити найкращі практики та потенційні напрямки для покращень. В результаті була обрана оптимальна архітектура системи, що включає використання Raspberry Pi як MQTT-брокера і NodeMCU ESP8266 для управління датчиками та виконавчими пристроями.

У розділі проектування та реалізації апаратно-програмної частини було розроблено інтегровану систему, що забезпечує надійну та ефективну роботу всіх компонентів. Програма, написана на MicroPython для NodeMCU ESP8266, успішно справляється із завданнями моніторингу та управління, а інтерфейс, реалізований у Node-RED, надає користувачеві інтуїтивно зрозумілий спосіб взаємодії із системою розумного будинку.

Створена система має високий ступінь інтеграції та функціональності, забезпечуючи комфорт, безпеку та енергоефективність у повсякденному житті. Вона дозволяє легко налаштовувати параметри, отримувати дані від датчиків та керувати пристроями через зручний веб-інтерфейс. Використання сучасних технологій, таких як інтернет речей (IoT), забезпечило гнучкість та масштабованість системи, роблячи її перспективним рішенням для подальшого розвитку.

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Рожкова Е.О.</i>			<i>ВИСНОВКИ</i>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Тиханський М.П.</i>						
<i>Н. Контр.</i>		<i>Маринич І.А.</i>			<i>КНУ АКІТ-21ск</i>			
<i>Затвердив</i>		<i>Тронь В.В.</i>						

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Розумний дім. Kronzen: веб-сайт. URL: <https://www.khomzen.com> (Дата звернення: 10.04.2024)
2. Smart Home Concept – an overview. Sciencedirect: веб-сайт. URL: <https://www.sciencedirect.com/topics/engineering/smart-home-concept> (Дата звернення: 10.04.2024)
3. (PDF) The Smart Home Concept: our immediate future. ResearchGate: веб-сайт. URL: https://www.researchgate.net/publication/224696459_The_Smart_Home_Concept_our_immediate_future (Дата звернення: 15.04.2024)
4. What are the Types of Smart Home Automation Systems? Smart Home Works: веб-сайт. URL: <https://smarthomeworks.com.au/2022/11/03/types-home-automation-systems> (Дата звернення: 15.04.2024)
5. Top 7 Smart Home Protocols Compared. Lembergsolutions: веб-сайт. URL: <https://lebergsolutions.com/blog/smart-home-protocols-explained> (Дата звернення: 15.04.2024)
6. Система управління інфраструктурою будинку «розумний дім». пат. 81054 Україна: МПК G08B19/00, №u201212028, заявл. 19.10.2012; опубл. 25.06.2013, Бюл № 12/2013
7. Бездротовий інтернет речей, клімат-контроль та система розумний дім. пат. US20240121133 США: МПК H04L 12/28, G06N 20/00, G10L 15/22, G10L 15/30, №18133474, заявл. 11.04.2023; опубл. 11.04.2024
8. Система та спосіб налаштування мовливних інтерфейсів розумного будинку за використанням персоналізованих профілів мовлення. пат. US201662359005P США: МПК G10L15/22, G10L15/07, № 23178996.7, заявл. 28.06.2017; опубл. 04.10.2023, Бюл. №35/2023

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>		
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розроб.</i>		<i>Рожкова Е.О.</i>			<i>Лім.</i>	<i>Арк.</i>	<i>Архуів</i>
<i>Перевір.</i>		<i>Тиханський М.П.</i>					
<i>Н. Контр.</i>		<i>Маринич І.А.</i>			<i>СПИСОК ЛІТЕРАТУРИ</i>		
<i>Затвердив</i>		<i>Тронь В.В.</i>			<i>КНУ АКІТ-21ск</i>		

9. Система розумний будинок та управління нею. пат. EP20210150969 Нідерланди: МПК G G08B15/00, G05B15/02, № 21162963.9, заявл. 16.03.2021; опубл. 13.07.2022, Бюл № 28/2022
10. What is a Raspberry Pi? Opensource: веб-сайт. URL: <https://opensource.com/resources/raspberry-pi> (Дата звернення: 25.04.2024)
11. Getting Started With the Raspberry Pi 3B+ - Little Bird Guides. Little Bird Electronics: веб-сайт. URL: <https://learn.littlebirdelectronics.com.au/raspberry-pi/getting-started-with-the-raspberry-pi-3b> (Дата звернення: 25.04.2024)
12. Raspberry Pi. Wikipedia: веб-сайт. URL: https://ru.wikipedia.org/wiki/Raspberry_Pi (Дата звернення: 25.04.2024)
13. Raspberry Pi для розумного дому (Smart Home). evo net: веб-сайт. URL: <https://evo.net.ua/ru/raspberry-pi-dlya-avtomatizatsii-umnogo-doma-smart-home/> (Дата звернення: 25.04.2024)
14. Advanced Home Automation using Raspberry Pi 3| Udemy. Udemy: веб-сайт. URL: <https://www.udemy.com/course/advanced-home-automation-using-raspberry-pi-3/> (Дата звернення: 26.04.2024)
15. The Evolution of Smart Home Technology. Medium: веб-сайт. URL: <https://tcaflisch.medium.com/the-evolution-of-smart-home-technology-c3b3f918fc8c> (Дата звернення: 27.04.2024)
16. Fibaro Home Center 3 Lite. hellosmarthome: веб-сайт. URL: <https://www.hellosmarthome.nl/post/review-fibaro-home-center-3-lite/> (Дата звернення: 29.04.2024)
17. Home Assistant 101. Посібник для початківців. DOU: веб-сайт. URL: <https://dou.ua/forums/topic/38947/> (Дата звернення: 29.04.2024)
18. X10: The Revolutionary Smart Home Device You've Never Heard Of. Medium: веб-сайт URL: <https://medium.com/digitalshroud/x10-the-revolutionary-smart-home-device-youve-never-heard-of-48790e272e1f> (Дата звернення: 29.04.2024)
19. Loxone – новатор домашньої автоматизації – системи опалення viessman.

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

- viessman: веб-сайт. URL: http://viessmann.com.ua/novost-207/Loxone_-_novator_domashney_avtomatizacii.html (Дата звернення: 29.04.2024)
20. What is the best smart home system? | Control4, KNX, Savant. Acoustic Pixel: веб-сайт. URL: <https://www.acousticpixel.co.uk/inspiration/best-smart-home-system/> (Дата звернення: 29.04.2024)
21. Чому MQTT є важливою частиною IoT? Fiberroad: веб-сайт. URL: <https://fiberroad.com/uk/why-is-mqtt-an-important-part-of-iot/> (Дата звернення: 02.05.2024)
22. WiFi Плата NodeMCU V2 ESP8266 (CP2102). Arduino: веб-сайт. URL: <https://arduino.ua/ru/prod1495-wifi-plata-nodemcu-v2-esp8266-cp2102> (Дата звернення: 06.05.2024)
23. DHT10/DHT11/DHT12/DHT21/DHT22 - Digital-Output Relative Humidity & Temperature Sensor Module. NET nanoFramework: веб-сайт. URL: <https://docs.nanoframework.net/devicesdetails/Dhtxx/README.html> (Дата звернення: 06.05.2024)
24. Інструкція по роботі з макетною платою. Robx: веб-сайт. URL: <https://robx.org/wiki/elektronika/prototip/maketnaya-plata/> (Дата звернення: 06.05.2024)
25. Active buzzer module for Arduino buy Kiev Ukraine. greenchip: веб-сайт. URL: <https://greenchip.com.ua/23-0-481-2.html> (Дата звернення: 06.05.2024)
26. Модуль датчика газу MQ-2 – бутан, пропан, метан, спирт, водень, сигаретний дим. 3V3: веб-сайт. URL: https://3v3.com.ua/product_7531.html (Дата звернення: 06.05.2024)
27. Blynk Controlled WS2812B Neopixel LED Strip with NodeMCU. How To Electronics: веб-сайт. URL: <https://how2electronics.com/ws2812b-neopixel-led-strip-nodemcu/> (Дата звернення: 11.05.2024)
28. Повне керівництво по адресній світлодіодній стрічці. Ledyi: веб-сайт. URL: https://www.ledyilighting.com/ru/the-ultimate-guide-to-addressable-led-strip/#WS2811_Vs_WS2812_Vs_WS2813
29. MQTT broker with Mosquitto and Node-RED. GitHub: веб-сайт. URL:

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

- <https://github.com/mkgeiger/mqtt-broker> (Дата звернення: 19.05.2024)
30. ESP8266 - Вступ. Модуль Wemos D1 Mini. ITMaster: веб-сайт. URL: <https://itmaster.biz.ua/electronics/esp8266/esp8266-intro.html> (Дата звернення: 21.05.2024)
31. Documentation – Node-RED. Node-RED: веб-сайт. URL: <https://nodered.org/docs/> (Дата звернення: 24.05.2024)
32. MicroPython: ESP32/ESP8266 with DHT11/DHT22 Temperature and Humidity Sensor. Random Nerd Tutorias: веб-сайт. URL: <https://randomnerdtutorials.com/esp32-esp8266-dht11-dht22-micropython-temperature-humidity-sensor/> (Дата звернення: 25.05.2024)
33. Методичні рекомендації до виконання кваліфікаційної роботи бакалавра для студентів спеціальності 151 – «Автоматизація та комп'ютерно-інтегровані технології» усіх форм навчання / [Укладачі: Моркун Н.В., Маринич І.А.]. – Кривий Ріг: Видавничий центр «КНУ», 2019, 50с.
34. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура і правила оформлення. Київ, ДП «УкрННЦ», 2015. 26с. (Інформація та документація).
35. ДСТУ 8302:2015. Бібліографічне посилання. Загальні вимоги та правила складання Київ, ДП «УкрННЦ», 2016. 16 с. (Інформація та документація).
36. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень в українській мові. Загальні вимоги та правила. Київ, ДП «УкрННЦ», 2013. 23 с. (Інформація та документація).
37. ДСТУ 3651.0-97 Метрологія. Одиниці фізичних величин. Основні одиниці фізичних величин Міжнародної системи одиниць. Основні положення, назви та позначення Київ, Держстандарт України, 1998. 27 с. (Інформація та документація).

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

Код для NodeMCU ESP8266 CP2102

```

import time #Імпортування модулю time для роботи з затримками
import machine #Імпортування модулю machine для роботи з апаратними інтерфейсами ESP
import dht #Імпортування модулю dht для роботи з датчиком DHT11
import network #Імпортування модулю network для роботи з Wi-Fi
import neopixel #Імпортування модулю neopixel для роботи зі світлодіодною стрічкою WS2812
from umqtt.simple import MQTTClient #Імпортування MQTTClient із umqtt.simple для роботи з MQTT

#Параметри підключення к Wi-Fi
SSID = 'morojenko' #Назва Wi-Fi мережі
PASSWORD = r'morojenko))' #Пароль від Wi-Fi мережі

#Параметри підключення к MQTT брокеру
MQTT_BROKER = '192.168.0.102' #IP-адреса MQTT брокера
MQTT_PORT = 1883 #Порт для підключення до MQTT брокеру
MQTT_TOPIC_DHT11 = 'home/sensors/dht11' #Топік для відправки даних з DHT11
MQTT_TOPIC_ALARM = 'home/sensors/alarm' #Топік для відправки даних об активації сигналізації
MQTT_TOPIC_LED_BRIGHTNESS = 'home/led/brightness' #Топік для управління яскравістю
світлодіодної стрічки
MQTT_TOPIC_LED_COLOUR = 'home/led/colour' #Топік для управління кольором світлодіодної
стрічки
MQTT_TOPIC_LED_POWER = 'home/led/power' #Топік для вмикання/вимикання світлодіодної
стрічки

#Налаштування підключення до Wi-Fi
def connect_wifi():
    wifi = network.WLAN(network.STA_IF) #Створення об'єкта WLAN в режимі Station
    wifi.active(True) #Активація Wi-Fi
    wifi.connect(SSID, PASSWORD) #Підключення до Wi-Fi мережі
    while not wifi.isconnected(): #Очікування підключення до мережі
        pass
    print('Connected to WiFi:', wifi.ifconfig()) #Виведення інтформації о підключенні

```

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

```

#Налаштування MQTT клієнта
def connect_mqtt():
    client = MQTTClient('esp8266', MQTT_BROKER, port=MQTT_PORT) #Створення об'єкта MQTTClient
    client.set_callback(mqtt_callback) #Встановлення функції обратного зв'язку
    client.connect() #Підключення до MQTT брокеру
    client.subscribe(MQTT_TOPIC_LED_BRIGHTNESS) #Підписка на топик управління яскравістю
    client.subscribe(MQTT_TOPIC_LED_COLOUR) #Підписка на топик управління кольором
    client.subscribe(MQTT_TOPIC_LED_POWER) #Підписка на топик управління
    #вмиканням/вимиканням
    print('Connected to MQTT broker') #Виведення повідомлення про вдале підключення
    return client #Повернення об'єкта клієнту

#Ініціалізація датчика DHT11
dht_pin = machine.Pin(2) #Підключення датчика до піна D4 (GPIO2)
dht_sensor = dht.DHT11(dht_pin) #Створення об'єкта DHT11 з вказаним піном

#Ініціалізація датчика диму MQ-2 та зумера YL-44
mq2_pin = machine.ADC(0) #Підключення датчика до аналогового входу ADC0
buzzer_pin = machine.Pin(5, machine.Pin.OUT) #Підключення зумера до піна D1 (GPIO5)

#Ініціалізація світлодіодної стрічки WS2812
nr_pin = machine.Pin(4) #Підключення стрічки до піна D2 (GPIO4)
num_leds = 23 #Кількість світлодіодів у стрічці
nr = neopixel.NeoPixel(nr_pin, num_leds) #Створення об'єкта NeoPixel з вказаним піном та
#кількістю світлодіодів

#Встановлення порогового значення для спрацьовування сигналізації
ALARM_THRESHOLD = 300 #Порогове значення для рівня диму

#Змінні для управління світлодіодною стрічкою
led_on = False #Флаг включення світлодіодної стрічки
led_color = (0, 0, 0) #Колір світлодіодної стрічки (RGB)
led_brightness = 255 #Яскравість світлодіодної стрічки, значення від 0 до 255

# Ініціалізація змінної для сигналізації
alarm_triggered = False

```

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72


```

#Зворотній виклик для обробки повідомлень MQTT
def mqtt_callback(topic, msg):
    global led_on, led_color, led_brightness #Об'явлення глобальної змінної
    if topic == MQTT_TOPIC_LED_POWER: #Перевірка повідомлення топіку
        if msg == b'ON': #Якщо повідомлення "ON"
            led_on = True #Вмикання світлодіодної стрічки
        elif msg == b'OFF': #Якщо повідомлення "OFF"
            led_on = False #Вимикання світлодіодної стрічки
    elif topic == MQTT_TOPIC_LED_BRIGHTNESS: #Якщо топік для управління яскравістю
        try:
            led_brightness = int(msg.decode('utf-8')) #Отримання значення яскравості
            if led_brightness < 0:
                led_brightness = 0
            elif led_brightness > 255:
                led_brightness = 255
        except ValueError:
            print("Invalid brightness value") #Виведення повідомлення про помилку
    elif topic == MQTT_TOPIC_LED_COLOUR: #Якщо топік для управління кольором
        try:
            parts = msg.decode('utf-8').split(',') #Розбивання повідомлення на частини
            r = int(parts[0]) #Отримання значення червоного кольору
            g = int(parts[1]) #Отримання значення зеленого кольору
            b = int(parts[2]) #Отримання значення блакитного кольору
            led_color = (r, g, b) #Встановлення кольору
        except:
            print("Invalid color value") #Виведення повідомлення про помилку

#Функція для оновлення стану світлодіодної стрічки
def update_leds():
    global led_on, led_color, led_brightness #Об'явлення глобальної змінної
    if led_on: #Якщо світлодіодна стрічка ввімкнена
        for i in range(num_leds): #Проходження по всім світлодіодам
            np[i] = (led_color[0] * led_brightness // 255, #Встановлення кольору з урахуванням яскравості
                    led_color[1] * led_brightness // 255,
                    led_color[2] * led_brightness // 255)

```

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

```

else: #Якщо світлодіодна стрічка вимкнена
    for i in range(num_leds): #Проходження по всім світлодіодам
        np[i] = (0, 0, 0) #Вимикання світлодіодів
    np.write() #Застосування значень

#Основна функція
def main():
    time.sleep(5) #Затримка для стабілізації живлення
    connect_wifi() #Підключення до Wi-Fi
    client = connect_mqtt() #Підключення до MQTT

    while True:
        try:
            #Зчитування даних з DHT11
            dht_sensor.measure() #Запуск виміру
            temp = dht_sensor.temperature() #Зчитування температури
            hum = dht_sensor.humidity() #Зчитування вологості
            print('Temperature:', temp, 'C') #Виведення температури
            print('Humidity:', hum, '%') #Виведення вологості

            #Формування повідомлення для відправки
            payload = '{ "temperature": %d, "humidity": %d }' % (temp, hum) #Формування строки з даними
            client.publish(MQTT_TOPIC_DHT11, payload) #Відправлення даних на брокер
            print('Data sent to MQTT broker') #Виведення повідомлення про відправку

            #Зчитування даних з MQ-2
            smoke_level = mq2_pin.read() #Зчитування рівня диму
            print('Smoke level:', smoke_level) #Виведення рівня диму

            #Перевірка рівня диму з пороговим значенням
            if smoke_level > ALARM_THRESHOLD: and not alarm_triggered: #Якщо рівень диму перевищує
                порогове значення
                    buzzer_pin.on() #Ввімкнення зумеру
                    alarm_triggered = True #Встановлення флагу активності тривоги
                    alarm_payload = '{ "alarm": "Smoke detected!" }' #Формування повідомлення про активацію
                сигналізації

```

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

```

client.publish(MQTT_TOPIC_ALARM, alarm_payload) #Відправка повідомлення на брокер
print('Alarm triggered!') #Виведення повідомлення про активацію сигналізації
elif smoke_level <= ALARM_THRESHOLD and alarm_triggered: #Якщо рівень диму не
перевищує порогове значення
    buzzer_pin.off() #Вимкнення зумеру
    alarm_triggered = False #Вимикання флагу активності тривоги

#Оновлення світлодіодної стрічки
update_leds() #Оновлення світлодіодної стрічки

#Пауза перед наступною відправкою
time.sleep(60) #Затримка 1хв
except OSError as e:
    print('Failed to read sensor.') #Виведення повідомлення про помилку
    time.sleep(2) #Затримка 2 секунди

if __name__ == '__main__':
    main() #Запуск основної функції

```

					<i>КНУ КРБ.151.24.02.00.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75