

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДОМАШНЬОЇ АВТОМАТИЗАЦІЇ
ДЛЯ БЕЗДРОТОВОЇ МЕРЕЖІ СЕНСОРІВ ТА АКТУАТОРІВ
HOME AUTOMATION SOFTWARE FOR WIRELESS NETWORK
OF SENSORS AND ACTUATORS

Виконав: студент 4 курсу, групи КНТ-129

Спеціальності 121 Інженерія програмного

забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

АБАЛЬМАС А.О.

(ПРИЗВИЩЕ та ініціали)

Керівник ПАРХОМЕНКО А.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛЕНЬОВА І.Я.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук та технологій

Кафедра програмних засобів

Ступінь вищої освіти бакалавр

Спеціальність 121 Інженерія програмного забезпечення

(код і найменування)

Освітня програма (спеціалізація) Інженерія програмного забезпечення

(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.

Сергій СУББОТІН

“ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

АБАЛЬМАСА Антона Олеговича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення домашньої автоматизації для бездротової мережі сенсорів та актуаторів. Home Automation Software for Wireless Network of Sensors and Actuators.

Керівник проєкту (роботи) к.т.н., доцент, ПАРХОМЕНКО Анжеліка Володимирівна

(ПРИЗВИЩЕ, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “ 4 ” квітня 2023 року № 87

2. Строк подання студентом проєкту (роботи) 1 червня 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Дослідження структурних та функціональних особливостей бездротових мереж сенсорів та актуаторів. 3. Проєктування програмно-апаратного забезпечення БМСА. 4. Конструювання програмно-апаратного забезпечення БМСА.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ПАРХОМЕНКО А.В., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “17” квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання
2	Аналіз предметної області	2 тиждень	Розділ 1
3	Дослідження особливостей БМСА	3 тиждень	Розділ 2
4	Розробка архітектури системи	4 тиждень	Розділ 3
5	Розробка програмно-апаратного забезпечення	5 тиждень	Розділ 4
6	Розгортання БМСА на СДА	6 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	7 тиждень	ПЗ, Додатки
8	Захист роботи	8 тиждень	

Студент

_____ Антон АБАЛЬМАС
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

_____ Анжеліка ПАРХОМЕНКО
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 93 с., 13 табл., 31 рис., 2 дод., 72 джерел.

БЕЗДРОТОВА МЕРЕЖА СЕНСОРІВ ТА АКТУАТОРІВ, СИСТЕМА ДОМАШНЬОЇ АВТОМАТИЗАЦІЇ, WI-FI МОДУЛЬ, ПРОГРАМНО-АПАРATНЕ РІШЕННЯ, ПРОТОКОЛ ПЕРЕДАЧІ ДАНИХ.

Об'єкт дослідження – процес збору, обробки та передачі інформації в бездротовій мережі сенсорів та актуаторів.

Предмет дослідження – методи та засоби розробки програмного та апаратного забезпечення бездротових мереж сенсорів та актуаторів (БМСА) для систем домашньої автоматизації (СДА).

Мета роботи – дослідження та реалізація масштабованого, доступного програмно-апаратного рішення для БМСА, що забезпечить ефективне функціонування СДА.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, середовище програмування Arduino IDE, мова програмування C++, платформи домашньої автоматизації OpenHAB та Domoticz, Wi-Fi модуль ESP8266, датчики DS18B20, DHT11, PIR, реле.

Результати. Розроблено апаратне та програмне забезпечення для БМСА, що дозволяє контролювати показники середовища та формувати керуючі впливи на обладнання СДА.

Висновки. Сучасні СДА можуть базуватися на реалізованій БМСА для ефективного розв'язку завдань домашньої автоматизації завдяки розробленим програмним рішенням, використанню недорогих компонентів, а також можливостям масштабування.

Галузь використання – системи домашньої, офісної та промислової автоматизації.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 93 pages, 13 tables, 31 figures, 2 appendixes, 72 sources.

WIRELESS NETWORK OF SENSORS AND ACTUATORS, HOME AUTOMATION SYSTEM, WI-FI MODULE, SOFTWARE AND HARDWARE SOLUTION, DATA TRANSMISSION PROTOCOL.

Object of study is the process of collecting, processing and transmitting information in a wireless network of sensors and actuators.

Subjects of study are methods and tools of development of software and hardware of wireless networks of sensors and actuators (WNSA) for home automation systems (HAS).

The goal of the work is research and implementation of a scalable, affordable software and hardware solution for the , which will ensure the effective functioning of the HAS.

Materials, methods and technical tools: structural and object-oriented programming, Arduino IDE programming environment, C++ programming language, OpenHAB and Domoticz home automation platforms, Wi-Fi module ESP8266, sensors DS18B20, DHT11, PIR, relays.

The results. Hardware and software for WNSA have been developed, which allows monitoring of environmental indicators and forming control influences on HAS equipment.

Conclusions. Modern HAS can be based on implemented WNSA for effective solution of home automation tasks due to developed software solutions, usage of inexpensive components, as well as scaling capabilities.

The fields of application are home, office and industrial automation systems.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Огляд технологій домашньої автоматизації	10
1.2 Дослідження особливостей існуючих СДА	12
1.3 Постановка завдань дипломної роботи.....	20
2 Дослідження структурних та функціональних особливостей бездротових мереж сенсорів та актуаторів.....	21
2.1 Структура та галузі використання БМСА.....	21
2.2 Особливості БМСА.....	23
2.3 Дослідження бездротових протоколів для БМСА.....	24
2.4 Висновки до розділу 2.....	28
3 Проектування програмно-апаратного забезпечення БМСА.....	29
3.1 Аналіз вимог.....	29
3.2 Архітектура системи.....	31
3.3 Аналіз та вибір апаратних засобів розробки.....	33
3.3.1 Вибір центрального комп'ютера	36
3.3.2 Вибір протокола обміну повідомленнями	36
3.3.3 Вибір Wi-Fi модуля.....	38
3.3.4 Проектування вузлів БМСА.....	41
3.4 Аналіз та вибір програмних засобів розробки.....	44
3.4.1 Вибір середовища розробки програми для Wi-Fi модуля	44
3.4.2 Вибір платформи домашньої автоматизації.....	47
3.5 Висновки до розділу 3.....	49
4 Конструювання програмно-апаратного забезпечення БМСА.....	50
4.1 Розробка алгоритму функціонування системи	50
4.2 Розробка MQTT тем.....	52
4.3 Розробка програмного забезпечення для модулів БМСА.....	54

4.4 Розгортання СДА Domoticz на БМСА.....	57
4.4.1 Конфігурування системи.....	57
4.4.2 Реалізація MQTT комунікації.....	58
4.4.3 Використання Node-RED.....	59
4.4.4 Реалізація алгоритму контролю.....	60
4.5 Розгортання СДА на платформі OpenHAB.....	62
4.5.1 Конфігурування системи.....	62
4.5.2 Реалізація алгоритму контролю.....	65
4.6 Висновки до розділу 4.....	65
Висновки.....	67
Перелік джерел посилань.....	68
Додаток А Текст програми.....	76
Додаток Б Слайди презентації.....	85

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

A/D	– Analog to Digital;
API	– Application Programming Interface;
BLE	– Bluetooth Low Energy;
CoAP	– Constrained Application Protocol;
IDE	– Integrated Development Environment;
IoT	– Internet of Things;
GUI	– Grafical User Interface;
HAS	– Home Automation System;
LPWAN	– Low Power Wide Area Networks;
MQTT	– Message Queue Telemetry Transport;
PIR	– Passive Infrared;
TCP	– Transmission Control Protocol;
UDP	– User Datagram Protocol;
WLAN	– Wireless Local Area Network;
WPAN	– Wireless Personal Area Network;
АЦП	– аналого-цифровий перетворювач;
БМСА	– бездротова мережа сенсорів та актуаторів;
ІЧ	– інфрачервоний;
МК	– мікроконтролер;
ОВКП	– опалення, вентиляція та кондиціонування повітря;
ПЗ	– програмне забезпечення;
СДА	– система домашньої автоматизації;
СРБ	– система Розумний будинок.

ВСТУП

Технологія домашньої автоматизації набуває все більшого значення в нашому житті, оскільки вона пропонує численні переваги, зокрема, більший комфорт, безпеку та енергоефективність. Система домашньої автоматизації (СДА) зазвичай включає в себе центральний комп'ютер з розгорнутим програмним забезпеченням (ПЗ) домашньої автоматизації і декількома розподіленими датчиками та виконавчими механізмами (актуаторами) [1].

Дротові з'єднання між центральним комп'ютером і вузлами сенсорів/актуаторів добре відомі, однак бездротові рішення є більш сучасною тенденцією, що надає можливість модернізувати існуючий звичайний будинок у «розумний» без суттєвої зміни апаратних установок. Ці рішення можуть базуватися на відомих відкритих платформах домашньої автоматизації (наприклад, Home Assistant, OpenHAB та Domoticz), а також спеціально розробленому обладнанні для бездротової передачі даних, зокрема заснованому на мікроконтролерах (МК) з інтегрованою мікросхемою Wi-Fi. В той же час існує багато проблем, пов'язаних з реалізацією бездротових мереж сенсорів та актуаторів (БМСА), зокрема необхідність підвищення надійності та безпеки роботи такої мережі за умови одночасного зниження її власного енергоспоживання [1].

Тому, актуальною є задача подальшого розвитку методів та засобів домашньої автоматизації, що базуються на програмному забезпеченні з відкритим вихідним кодом та БМСА.

Мета роботи – дослідження та реалізація масштабованого, доступного програмно-апаратного рішення для БМСА, що забезпечить ефективне функціонування СДА.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд технологій домашньої автоматизації

На сьогоднішній день технології домашньої автоматизації стають популярною парадигмою через свої численні переваги. У СДА побутова техніка підключається до існуючої або спеціалізованої мережі, і її можна контролювати і автоматизувати за допомогою центральної програми [1].

На рисунку 1.1 представлено базові підсистеми СДА [2].



Рисунок 1.1 – СДА на основі сенсорів та актуаторів [2]

Системи домашньої автоматизації можуть поліпшити якість повсякденного життя, оскільки вони роблять життя людини більш: зручним, здоровим, безпечним та енергоефективним. Комфорт пов'язаний, головним чином, з вимірюванням та аналізом умов навколишнього середовища та коригуванням цих умов відповідно до індивідуальних потреб. Комфорт включає управління системами опалення, вентиляції та кондиціонування повітря (ОВКП), контроль ролет та жалюзі на вікнах, контроль інтенсивності та кольору освітлення, голосовий контроль, тощо. На додаток до зручності, за допомогою СДА, також може бути досягнуто покращення здоров'я, шляхом постійного моніторингу якості повітря та генерації попереджень у випадку

виявлення витоків газу або погіршення показників повітря. Більш просунуті рішення для домашньої автоматизації можуть також включати вимірювання життєво важливих функцій людини, таких як моніторинг артеріального тиску і серцевого ритму [3].

З точки зору безпеки, СДА мають можливість виявляти та повідомляти випадки пожежі, крадіжки з проникненням до приміщення. Це включає виявлення факту відкритості дверей та вікон, керування замками воріт, а також застосування систем відеоспостереження з функціями розпізнавання обличчя.

Використовуючи СДА та відповідні алгоритми планування, можна досягти енергоефективного контролю електричних споживачів, таких як водонагрівачі, системи ОВК, побутова техніка та ін.. Отже, вартість послуг в такому будинку може бути значно зменшена, оскільки деякі споживачі можуть автоматично вимикатися або працювати з меншою потужністю в певних ситуаціях.

На додаток до перерахованих вище переваг, рішення для домашньої автоматизації також можуть бути дуже корисними для користувачів з деякими типами фізичних обмежень, таких як порушення зору або слуху або, наприклад, деменція [4]. Технологія домашньої автоматизації може надати їм незалежність і зробити їх життя набагато більш безпечним і комфортним. СДА також забезпечують більш незалежне життя для людей похилого віку, частка яких швидко зростає у світі. За прогнозами Євростату [5], частка людей похилого віку (80 років і старше) у загальній кількості населення ЄС-28, за прогнозами, зросте з 5,4% (27,3 млн.) у 2016 році до 12,7% (66,1 млн.) до 2080 року. Згідно з дослідженнями, СДА з голосовим контролем мають великий потенціал для полегшення повсякденного життя для літніх людей та осіб з особливими потребами [6].

Хоча СДА пропонують численні переваги користувачам (звичайним, похилого віку та користувачам з обмеженими можливостями), вони, зазвичай, дуже дорогі і відносно закриті (користувач не може налаштувати систему на власні потреби). Крім того, у більшості випадків необхідні також модифікації

електричних установок, що перешкоджає багатьом користувачам інвестувати в нову технологію домашньої автоматизації. З іншого боку, за допомогою недорогих апаратних засобів і ПЗ для домашньої автоматизації з відкритим вихідним кодом [7] можна створити потужне, гнучке і недороге рішення для СДА.

Протягом останніх років спостерігається використання розумних голосових помічників (Amazon's Alexa, Google's Assistant і Apple's Siri) у поєднанні з динаміками (Amazon's Echo, Google's Home або Apple's HomePod). Ці рішення дозволяють керувати роботою побутових приладів, але вони не є заміною СДА через дуже обмежену підтримку сценаріїв автоматизації.

1.2 Дослідження особливостей існуючих СДА

В останні роки з'явилося багато досліджень, присвячених СДА та системам типу Розумний будинок (СРБ).

Дослідники [1] представили рішення для домашньої автоматизації, яке базується на бездротових вузлах Wi-Fi, центральному комп'ютері (Raspberry Pi) з розгорнутим MQTT (Message Queue Telemetry Transport) брокером та ПЗ для платформи OpenHAB. Бездротові вузли, що базуються на ESP8266, здійснюють зв'язок з центральним комп'ютером через виділену або існуючу мережу Wi-Fi, використовуючи протокол MQTT.

В роботі [8] автори представляють результати розробки простої СДА, яка базується на застосуванні Raspberry Pi і Android. Система складається з групи датчиків, пристрою Raspberry Pi як основної серверної системи, маршрутизатора Wi-Fi і побутової техніки. Всі датчики підключаються безпосередньо до Raspberry Pi, а побутова техніка з'єднується з головним комп'ютером через маршрутизатор Wi-Fi. Raspberry Pi отримує дані від датчиків і пересилає їх в хмару, використовуючи існуючу мережу Wi-Fi. Користувачі можуть спостерігати й контролювати побутову техніку за

допомогою програми на Android, яка комунікує з головним комп'ютером за допомогою зв'язку Bluetooth.

Автори роботи [9] представили СРБ на базі МК ATMEL ATmega128, датчиків освітленості та температури, а також модуля Bluetooth - UART (HC-05). Підключеними пристроями можна керувати віддалено за допомогою Bluetooth і програми на Android.

В статті [10] комп'ютер Raspberry Pi був використаний як сенсорний вузол для СДА. Raspberry Pi збирає інформацію з 2 аналогових температурних датчиків (B57045K10 і NTC 10k) через зовнішній 8-розрядний A/D і D/A конвертер I2C (NXP PCF8591). Отримані дані доступні віддаленому користувачеві за допомогою сервера Apache Tomcat і вебслужб на основі RESTful.

Представлені в [11] рішення для домашньої автоматизації базуються на МК PIC, Arduino і модулі зв'язку LM400 Bluetooth. Розроблена підсистема безпеки працює за допомогою PIC12F675 та датчику PIR, підсистема керування освітленням включає Arduino UNO і датчик освітленості, а підсистема вимірювання температури використовує датчик LM35. Керувати пристроями можна віддалено за допомогою Android застосунку.

Інше рішення для СДА запропоновано в [12]. Бездротовий вузол, який складається з NodeMCU, фоторезистора (LDR), світлодіода (LED) і джерела звукового сигналу, передає/отримує дані у/з брокера повідомлень Eclipse Mosquitto [12], що працює на ПК з Windows. Бездротовий вузол виступає як видавець і підписник водночас. Інтенсивність світла визначається NodeMCU і публікується брокеру MQTT, тоді як команди для керування світлодіодом і зумером виводяться з брокера. У представленому рішенні клієнтські програми MQTTLens (Google Chrome) і MyMQTT (застосунок для Android) використовувалися для отримання даних сенсорів і перемикання світлодіода й зуммера. Обидві програми здатні підписатися та публікувати теми у MQTT.

В [14] автори представляють модифіковану версію [12]. У цьому рішенні плата ESP8266 NodeMCU взаємодіє з сенсором температури та

вологості DHT11 і фоторезистром LDR для контролю стану навколишнього середовища. Рівень яскравості 8*8 Neopixel matrix контролюється з урахуванням вимірюваного рівня інтенсивності світла. У цьому рішенні хмарна служба Adafruit.io була використана в якості брокера MQTT, тоді як графічний інтерфейс користувача (GUI) був розроблений з використанням приладової панелі Adafruit [15].

В роботі [16] автори представили рішення для домашньої автоматизації з використанням NodeMCU, підключеного до чотирьохканальних однополюсних реле Double Throw (SPDT). У цьому рішенні плата NodeMCU діє як вебсервер. Користувач може керувати побутовими приладами (світильниками, вентиляторами, лампочками та зарядними пристроями) за допомогою Android програми, яка була розроблена з використанням відкритого джерела MIT App Inventor 2. Розроблена програма дозволяє дистанційно керувати підключеними пристроями за допомогою об'єктів графічного інтерфейсу або за допомогою голосових команд.

Систему бездротового реагування на надзвичайні ситуації для пожежної небезпеки на основі технологій Інтернету речей (Internet of Things, IoT) представлено в роботі [16]. Бездротовий вузол включає модуль ESP-32, датчик виявлення полум'я, газові датчики (MQ-2 і MQ-5) і модуль GPS. Датчики виявляють небезпеку і попереджають місцеві організації аварійно-рятувальних служб, такі як відділення пожежної охорони та поліція, надсилаючи дані і координати місця розташування небезпеки до хмарної служби Adafruit.io [15] з використанням протоколу MQTT.

У роботі [18] була представлена система моніторингу приміщення для контролю показників життєзабезпечення навколишнього середовища. Система складається з сенсора iAQ (якість повітря в приміщенні), iAQ шлюзу, веб-порталу iAQ і мобільного застосунку iAQ. Система сенсорів iAQ, яка базується на платі Arduino Mega, містить п'ять датчиків: температури повітря (SHT10), відносної вологості (SHT10), монооксиду вуглецю (MQ7), вуглекислого газу (T6615) і світлочутливого резистору (GL5528). Зв'язок між

датчиком iAQ і шлюзом iAQ здійснюється за допомогою технології ZigBee з модулем XBee. Шлюз iAQ, який базується на модулі Wemos Mini D1, передає отримані дані (від датчика iAQ) до бази даних MySQL за допомогою вебслужб RESTfull. Користувач може отримати доступ до збережених даних через вебпортал iAQ або за допомогою мобільного застосунку Android iAQ.

У дослідженні [18] автори представляють недорогий бездротовий акустичний датчик для віддаленого моніторингу стану людей похилого віку. Система складається з електротного всеспрямованого мікрофона (CMA-4544PF-W), операційного підсилювача (op-amp), оптимізованого спеціально для використання в якості мікрофонного передпідсилювача (Maxim MAX4466), і МК ESP32, здатного записувати звуки навколишнього середовища щонайменше при 10 кГц і 12-бітній роздільній здатності. Отримані дані спочатку зберігаються в круговому буфері на МК ESP32, а потім передаються на сервер у необробленому форматі. Представлене рішення дозволяє відстежувати небезпечні ситуації шляхом ідентифікації звуків навколишнього середовища і, отже, дозволяє підвищити якість життя і безпеку людей похилого віку.

В [20] автори представляють “Розумну лампу”, що дозволяє оптимізувати тепловий комфорт у приміщенні та заощадити енергію. Цей пристрій включає плату Arduino Mega, датчик температури і відносної вологості (DHT22), модуль реального часу (RTC), інфрачервоний (ІЧ) світлодіод для керування ОВКП і модуль Bluetooth для передачі даних на смартфон або ноутбук. Електронні компоненти вбудовані в стандартну основу лампи. Використовуючи Android програму, створену за допомогою фреймворка MIT App Inventor, користувач може контролювати температуру і відносну вологість і перевіряти робочий стан системи ОВКП. Алгоритм керування, що розгорнутий на МК, автоматично вимикає систему ОВКП, використовуючи ІК-зв'язок поза робочим часом.

У роботі [21] автори представляють рішення для контролю якості повітря та освітленості в приміщенні. Система складається зі станцій

моніторингу та управління. Перша включає Arduino UNO, модуль зв'язку XBee S2, годинник реального часу (DS1307), датчик CO2 (K30) і фоторезистор, друга – модуль XBee S2 і 2-канальний релейний модуль. На основі фактичних і бажаних значень система дозволяє активувати/деактивувати системи освітлення та вентиляції повітря.

Платформа OpenHAB також використовувалася в роботі [22], де автори представляють систему визначення падіння на основі акселерометра. Система контролю падіння включає мікропроцесор Nordic Semiconductor NRF51822 з вбудованим приймачем Bluetooth з низьким енергоспоживанням (BLE) і акселерометром I2C (Bosch BMI055). Мікропроцесор отримує значення прискорення від датчика BMI055, аналізує дані, і при виявленні падіння надсилає інформацію про це в GUI OpenHAB.

Рішення СДА з використанням IoT технологій представлено в [23]. У цьому рішенні, хмарний сервер Amazon використовується для зберігання даних, зворотного зв'язку та управління датчиками. Експериментальна установка складається з мінікомп'ютера Raspberry Pi, плати реле, датчиків температури і вологості, механічних кінцевих вимикачів, датчика освітленості та датчика PIR для виявлення руху. Raspberry Pi періодично отримує дані від підключених датчиків і завантажує їх до хмарного сервісу Amazon. Користувач має можливість спостерігати за станом підключених сенсорів та актуаторів за допомогою Android програми.

В деяких розглянутих роботах представлено результати досліджень, присвячених виявленню аномальної поведінки мешканців в СДА з використанням передових алгоритмів обробки даних. Так, у роботі [24], наприклад, автори представляють візуальну систему, яка виявляє, розпізнає незвичайну поведінку мешканців і прогнозує їх стан здоров'я.

В роботі [25] автори представляють виявлення аномальної поведінки резидентів з використанням алгоритмів машинного навчання, а в [26] представлено оцінку умов благополуччя літніх людей, що базується на використанні побутових приладів.

В [27] автори вводять контролер нечіткої логіки для покращення управління енергоспоживанням в будинку. Нечітка логіка також була використана в [28], де автори представляють прогностичну модель для виявлення газів.

Результати дослідження прикладів використання МК та графічного інтерфейсу у розглянутих рішеннях для СДА наведено в таблицях 1.1 та 1.2.

Таблиця 1.1 – Огляд використовуваних в СДА МК

Мікроконтролер	Посилання
ATmega128	[9]
ESP8266	[1], [12], [14], [16]
ESP32	[17], [19]
Raspberry Pi	[8], [10], [23]
Arduino UNO	[11], [21]
Arduino Mega	[18], [20]
PIC (16F877, PIC12F675)	[11]

Таблиця 1.2 – Огляд використовуваних інтерфейсів в СДА

GUI	Посилання
Android	[8], [9], [11], [18], [16], [20], [23]
OpenHAB	[1], [22]
Вебсервіс (RESTful)	[10], [18]
Adafruit.io dashboard	[14], [17]

На основі проведених досліджень варіантів реалізації СДА, можна представити їх узагальнену класифікацію (рис. 1.2).

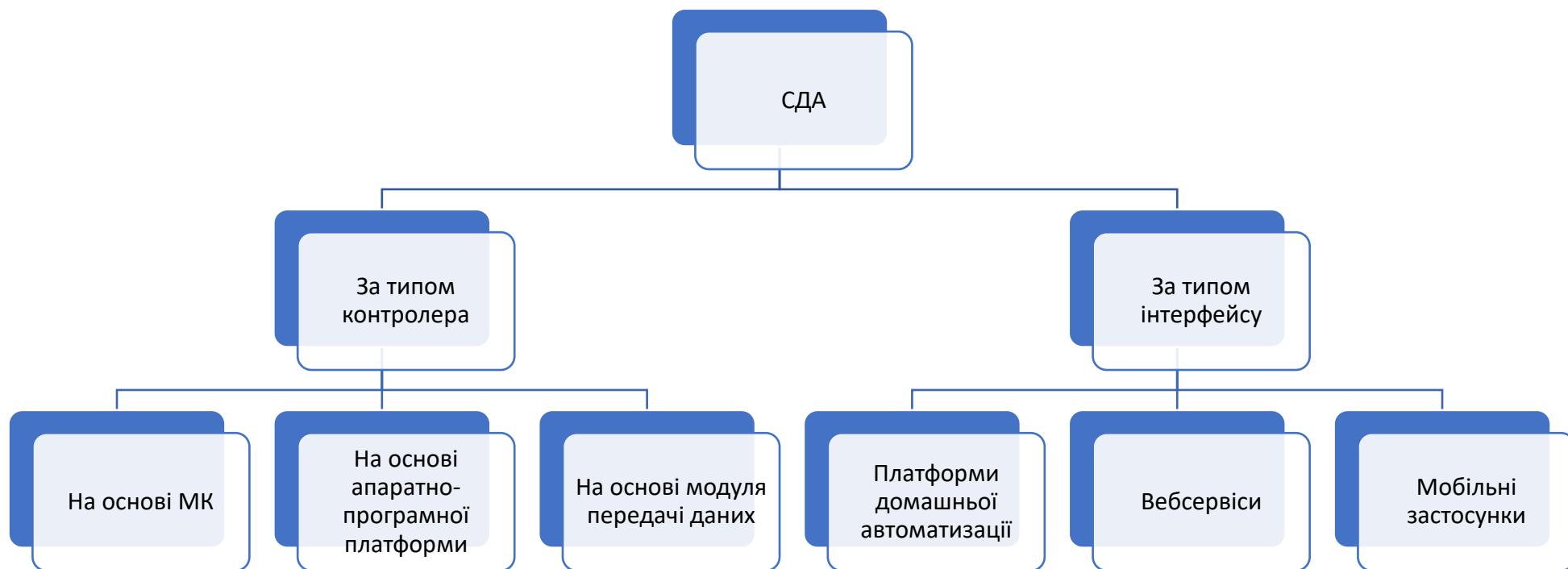


Рисунок 1.2 – Узагальнена класифікація СДА

Таким чином, рішення, що засновані на МК без інтегрованого комунікаційного блоку, включають додатковий комунікаційний модуль, наприклад, ZigBee ([18], [21]), Bluetooth ([9], [11], [20]) або інший, що дозволяє здійснювати обмін даними між центральним комп'ютером і бездротовим вузлом. Оскільки модулі зв'язку збільшують загальний розмір і ціну такого вузла, за допомогою цього підходу досить складно створити оптимізоване та економічно ефективне рішення.

У деяких інших рішеннях ([8], [10], [23]) вузли сенсорів/актуаторів були реалізовані за допомогою мінікомп'ютерів Raspberry Pi. Хоча цей вбудований комп'ютер є дуже потужним, він не дуже підходить для бездротового вузла з наступних причин: ціна, споживання енергії, розмір і відсутність А/D-перетворювача. Ціна вузла на Raspberry Pi з додатковими датчиками і актуаторами може бути досить високою. Крім того, плата Raspberry Pi не дуже мініатюрна і має набагато більш високе енергоспоживання, ніж більшість МК. Raspberry Pi не включає А/D-перетворювач, тому в разі використання аналогових датчиків необхідно використовувати зовнішній перетворювач. Отже, складність і ціна такого бездротового вузла можуть бути відносно високими.

Деякі розглянуті рішення діють за допомогою хмарних сервісів (наприклад, [14], [17], [23]), що означає, що для таких рішень потрібний постійний доступ до Інтернету для належного функціонування. Такі СДА легше реалізувати, однак, вони завжди підвищують ризики безпеки, надійності та конфіденційності. Доступ до Інтернету та хмарних сервісів досить простий у реалізації, але доступ до цих послуг не є необхідним для базової та належної роботи СДА.

У більшості розглянутих СДА користувачі не можуть реалізувати власні алгоритми керування або планування. В той же час, використовуючи правила OpenHAB або алгоритми (потoki) Node-RED, користувачі можуть реалізувати власні алгоритми управління або планування або навіть додати нові функціональні можливості, такі як машинне навчання.

У більшості заявлених рішень ([8], [9], [11], [16], [18], [20], [23]) користувачі мають можливість спостерігати за станом підключених сенсорів і керувати актуаторами за допомогою розробленої Android програми. У більшості випадків застосунок для Android розроблено для певної конфігурації. Отже, у випадку додаткових сенсорів або актуаторів, застосунок повинен бути змінений і відновлений.

1.3 Постановка завдань дипломної роботи

Як показали проведені дослідження, сучасні СДА дозволяють підвищити рівень комфорту, безпеки та заощаджувати ресурси, а відповідно і кошти мешканців будинків та квартир. Існуючі системи базуються на використанні різних апаратних та програмних рішень. Відкриті платформи домашньої автоматизації, зокрема OpenHAB, дозволяють швидко розгорнути СДА та реалізувати власні алгоритми управління побутовими приладами.

Тим не менш, розробники і користувачі СДА стикаються з чотирма основними проблемами: висока вартість, погана керованість і гнучкість, а також труднощі з досягненням безпеки. Тому, подальше удосконалення технологій домашньої автоматизації залишається актуальною задачею.

Мета роботи – дослідження та реалізація масштабованого, доступного програмно-апаратного рішення для БМСА, що забезпечить ефективне функціонування СДА.

Для досягнення поставленої мети треба розв'язати наступні завдання:

- дослідити технології реалізації БМСА;
- виконати аналіз вимог та розробку архітектури СДА на основі БМСА;
- виконати проектування та конструювання програмного та апаратного забезпечення БМСА;
- розробити конфігурацію СДА на основі БМСА.

2 ДОСЛІДЖЕННЯ СТРУКТУРНИХ ТА ФУНКЦІОНАЛЬНИХ ОСОБЛИВОСТЕЙ БЕЗДРОТОВИХ МЕРЕЖ СЕНСОРІВ ТА АКТУАТОРІВ

2.1 Структура та галузі використання БМСА

БМСА складаються з вузлів, що живляться від автономних джерел живлення, з обмеженими можливостями обчислення та радіозв'язку (рис. 2.1). З точки зору функціонування, БМСА об'єднує чотири підсистеми: підсистему сенсорів, яка сприймає навколишнє середовище, підсистему обробки, яка виконує локальні обчислення отриманих даних, підсистему зв'язку, яка відповідає за обмін повідомленнями з сусідніми вузлами та сервером, а також підсистему актуаторів, що виконують певні дії у відповідь на керуючі впливи, що формуються на сервері. Фактично, вузли БМСА – це бездротові мікрокомп'ютери, які сприймають, обробляють, передають дані та перетворюють їх на дії [30].

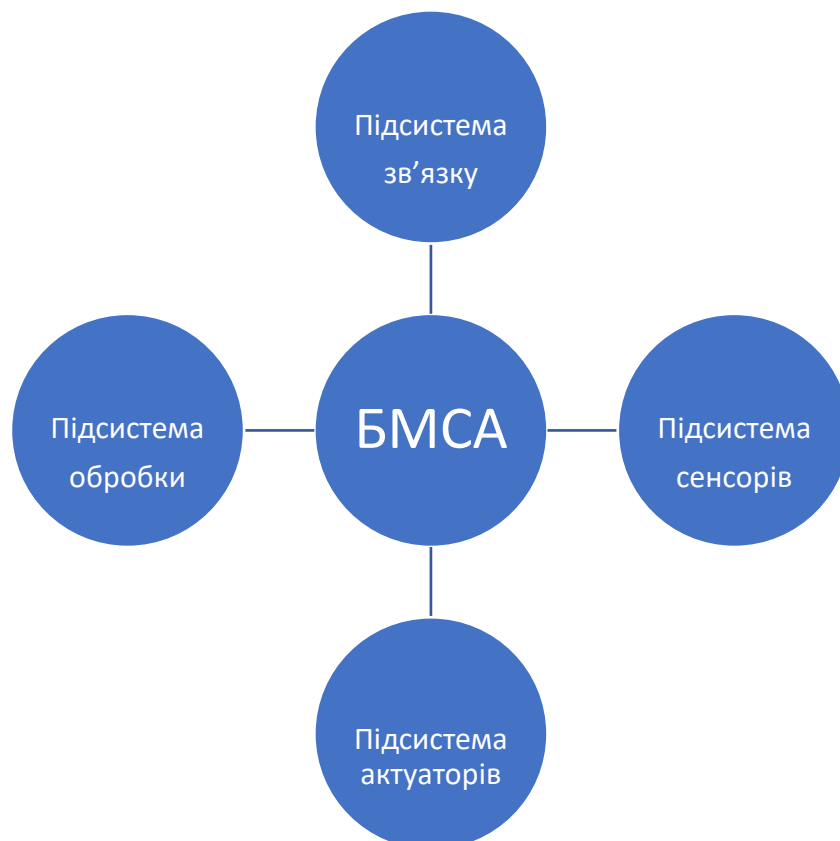


Рисунок 2.1 – Схематичне зображення структури БМСА [30]

Таким чином, БМСА складається з бездротових сенсорних пристроїв, які працюють разом для досягнення спільної мети [31].

БМСА може мати одну або декілька базових станцій, які збирають дані з усіх сенсорних пристроїв. Ці базові станції служать інтерфейсом, через який БМСА взаємодіє із зовнішнім світом. Основна передумова побудови БМСА полягає у виконанні мережевого зондування з використанням багатьох рудиментарних датчиків замість використання більш традиційного підходу до розробки кількох дорогих і складних сенсорних модулів [32].

Потенційною перевагою мережевого зондування є більше охоплення, точність і надійність за умови нижчої вартості [31]-[32].

БМСА є активною сферою із різними застосуваннями:

- безпека;
- моніторинг навколишнього середовища;
- система охорони здоров'я;
- моніторинг Сонячної системи та загроз в космосі;
- моніторинг наземних та водних ресурсів;
- збір розвідданих для оборони;
- точне землеробство;
- моніторинг цивільної структури;
- аналіз і прогноз погоди та клімату;
- моніторинг виробничих процесів [31], [33].

Для кожної області застосування існують різні технічні проблеми, які дослідники намагаються вирішувати. Але, беззаперечним є факт, що використання БМСА покращило функціональність і «розумність» багатьох існуючих систем та застосувань. Зокрема, БМСА забезпечують гнучкі рішення для моніторингу, виявлення та сповіщення про різноманітні процеси та події в СДА.

2.2 Особливості БМСА

БМСА все частіше використовуються при реалізації СДА, оскільки надають сукупність переваг.

Зниження витрат на встановлення – обумовлене тим, що не потрібно прокладати кабель, оскільки матеріал, а також робота з прокладання кабелів (наприклад, у стінах) є дорогими.

Легке розгортання, інсталяція та покриття – бездротові вузли можна встановити практично будь-де, підключення до домашньої мережі здійснюється миттєво шляхом простого монтажу вузлів у зоні, що також допомагає збільшити зону покриття.

Масштабованість системи та просте розширення – розгортання бездротової мережі є особливо вигідним, коли через нові або змінені вимоги необхідно розширення мережі. На відміну від дротових інсталяцій, додаткові вузли не вимагають додаткових кабелів, що робить розширення досить тривіальним.

Естетичні переваги – реалізація БМСА допомагає задовольнити естетичні вимоги, наприклад, для репрезентативних будинків з повністю скляною архітектурою, або історичних будівель, де дизайн або причини консервації не дозволяють прокладати кабелі.

Інтеграція мобільних пристроїв – з бездротовими мережами, зв'язування мобільних пристроїв, таких як КПК і смартфони, з СДА стає можливим всюди і в будь-який час, оскільки точне фізичне розташування пристрою не є вирішальним для з'єднання (поки пристрій знаходиться в зоні досяжності мережі) [34].

Таким чином, зменшення кількості кабелів, динамічне формування мережі, низька вартість і просте розгортання – основні переваги БМСА. Тому, вони є не лише привабливим вибором під час ремонту та реконструкції приміщень, але й пріоритетним напрямом при реалізації нових проєктів.

2.3 Дослідження бездротових протоколів для БМСА

Бездротова мережа малого радіусу дії підтримується чотирма протоколами: Bluetooth, UWB, ZigBee та Wi-Fi, які відповідають стандартам IEEE 802.15.1, 802.15.3, 802.15.4 та 802.11a/b/g, відповідно. IEEE визначає лише фізичний (physical layer, PHY) і MAC (Medium Access Control) шари в їх стандартах.

Автори [35] порівняли основні функції та поведінку протоколів Bluetooth and Wi-Fi з точки зору різних показників, включаючи потенціал, топологію мережі, безпеку, якість підтримки обслуговування та енергоспоживання.

У [36] дослідники порівнювали MAC IEEE 802.11e та IEEE 802.15.3. Їх результати показали, що різниця в пропускну здатності між ними досить мала. Крім того, керування живленням 802.15.3 легше, ніж 802.11e.

Для ZigBee і Bluetooth автор [37] досліджував їхні сильні та слабкі та стверджував, що ZigBee через протокол 802.15.4 може задовольнити більшу кількість реальних потреб, ніж Bluetooth, завдяки довгостроковій роботі від батареї, більшому корисному діапазону, гнучкості у ряді вимірів і надійності комірчастої мережевої архітектури.

Bluetooth, також відомий як стандарт IEEE 802.15.1 заснований на системі бездротового радіозв'язку, призначений для малого радіусу дії і дешевих пристроїв для заміни кабелів периферійних пристроїв комп'ютера, таких як миші, клавіатури, джойстики та принтери. Цей діапазон програми відомі як бездротова персональна мережа (WPAN).

UWB свого часу привернув велику увагу як високошвидкісний бездротовий зв'язок малої дальності для приміщень [38]. Пропускна здатність UWB становить понад 110 Мбіт/с (до 480 Мбіт/с), що може задовольнити більшість мультимедійних застосунків, таких як доставка аудіо та відео в домашній мережі, а також може виступати в якості бездротової заміни кабелю послідовної високошвидкісної шини (USB 2.0 та IEEE 1394).

ZigBee через IEEE 802.15.4 визначає специфікації для низькошвидкісної WPAN (LR-WPAN) для підтримки простих пристроїв, які споживають мінімальну енергію та зазвичай працюють у персональному робочому просторі розміром 10 м. ZigBee забезпечує самоорганізовану, багатострибкову та надійну сітчасту мережу з тривалим терміном служби батареї [39] - [40].

Wi-Fi включає IEEE 802.11a/b/g стандарти для бездротових локальних мереж (WLAN). Він дозволяє користувачам користуватися Інтернетом на широкосмуговій швидкості при підключенні до точки доступу (AP) або в режимі ad hoc. Архітектура IEEE 802.11 складається з кількох компонентів, які взаємодіють, щоб забезпечити бездротову локальну мережу, яка підтримує мобільність станції прозоро для верхніх рівнів.

Таблиця 2.1 узагальнює основні відмінності між чотирма розглянутими протоколами. Кожен протокол базується на стандарті IEEE [41].

Очевидно, що UWB і Wi-Fi забезпечують більш високу швидкість передачі даних, ніж Bluetooth і ZigBee. Загалом Bluetooth, UWB і ZigBee призначені для зв'язку WPAN (близько 10 м), тоді як Wi-Fi орієнтований на WLAN (близько 100 м). Однак ZigBee також може досягати 100 м у деяких застосунках.

Bluetooth і ZigBee призначені для портативних пристроїв малого радіусу дії та обмеженого заряду акумулятора. Отже, вони пропонують дуже низьке енергоспоживання, що у деяких випадках не матиме відчутного впливу на час роботи акумулятора. UWB пропонується для застосунків із високою швидкістю передачі даних. З іншого боку, Wi-Fi розроблено для більш тривалого з'єднання та підтримки пристроїв із суттєвим енергоспоживанням [41].

Поточне споживання енергії в умовах передачі (TX) і прийому (RX) для кожного протоколу показано в таблиці 2.2 [41].

Таблиця 2.1 – Порівняння протоколів передачі даних [41]

Стандарт	Bluetooth	UWB	ZigBee	Wi-Fi
IEEE специфікація	802.15.1	802.15.3a *	802.15.4	802.11a/b/g
Діапазон частот	2.4 ГГц	3.1-10.6 ГГц	868/915 МГц; 2.4 ГГц	2.4 ГГц; 5 ГГц
Максимальна швидкість сигналу	1 Мб/с	110 Мб/с	250 Кб/с	54 Мб/с
Номинальний діапазон	10 м	10 м	10 - 100 м	100 м
Номинальна потужність передачі	0 - 10 dBm	-41.3 dBm/МГц	(-25) - 0 dBm	15 - 20 dBm
Кількість радіочастотних каналів	79	1-15	1/10; 16	14 (2.4 ГГц)
Пропускна здатність каналу	1 МГц	500 МГц - 7.5 ГГц	0.3/0.6 МГц; 2 МГц	22 МГц
Тип модуляції	GFSK	BPSK, QPSK	BPSK (+ ASK), O-QPSK	BPSK, QPSK COFDM, CCK, M-QAM
Поширення	FHSS	DS-UWB, MB-OFDM	DSSS	DSSS, CCK, OFDM
Максимальна кількість вузлів комірок	8	8	> 65000	2007
Захист даних	16-біт CRC	32-біт CRC	16-біт CRC	32-біт CRC

Таблиця 2.2 – Споживання чіпсетами при використанні різних протоколів [41]

Стандарт	Bluetooth	UWB	ZigBee	Wi-Fi
Чіпсет	BlueCore2	XS110	CC2430	CX53111
Робоча напруга, В	1.8	3.3	3.0	3.3
Передача даних, мА	57	~227.3	24.7	219
Прийом даних, мА	47	~227.3	27	215
Швидкість передачі даних, Мб/с	0.72	114	0.25	54

На рисунку 2.2 показано енергоспоживання в мВт для кожного протоколу. Очевидно, що протоколи Bluetooth і ZigBee споживають менше енергії в порівнянні з UWB і Wi-Fi. Базуючись на швидкості передачі даних, порівняння нормалізованого споживання енергії наведено на рисунку 2.3. З точки зору одиниці вимірювання МДж/Мб, UWB і Wi-Fi мають кращу ефективність споживання енергії [41].

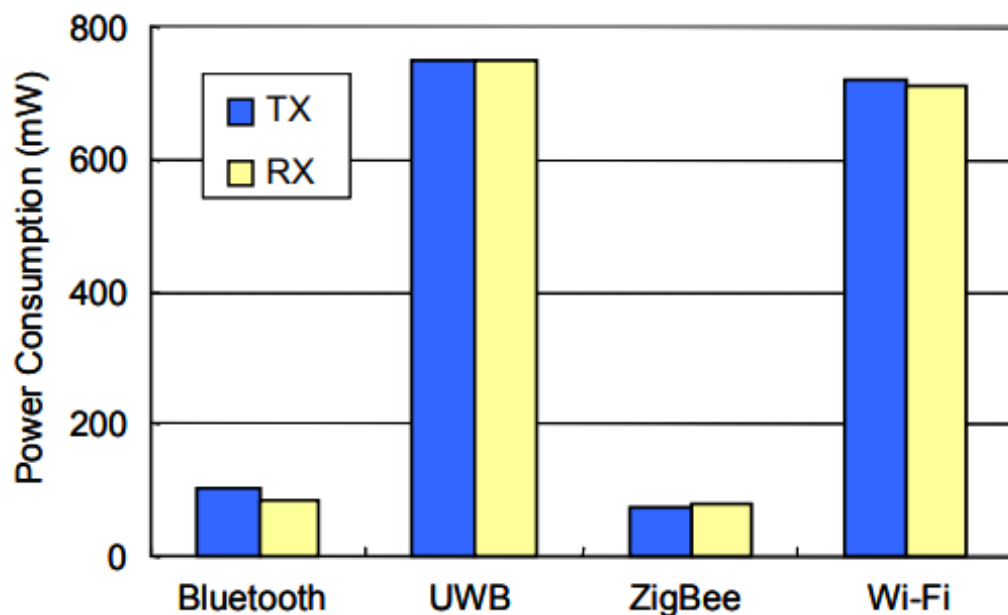


Рисунок 2.2 – Порівняння енергоспоживання при використанні різних протоколів [41]

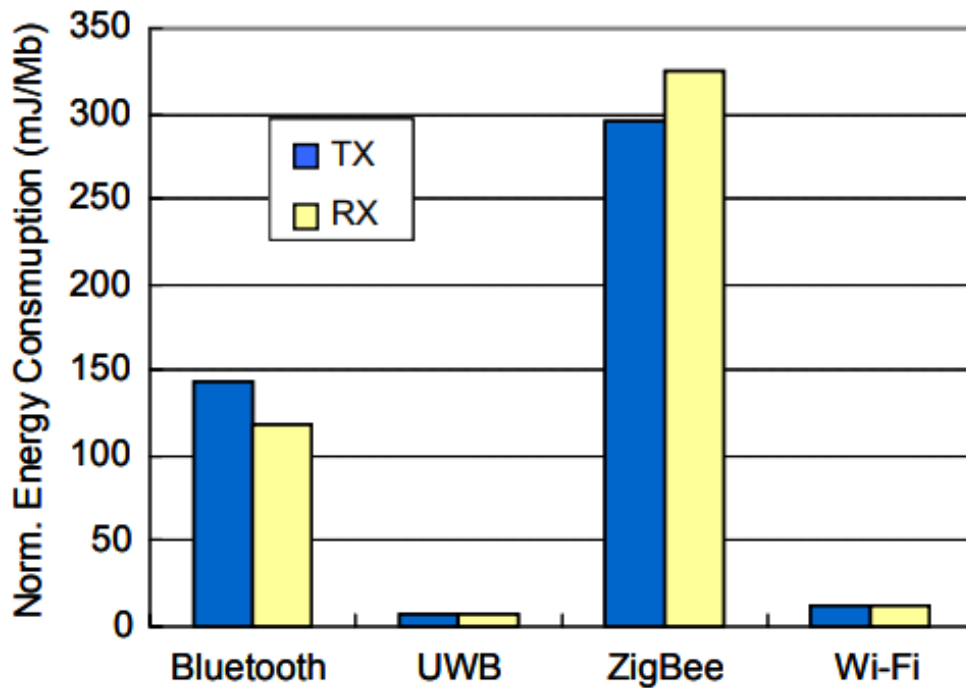


Рисунок 2.3 – Порівняння нормалізованого енергоспоживання при використанні різних протоколів [41]

2.4 Висновки до розділу 2

Як показали проведені дослідження, Bluetooth і ZigBee підходять для застосунків із низькою швидкістю передачі даних із обмеженим зарядом акумулятора (наприклад, для мобільних пристроїв та сенсорних мереж, що працюють від батарей), завдяки їх низькому енергоспоживанню, що забезпечує тривалий термін служби. З іншого боку, для реалізацій систем з високою швидкістю передачі даних (таких як системи аудіо/відеоспостереження) UWB і Wi-Fi будуть кращими рішеннями через їх низьке нормоване енергоспоживання.

Оскільки Wi-Fi технологія є поширеною та на ринку є багато доступних користувачам комунікаційних Wi-Fi модулів, є сенс обрати саме цю технологію для реалізації БМСА.

3 ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ БМСА

3.1 Аналіз вимог

У наведеному нижче списку подано огляд найважливіших вимог до проєктованої системи.

Зручний інтерфейс: користувач може легко керувати системою локально або віддалено за допомогою простого вебінтерфейсу СДА.

Безпека та автентифікація: лише авторизований користувач може увійти в систему (локально чи віддалено) для керування, контролю та моніторингу. Якщо система виявляє зловмисників, вона має негайно сповістити власника системи та на деякий час заблокувати можливість входу.

Низька вартість вузла/велика кількість вузлів: якщо говорити про СДА, то для забезпечення повноцінної автоматизації можуть знадобитися десятки вузлів. Однак ринок вимагає конкурентоспроможної продуктивності (порівняно з дротовими мережами), щоб забезпечити низьку вартість системи. Крім того, протоколи також повинні масштабуватися до великої кількості вузлів, наприклад, забезпечуючи надійну доставку повідомлень.

Охоплення великої території: проблема полягає в тому, що пристрої СДА можуть бути розосереджені на великих площах. Оскільки трансивери не повинні споживати так багато енергії, вони не можуть бути створені з діапазоном передачі, достатнім для того, щоб датчики могли досягати відповідних контролерів або виконавчих механізмів безпосередньо. Крім того, вони можуть покладатися на інфраструктуру точок доступу та дротову магістральну мережу (або особливо чутливі приймачі).

Масштабованість системи: тобто здатність системи, мережі або процесу ефективно справлятися зі зростаючим обсягом даних або її здатність розширюватися відповідно до цього зростання. Наприклад, оновлення

системи шляхом додавання/видалення модуля апаратного інтерфейсу повинно бути легким завданням.

Діаграму діяльності представлено на рисунку 3.1. Користувач СДА на основі використання БМСА матиме можливість моніторити показники середовища (температура, вологість, присутність), а також контролювати роботу обладнання на основі інформації, що надходить від сенсорів та актуаторів.

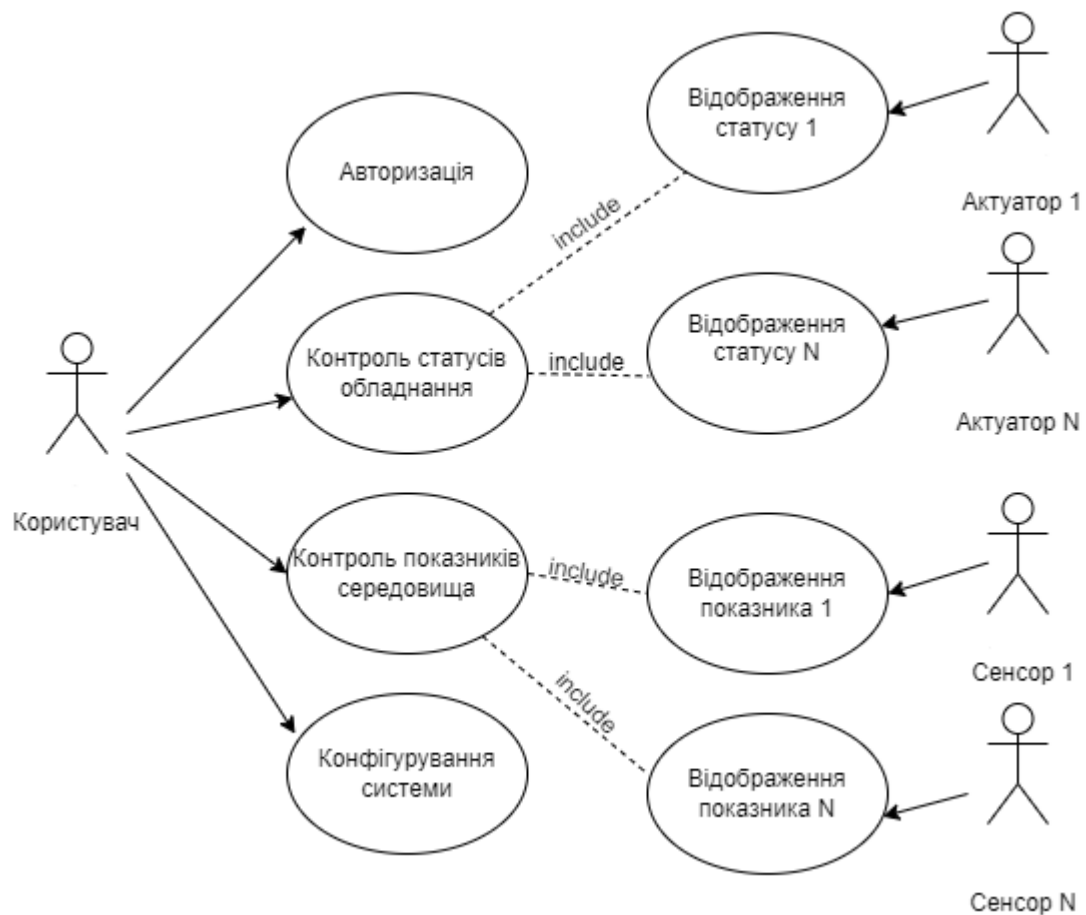


Рисунок 3.1 – UML діаграма діяльності

Щоб почати роботу з системою, користувач має авторизуватися через інтерфейс входу, який передає дані в модуль безпеки СДА для фіксації входу та подальших дій користувачів системи.

Отримання статусу актуатора або показника датчика здійснюється шляхом отримання цих даних безпосередньо від модуля зв'язку, який, у свою чергу, отримує необхідні дані через апаратний інтерфейс, представлений у

МК. Поки цей процес виконується, а статус неодноразово перевіряється, будь-які зміни записуватимуться в СДА і з'являтимуться в інтерфейсі системи.

Зміна статусу актуатора через інтерфейс керування здійснюється шляхом надсилання даних безпосередньо до модуля зв'язку, який, у свою чергу, надсилає їх до апаратного забезпечення через інтерфейс, представлений у МК, щоб застосувати необхідні зміни. Під час процесу надсилання даних і зміни приводу зміни записуються в СДА і відображаються в інтерфейсі системи.

Налаштування нової конфігурації системи виконується за допомогою правил, які прописуються та зберігаються з подальшою активацією.

3.2 Архітектура системи

Узагальнену архітектуру проєктованої системи представлено на рисунку 3.2.

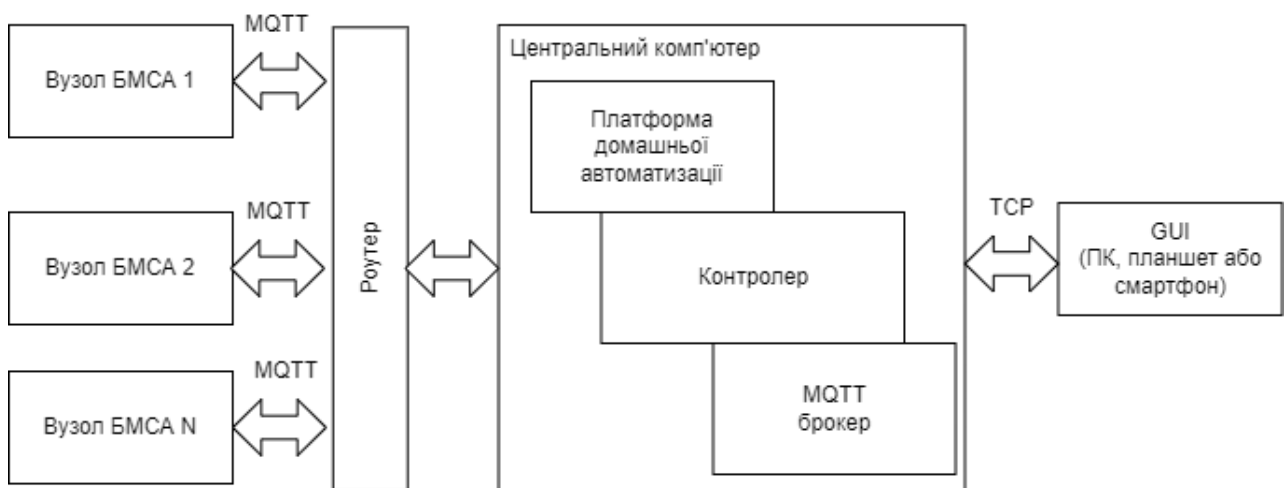


Рисунок 3.2 – Узагальнена архітектура системи

Архітектура системи складається з трьох основних частин:

- центральний комп'ютер з розгорнутим програмним забезпеченням домашньої автоматизації;

- кілька вузлів БМСА на основі комунікаційних модулів та підключених сенсорів/актуаторів;
- бездротовий маршрутизатор (роутер).

Вузли БМСА можуть бути створені на основі Wi-Fi-модулів, тобто електронних пристроїв з підтримкою Wi-Fi з приєднаними датчиками (вузол-сенсор), виконавчими пристроями (вузол-актуатор) або комбінацією обох (вузол сенсор/актуатор). Використовувані Wi-Fi вузли можуть базуватися на МК ESP8266 або ESP32 з вбудованою мікросхемою Wi-Fi, що комунікує з центральним комп'ютером через бездротовий маршрутизатор.

Система домашньої автоматизації, розгорнута на контролері центрального комп'ютера, отримує інформацію з різних вузлів-сенсорів (температура, вологість, виявлення руху і т.д.) і, на основі налаштувань користувача та сценаріїв автоматизації, керує актуаторами (двигунами, обігрівачами і т.д.).

Важливою частиною програмного забезпечення для домашньої автоматизації є графічний інтерфейс користувача (GUI). У представленому рішенні користувачі можуть отримати доступ до графічного інтерфейсу домашньої автоматизації за допомогою веббраузера або програми для смартфонів. Користуючись графічним інтерфейсом, користувач може спостерігати за показниками встановлених сенсорів і змінювати стани (увімкнено/вимкнено) окремих актуаторів. Більш обізнані користувачі можуть створювати та застосовувати власні сценарії автоматизації або алгоритми планування, які вмикають/вимикають побутову техніку відповідно до певних умов часу або події.

Також має бути розгорнуто MQTT брокер Mosquitto [13]. Mosquitto – це легкий і відкритий брокер повідомлень, що реалізує протокол MQTT [42]. За допомогою цього протоколу здійснюється обмін даними між центральним комп'ютером і вузлами БМСА.

Вузли-сенсори використовують тільки механізм MQTT публікування, в той час як вузли-актуатори використовують механізм MQTT підписки. У

випадку з вузлом-сенсор/актуатор, механізми публікації та підписки реалізуються на Wi-Fi модулі. Вузол-сенсор отримує дані з приєднаних датчиків, обробляє дані і публікує дані до брокера MQTT, який оновлює дані в базі даних. Брокер MQTT потім пересилає нещодавно отримані дані всім підписникам MQTT.

3.3 Аналіз та вибір апаратних засобів розробки

3.3.1 Вибір центрального комп'ютера

В якості центрального комп'ютера було вирішено обрати мінікомп'ютер з метою мініатюризації проєктного рішення. Серед існуючих варіантів було розглянуто два найбільш популярних: Raspberry Pi B+ та Intel Edison. Порівняльну характеристика цих мінікомп'ютерів наведено в таблиці 3.1 [43].

Як показує порівняльний аналіз, Edison програє порівняно з Raspberry Pi через:

- відсутність USB;
- відсутність відео;
- швидкість процесора;
- високу вартість;
- роз'єм вводу/виводу неможливо використовувати без додаткової плати.

В роботі [44] автори порівнюють недорогий обчислювальний термінал Raspberry Pi з вбудованим обчислювальним IoT модулем Edison шляхом запуску алгоритмів машинного навчання на обох пристроях. Було визначено такі показники як час роботи, середнє використання ЦП і пам'яті. Показано, що існує різниця у швидкості процесора та енергоспоживанні в Edison та Raspberry Pi. Ідеальна система мінімізує час роботи, максимізує використання процесора та використовує невеликий обсяг пам'яті [44].

Таблиця 3.1 – Порівняльна характеристика Raspberry Pi B+ та Intel Edison [43]

Характеристика	Raspberry Pi B+	Intel Edison
Вартість, грн	1330	2775
Відео	HDMI, Direct LCD, Composite	-
Аудіо	Стерео вихід	I2S вихід
Flash пам'ять	Мікро SD рознім	SDIO інтерфейс; 4Гб вбудована Flash пам'ять
Оперативна пам'ять	512Мб	1Гб
Процесор	700МГц ARM1176JZF-S	500МГц двоядерний Atom процесор; 100МГц Quark MCU
Інтерфейс вводу/виводу	27 пінів на рознімі 0.1"	70 пінів на міжплатному рознімі 0.4 мм
Ethernet	10/100 вбудований	-
Wi-Fi	-	Двodiaпазонний 802.11 (a/b/g/n)
Bluetooth	-	Bluetooth 2.1/4.0
USB	4 порти	1 USB-OTG
Інші інтерфейси	SPI, UART, I2C	SPI, UART, I2C, PWM
Енергоспоживання	5В * 600мА (~3Вт)	3.3В - 4.5В @ <1Вт
Розміри	85мм x 56мм x 19.5мм	60мм x 29мм x 8мм

На рисунку 3.3 показано порівняння Intel Edison і Raspberry Pi. Raspberry Pi або перевершив, або зрівнявся з Intel Edison за кожним з критеріїв [44].

Тому, модель Raspberry Pi 3 B [45] була обрана як центральний комп'ютер. Raspberry Pi (рис. 3.4) – це потужний одноплатний комп'ютер з інтегрованою бездротовою локальною мережею та Bluetooth. Raspberry Pi добре підходить на роль центрального комп'ютера СДА, оскільки має багато переваг: потужний процесор; відносно низьке енергоспоживання; малі габарити; низька ціна.

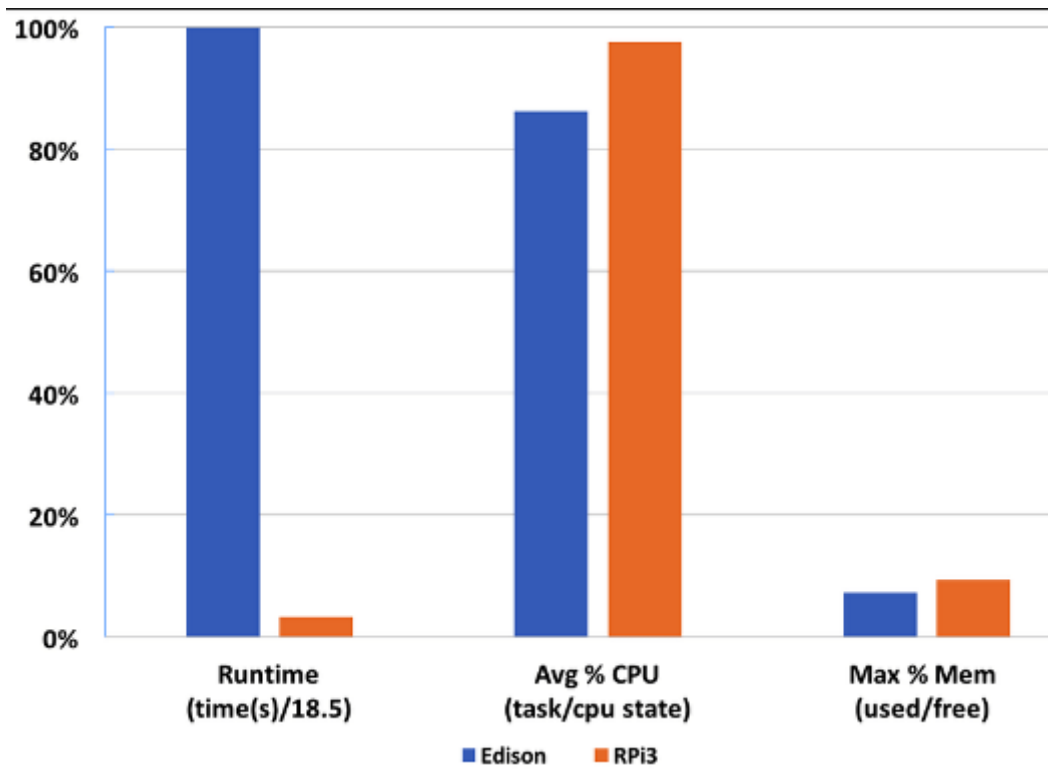


Рисунок 3.3 – Практичне порівняння роботи Raspberry Pi та Intel Edison [44]



Рисунок 3.4 – Raspberry Pi 3 B [45]

3.3.2 Вибір протокола обміну повідомленнями

Зв'язок між центральним комп'ютером і вузлами БМСА може бути реалізований за допомогою протоколу MQTT [42]. MQTT є відкритим і відносно простим протоколом обміну повідомленнями, призначеним для комунікацій Machine-to-Machine (M2M).

MQTT – це протокол обміну повідомленнями для публікації/підписки, де кожне повідомлення публікується на адресу, відому як «тема». Клієнт MQTT може бути або видавцем, або підписником. Видавець публікує теми до сервера MQTT (брокера), тоді як підписники отримують теми від брокера MQTT. Кожен клієнт, який підписався на певну тему, отримує кожне повідомлення, яке опубліковано в цій темі. Модель видавця/підписника дозволяє клієнтам MQTT спілкуватися один-на-один, один-до-багатьох і багато-до-одного.

MQTT стандартизований OASIS [42] і пропонує три рівні якості обслуговування (QoS) [4]. На рівні QoS 0 видавець не отримує жодного підтвердження від сервера MQTT для опублікованого пакета, тоді як на рівні QoS 1 видавець отримує підтверджене повідомлення, включаючи ідентифікатор пакета, з сервера MQTT. На рівні QoS 2 між клієнтом і сервером MQTT передаються чотири пакети [12].

Для реалізації даного проєкту пропонується використовувати MQTT-брокер під назвою Mosquitto [14], який працює на мінікомп'ютері Raspberry Pi. Mosquitto доступний для всіх основних платформ (Windows, Mac та ін.) і підходить для використання на всіх пристроях, від малопотужних одноплатних комп'ютерів до повноцінних серверів [13].

Протокол MQTT успішно впроваджено у багатьох застосунках для домашньої автоматизації та IoT [1], [4], [12], [14], [17], [46].

Як показали проведені дослідження, існують й інші протоколи обміну повідомленнями для комунікацій M2M: DDS, AMQP, JMS, REST, CoAP [47], [48]. Зокрема, CoAP (Constrained Application Protocol) – це протокол обміну

повідомленнями, орієнтований на запит на основі UDP, в той час як MQTT орієнтований на публікацію підписок і використовує TCP. Порівняння між MQTT і CoAP наведено в таблиці 3.2 [49].

Таблиця 3.2 – Порівняльна характеристика протоколів обміну повідомленнями [49]

Характеристика	MQTT	CoAP
Стандарт	ISO/IEC PRF 20922	RFC 7252
Парадигма	Видавець-підписник	Запит-відповідь
Кількість повідомлень	16	4
Розмір заголовку	2 байта	4 байта
Протокол транспортного рівня	TCP	UDP
RESTful	-	+

Комунікаційні інфраструктури M2M є основою будь-якої платформи IoT, забезпечуючи безпечний масивний обмін даними між сенсорами та актуаторами та інформаційними системами. Комунікаційні архітектури M2M можна класифікувати як дві основні категорії: монолітні та ієрархічні. Обидва підходи мають взаємодоповнюючі плюси та мінуси [50].

Ієрархічні архітектури є більш гнучкими, універсальними та масштабованими, ніж монолітні, але ними також складніше керувати та захищати їх, оскільки вони складаються з кількох сегментів мережі (і, отже, кількох проміжних вузлів) і поєднують різні комунікаційні технології, тому поверхня атаки є більшою [50].

З одного боку, ієрархічні архітектури включають різномірні інфраструктури, які поєднують технології малого радіусу дії та середньої швидкості передачі даних (наприклад, технології WPAN, такі як Bluetooth або Zigbee, або технології WLAN, такі як Wi-Fi) з технологіями високої швидкості передачі даних великого радіусу дії (як бездротові, такі як стільникові технології, так і дротові, на основі витой пари або волоконної оптики) [50].

З іншого боку, монолітні архітектури зазвичай покладаються на єдину технологію для з'єднання сенсорів і актуаторів та інформаційних систем. У цьому випадку досить часто комунікаційна інфраструктура належить телекомунікаційному оператору, який відповідає за її управління та обслуговування, за рахунок збільшення операційних витрат на платформу IoT. У цьому контексті зростає популярність технологій LPWAN (Low Power Wide Area Networks), таких як SigFox, LoRa, NB-IoT або LTE-M [51]. Ці технології забезпечують широке охоплення, низьке споживання та низьку швидкість передачі даних, тому вони особливо підходять для сценаріїв із середньою та високою щільністю фіксованих сенсорів/актуаторів, а також для застосунків, які охоплюють широкі території та потребують джерел довготривалого живлення, зокрема Розумне місто [51].

3.3.3 Вибір Wi-Fi модуля

Для подальшого розвитку IoT і розширення області його застосування потрібні потужні, недорогі та енергоефективні рішення, що мають малий форм-фактор, тобто менший розмір і вагу. Кожен IoT пристрій складається з МК і модуля бездротової комутації (зазвичай Wi-Fi) або їх комбінації в одному. Велика різноманітність комунікаційних модулів вже представлена на ринку і широко використовується для проєктування та розробки пристроїв IoT, наприклад Xbee, CC32, ESP32, ESP8266, тощо.

Результати порівняння цих модулів наведено в таблиці 3.3 [52]. Всі модулі відповідають стандарту IEEE 802.11 b/g/n. Очевидною перевагою є найнижча вартість модуля ESP8266 з одночасним збалансованим набором параметрів та характеристик, тому саме цей модуль буде обрано в якості основи вузлів БМСА.

Таблиця 3.3 – Порівняльна характеристика комунікаційних модулів [52]

Модуль	ESP32 (ESP-WROOM-32)	ESP8266 (ESP8266-12E)	CC32 (CC3220MODSF)	Xbee (XB2B-WFPS-001)
Центральний процесор	Tensilica Xtensa LX6 32 біт Dual-Core, 160/240 МГц	Tensilica LX106 32 біт, 80 МГц - 160 МГц	ARM Cortex-M4, 80 МГц	-
Оперативна пам'ять	520 Кб	36 Кб	Ч'256 Кб	-
FLASH пам'ять	2Мб (max 64Мб)	4Мб (max 16Мб)	1Мб (max 32Мб)	-
Програмування	C, C++, Lua, та ін.	C, C++, Lua, та ін.	C (SimpleLink SDK)	AT та API команди
Відкритість	+	+	-	-
Bluetooth	4.2 BR/EDR + BLE	-	-	-
UART	3	2	2	1
GPIO	32	17	21	10
SPI	4	2	1	1
I2C	2	1	1	-
ШИМ	8	-	6	-
Розмір, мм	25.5 x 18.0 x 2.8	24.0 x 16.0 x 3.0	20.5 x 17.5 x 2.5	24.0 x 22.0 x 3.0
Вартість, грн	360	240	720	1040

ESP8266 – це серія МК із низьким енергоспоживанням на мікросхемі (SoC) із вбудованим модулем бездротового зв'язку (Wi-Fi). ESP8266 включає в себе вдосконалену версію 32-розрядного процесора Tensilica серії L106 Diamond, вбудований SRAM, 10-розрядний аналого-цифровий перетворювач, антенні перемикачі, RF-балун, підсилювач потужності, підсилювач з низьким рівнем шуму, фільтри, модулі керування живленням та інші периферійні пристрої [53]. ESP8266 може досягати надзвичайно низького енергоспоживання [54], тому він підходить для пристроїв, що живляться від батарейок, таких як носима електроніка та програми IoT.

На основі модуля Olimex MOD-WIFI-ESP8266-DEV [55] (рис. 3.5) можна створити удосконалені Wi-Fi модулі, що окрім чіпа ESP8266EX, також містять 2 Мб флеш-пам'яті SPI, інтегровану плату антени, отвори для пайки плати, а також невеликий роз'єм U.FL для підключення зовнішньої антени. Цей модуль може бути припаяний безпосередньо на інші плати, оскільки він не має компонентів на нижній стороні.



Рисунок 3.5 – Olimex MOD-WIFI-ESP8266-DEV [55]

3.3.4 Проєктування вузлів БМСА

Для реалізації БМСА потрібно створити декілька різних типів вузлів.

Першим вузлом Wi-Fi є вузол «температура/реле», призначений для керування різними нагрівальними елементами (наприклад, водонагрівачами, електричними радіаторами, інфрачервоними нагрівальними панелями тощо). Вузол містить модуль Olimex ESP8266-DEV, датчик температури DS18B20 у водонепроникному корпусі [56], 1-канальне реле 16A, перетворювач Vigortronix 220 V до 5V AC/DC (VTX-214-001-105) [57] та інші електронні компоненти. Цей вузол може працювати як вузол-сенсор, як вузол-актуатор або комбінація обох.

DS18B20 (рис. 3.6) – це цифровий датчик температури, який забезпечує вимірювання температури в діапазоні від $-55\text{ }^{\circ}\text{C}$ до $+125\text{ }^{\circ}\text{C}$ з точністю $\pm 0,5\text{ }^{\circ}\text{C}$.



Рисунок 3.6 – Зовнішній вигляд датчика DS18B20 [56]

DS18B20 підключається до МК ESP8266 за допомогою однієї провідної шини. Він дозволяє отримувати дані в вигляді цифрового сигналу з 12-бітовою роздільною здатністю по протоколу 1-Wire. Цей протокол дозволяє підключити велику кількість таких датчиків, використовуючи всього 1 цифровий порт контролера, і всього 2 дроти для всіх датчиків (земля і сигнал). У цьому випадку застосовується так зване «паразитне живлення», при якому

датчик отримує енергію прямо з лінії сигналу. Кожен датчик має унікальний прошитий на виробництві 64-бітний код. Завдяки цьому, кілька таких датчиків можна підключити до одного МК. В постійній пам'яті DS18B20 можна зберегти діапазон значень температури, при виході за який сенсор буде переходити в режим оповіщення. На загальній шині з багатьох сенсорів МК може за раз дізнатися, які з них перейшли в цей режим. Таким чином стає легко визначити проблемну ділянку в контрольованому середовищі [56].

Другий модуль - це Wi-Fi вузол «температура/вологість». Вузол містить модуль Olimex ESP8266-DEV, датчик температури та вологості DHT11 [58], перетворювач змінного/постійного струму та інші електронні компоненти.

Цифровий датчик DHT11 (рис. 3.7) включає компонент для вимірювання вологості резистивного типу, компонент вимірювання температури NTC і високоефективний 8-бітний МК [58].

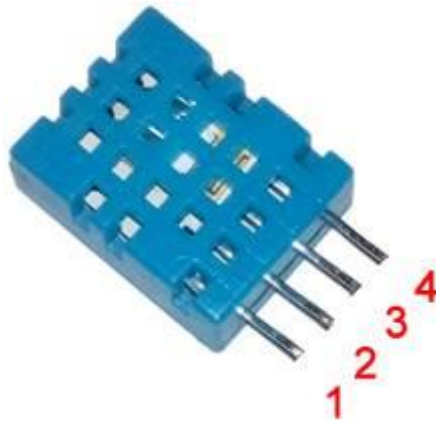
DHT11 цифровий датчик температури і вологості (рис. 3.7) є складовим датчиком, який містить калібрований цифровий вихідний сигнал з показаннями температури і вологості. Він має високу надійність і забезпечує довготривалу стабільну роботи. Його базові показники:

- для відносної вологості: діапазон: 20-95%, точність: $\pm 5\%$;
- для температури: діапазон: $0 \dots +50^{\circ}\text{C}$, точність: $\pm 2\%$;
- частота опитування не більше 1 Гц (не більше одного разу за 1 сек).

Для підключення до Arduino між виводами 1 і 2 ще потрібно включити резистор. Також необхідно скачати відповідну бібліотеку DHT.

Третій модуль – це вузол виявлення присутності (руху), який містить модуль Olimex ESP8266-DEV, пасивний інфрачервоний датчик (PIR) [59], перетворювач змінного/постійного струму та інші електронні компоненти.

Сенсор PIR використовується для вимірювання інфрачервоного (ІЧ) світла, що випромінюють об'єкти в діапазоні виявлення руху датчика, який можна налаштувати (рис.3.8) [59].



1 - Vcc – живлення +3,5 ~ 5,5 В; 2 - Data – передача даних;

3 - NC – не використовується; 4 - GND – заземлення.

Рисунок 3.7 – Зовнішній вигляд датчика DHT11 [58]

Оскільки ці сенсори дозволяють фіксувати рух (присутність), вони часто використовуються в системах сигналізації. Ці датчики малі за габаритами, недорогі, споживають мало енергії, легкі в експлуатації, практично не схильні до зносу. Основні технічні характеристики: зона роботи датчика: до 6 метрів; робоча напруга: 5 - 9В [59].



Рисунок 3.8 – Датчик руху [59]

PIR-sensor конструктивно поділено на дві половини. Це зумовлено тим, що для пристрою сигналізації важливо саме наявність руху в зоні чутливості, а не сам рівень випромінювання [59].

Принцип роботи датчика руху наступний: коли пристрій встановлений у порожній кімнаті, доза випромінювання, що отримується кожним елементом, постійна, як і напруга. З появою в кімнаті людини, вона насамперед потрапляє у зону огляду першого елемента, у якому утворюється позитивний електричний імпульс. Коли людина переміщається по кімнаті, разом із нею переміщається і теплове випромінювання, яке потрапляє вже на другий сенсор. Цей PIR-елемент генерує негативний імпульс. Різностямовані імпульси реєструються електронною схемою датчика, що робить висновок, що у полі зору датчика знаходиться людина [59].

Опис проєктованих вузлів БМСА зведено в таблицю 3.4.

Таблиця 3.4 – Опис проєктованих модулів БМСА

Вузол БМСА	Призначення	Компоненти
1	Керування нагрівальними елементами	Olimex ESP8266-DEV, датчик температури DS18B20 у водонепроникному корпусі, 1-канальне реле 16А, перетворювач Vigortronix 220 V до 5V AC/DC
2	Контроль мікроклімату	Olimex ESP8266-DEV, датчик температури та вологості DHT11, перетворювач Vigortronix 220 V до 5V AC/DC
3	Виявлення руху	Olimex ESP8266-DEV, пасивний інфрачервоний (PIR) сенсор, перетворювач Vigortronix 220 V до 5V AC/DC

3.4 Аналіз та вибір програмних засобів розробки

3.4.1 Вибір середовища розробки програми для Wi-Fi модуля

Існує багато середовищ розробки програмного забезпечення для МК, зокрема Arduino IDE, PlatformIO, Microsoft Visual Studio.

Arduino IDE – це крос-платформне інтегроване середовище розробки застосунків. Воно дозволяє легко писати код і завантажувати його до МК. Середовище написано на Java та базується на Processing та іншому програмному забезпеченні з відкритим кодом. Воно працює під керуванням ОС Windows, Mac OS X і Linux [60].

Arduino IDE – просте у використанні середовище, що підтримує всі плати Arduino, має вбудований менеджер бібліотек та відкритий код. Окрім оригінальної IDE Arduino компанія надає вебредактор Arduino, що працює онлайн і не потребує завантаження на комп'ютер. Але в Arduino IDE відсутні деякі функції, які є наприклад в Visual Studio. Такими наприклад є автозаповнення, показ параметрів функції, підказки щодо синтаксичних помилок тощо [62].

PlatformIO – це екосистема з відкритим кодом для розробки IoT застосунків. Характерні особливості: кросплатформна система побудови та менеджер бібліотек; безперервна інтеграція з IDE; готовність до хмарної компіляції [60].

PlatformIO – професійний інструмент розробки, який підходить для створення проєктів, пов'язаних із вбудованими системами. Цей інструмент розробки працює в операційних системах macOS, Windows і Linux, а також підтримує node.js, typescript і JavaScript. Функціональність платформи можна збільшити за допомогою розширень [61].

Він підтримує не лише плати Arduino, але й багато інших плат, таких як Raspberry, ESP32, ESP8266 та багато інших. Крім того, PlatformIO підтримує мультипроєкти, теми, керування бібліотеками. Він вимагає Python і має дві різні версії: одну створено на основі Atom, а іншу – як плагін коду Visual Studio [62].

Microsoft Visual Studio – це набір інструментів розробки ПЗ на основі компонентів та інших технологій для створення потужних високопродуктивних програм [60].

Microsoft Visual Studio – це платформа, яка працює на трьох операційних системах: Windows, macOS і Linux. Ця платформа надає редактори для C/C++, JavaScript, C#, HTML, CSS тощо. Щоб запрограмувати Arduino за допомогою Microsoft Visual Studio, потрібно встановити його розширення під назвою Visual Micro [61].

Це плагін для Microsoft Visual Studio, повністю сумісний з Arduino. Дозволяє легко та швидко писати код Arduino з підсвічуванням синтаксису, підказками про помилки коду тощо. Підтримує кілька плат Arduino, надає можливість використовувати бібліотеки Arduino за допомогою менеджера бібліотек [62].

Таблиця 3.4 – Порівняння середовищ розробки для Arduino [60] - [62]

Характеристика	Arduino IDE	Microsoft Visual Studio	PlatformIO
Простота	+	-	+
Підтримка інших плат	-	-	+
Платформа	Java	C/C++, JavaScript, C#	Python
Операційна система	OC Windows, Mac OS X і Linux	Windows, macOS і Linux	Windows, macOS, Linux, FreeBSD
Менеджер бібліотек	-	+	+
Підтримка плагінів та ін. інструментів	Processing	HTML, CSS	node.js, typescript, JavaScript
Наявність вебверсії	+	-	-
Додаткові функції	-	+	+

Таким чином, важливою перевагою Arduino IDE є наявність вебверсії, а також простий та зрозумілий інтерфейс, тому саме це середовище варто обрати для розробки.

3.4.2 Вибір платформи домашньої автоматизації

На сьогоднішній день існує цілий ряд доступних платформ домашньої автоматизації з відкритим кодом: OpenHAB, Home Assistant, Domoticz, Jeedom, Calaos та інші. В таблиці 3.5 наведено порівняння деяких найбільш відомих СДА [63]. Основна увага приділялася аналізу платформ OpenHAB та Domoticz, які є популярними на українському ринку.

Таблиця 3.5 – Порівняння платформ домашньої автоматизації [63]

Платформа	Мова	Веб-інтерфейс	Протоколи	Сервіс	API	Плагіни	Документація
OpenHAB	Java	+	+	MQTT	+	+ REST	+
Home-Assistant	Python 3	+	+	MQTT	REST/Python /Websocket APIs	+	+
Domoticz	C++	+	+	MQTT	+	+ JSON-based	+
Jeedom	PHP	+	+	MQTT	+ JSON RPC, HTTP-based	+	+ in French
Calaos	C++	+	+	-	+ JSON-based	-	+ in French

Domoticz [64] – це відкрита СДА, яка може бути встановлена на різних операційних системах (Linux, Apple і Windows) і розгорнута на декількох вбудованих системах, таких як Raspberry Pi або Synology NAS [65]. Domoticz дозволяє конфігурувати, контролювати та керувати різними пристроями, такими як світильники, перемикачі та різні датчики або лічильники. Вона має користувальницький інтерфейс HTML5 (рис. 3.9), який автоматично адаптується до настільних і мобільних пристроїв і сумісний з більшістю веббраузерів. На додаток до інтерфейсу веббраузера, Domoticz GUI можна також отримати за допомогою програми Domoticz на пристроях Android.

Додаткова інформація щодо СДА Domoticz доступна також у посібнику користувача [66].



Рисунок 3.9 – Графічний інтерфейс користувача Domoticz [64]

OpenHAB (Open Home Automation Bus) [67] – це відкрите програмне забезпечення для домашньої автоматизації [68], що створене з використанням Java. Завдяки архітектурі на основі плагінів, воно підтримує кілька технологій (Plugwise, EnOcean, Tellstick і т.д.), систем оповіщення (електронна пошта, Google Talk і т.д.) і багато пристроїв. OpenHAB використовує потужний і гнучкий механізм для розробки правил, з тригерами, сценаріями, діями, повідомленнями та голосовим управлінням на основі часу та різноманітних подій [67]. OpenHAB підтримується на Linux, MacOS, Windows, Raspberry Pi, PINE64, Docker, Synology та інших платформах. OpenHAB, який працює на спеціальному обладнанні, не вимагає інтеграції з хмарними службами, оскільки зберігає дані приватно і безпосередньо спілкується з локальними пристроями. OpenHAB складається з двох частин: OpenHAB Runtime і OpenHAB Designer. Runtime виконує процеси домашньої автоматизації, тоді як Designer представляє собою інструмент конфігурації для створення інтерфейсу користувача. Створений користувальницький інтерфейс (рис. 3.10) доступний за допомогою веб-браузера або openHAB Андроїд-застосунка.

OpenHAB є популярним програмним забезпеченням для домашньої автоматизації і використовувався в багатьох проєктах [1], [7], [22], [69].

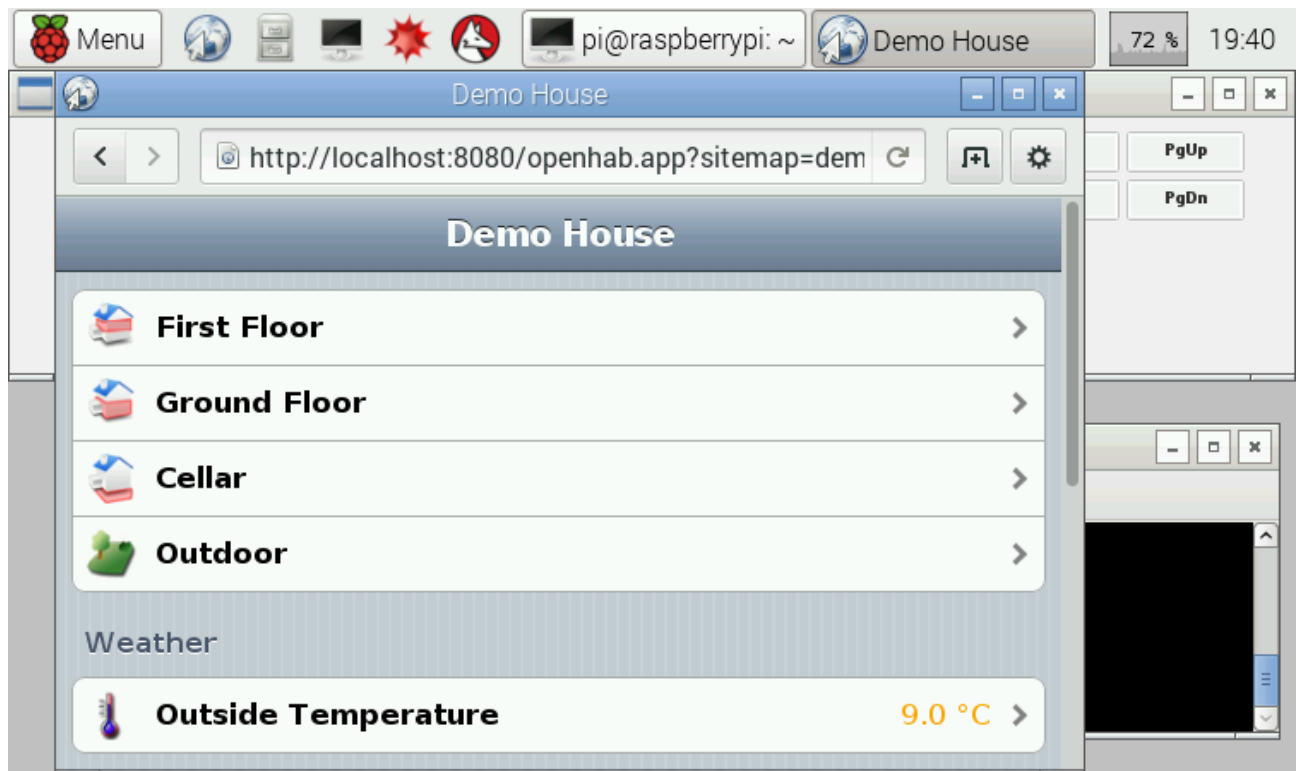


Рисунок 3.10 – Графічний інтерфейс користувача OpenHAB [67]

3.5 Висновки до розділу 3

В цьому розділі відповідно до мети та задач дипломної роботи виконано проєктування програмно-апаратного забезпечення БСМА: аналіз вимог, розробка архітектури системи, огляд та вибір інструментарію реалізації, вибір компонентів та проєктування вузлів БМСА.

4 КОНСТРУЮВАННЯ ПРОГРАМНО-АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ БМСА

4.1 Розробка алгоритму функціонування системи

На рисунку 4.1 наведено узагальнену схему алгоритму функціонування системи.



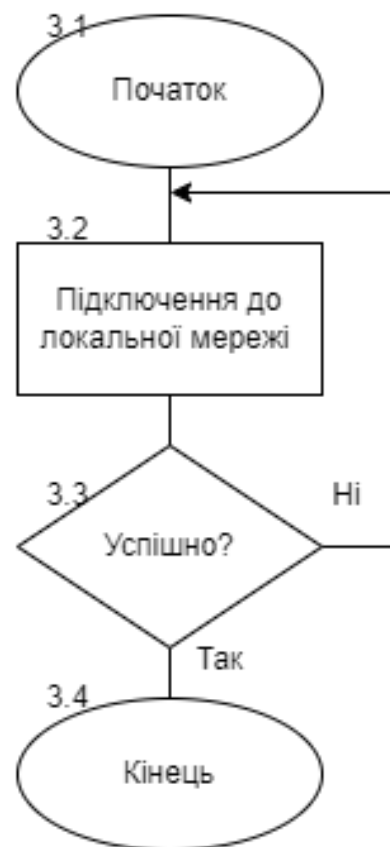
Рисунок 4.1 – Узагальнений алгоритм функціонування системи

Під час запуску система переходить у фазу ініціалізації, де МК ініціалізує локальні та глобальні змінні, модуль Wi-Fi, об'єкти MQTT (рис.4.2, а).

На наступному етапі МК підключається до локальної бездротової мережі і коли з'єднання встановлено, входить у нескінченний цикл (рис. 4.2, б). У цьому циклі МК періодично виконує наступні завдання: запитує та отримує дані від підключеного датчика(ів), обробляє ці дані, створює тему(и) MQTT і публікує тему(и) у MQTT брокер (рис. 4.3, а). Далі перевіряє, чи були отримані нові дані від MQTT брокера, обробляє нові дані, оновлює підключений(і) актуатор (и) і за потреби генерує тему відповіді (рис.4.3, б).



а



б

Рисунок 4.2 – Схема алгоритма процесу ініціалізації (а) та підключення до мережі (б)

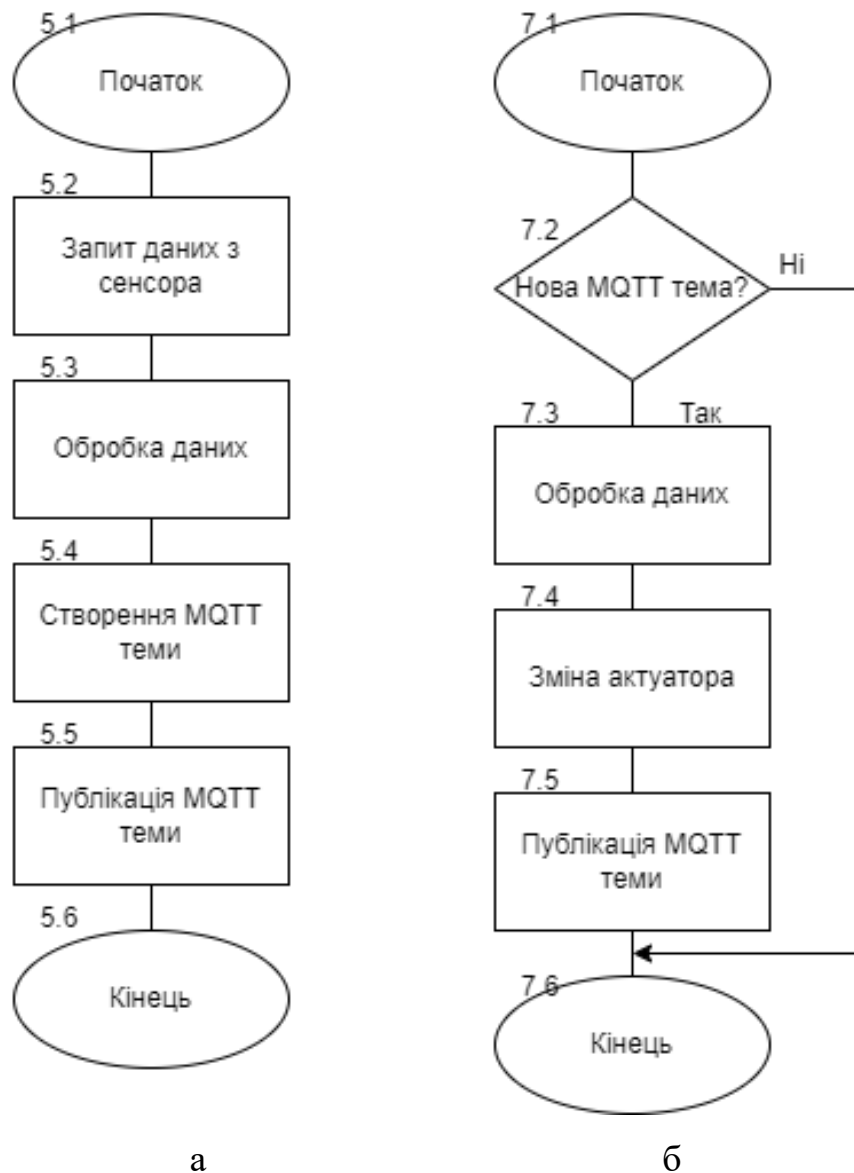


Рисунок 4.3 – Схема алгоритма процесу обробки даних сенсорів (а) та актуаторів (б)

4.2 Розробка MQTT тем

На рисунку 4.4 зображено схему взаємодії модулів БМСА з системою OpenHAB за протоколом MQTT, використовуючи брокер.

Розроблені вузли Wi-Fi отримують і генерують різні теми MQTT (таблиця 4.1). Вузол виявлення присутності (руху) публікує тему «md_node/out/m» про виявлення події руху. Вузол «температура/вологість» публікує виміряні дані температури («th_node/out/t») і вологості

(“th_node/out/h”) періодично (щохвилини). Вузол «температура/реле» публікує тему фактичної температури (“tr_node/out/t”) і отримує теми набору реле (“tr_node/in/r”) і запиту температури (“tr_node/in/t”). У темі запиту температури вузол Wi-Fi отримує та публікує фактичну температуру, а щодо отриманого значення в темі “tr_node/in/r” закриває або розмикає котушку реле.

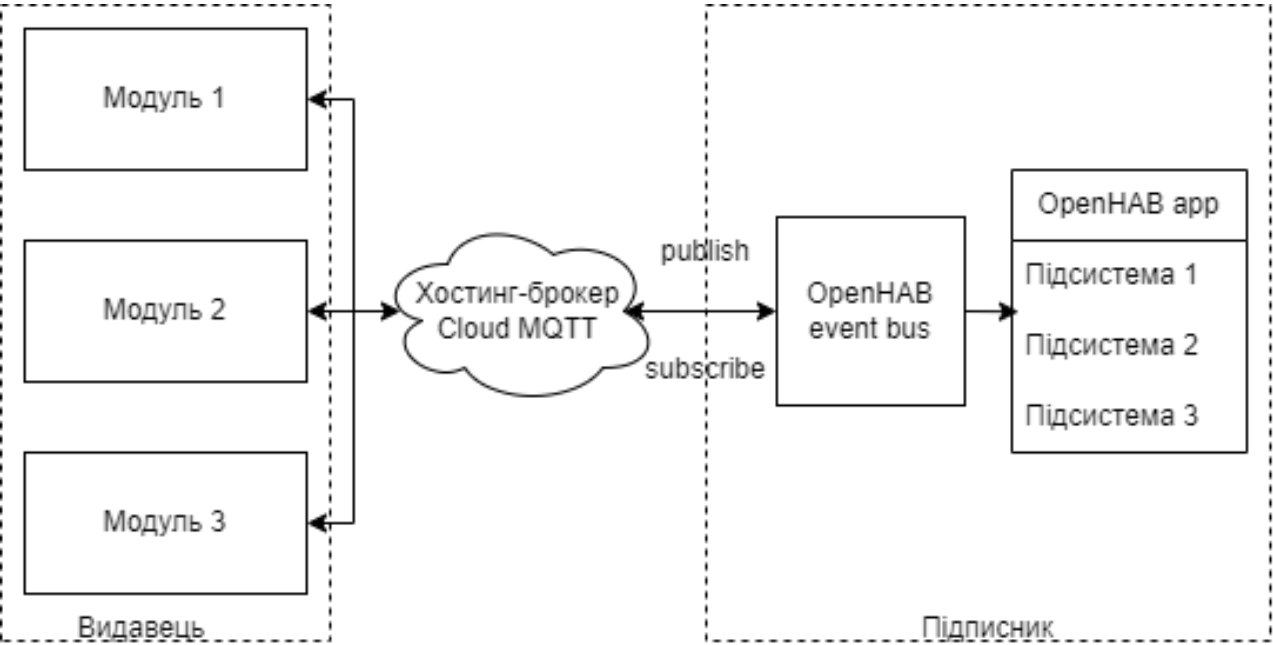


Рисунок 4.4 – Організація взаємодії компонентів системи

Таблиця 4.1 – Теми MQTT для вузлів БМСА

Вузол БМСА	MQTT тема	Видавець/ Підписник	Опис
1	tr_node/in/r	Підписник	Закрити/відкрити котушку реле
	tr_node/in/t	Підписник	Запит температури
	tr_node/out/t	Видавець	Вимірювання температури
	tr_node/out/s	Видавець	Статус реле
2	th_node/out/t	Видавець	Вимірювання температури
	th_node/out/h	Видавець	Вимірювання відносної вологості
3	md_node/out/m	Видавець	Визначення присутності (руху)

4.3 Розробка програмного забезпечення для модулів БМСА

Для організації функціонування модулів 1 та 2 розроблено програму опитування датчика температури. На рисунку 4.5 показано прототип модулю, створений в середовищі Tinkercad. У режимі симуляції необхідно натиснути на датчик та на області, що з'явилася, змінювати положення бігунка температури курсором миші, під час збільшення температури світлодіоди починають послідовно загорятись. Текст програми наведено в Додатку А.1.

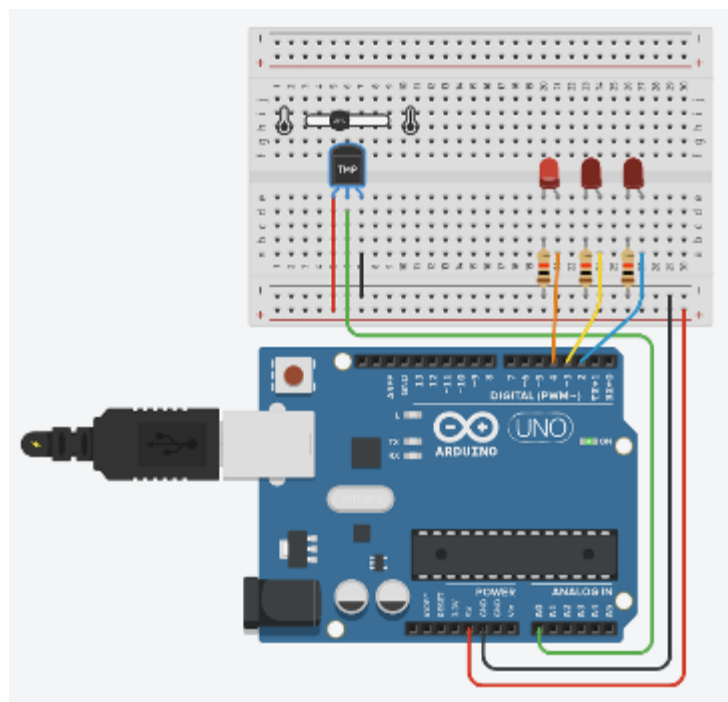


Рисунок 4.5 – Прототип модулю з датчиком температури

Для організації функціонування модуля 3 розроблено програму, що дозволяє працювати з датчиком виявлення руху/присутності. На рисунку 4.6 показано прототип модулю, створений в середовищі Tinkercad.

У режимі симуляції необхідно натиснути на датчик та на області, що з'явилася, змінювати положення об'єкта курсором миші, під час руху об'єкта у полі видимості датчика загоряється світлодіод. Текст програми наведено в Додатку А.2.

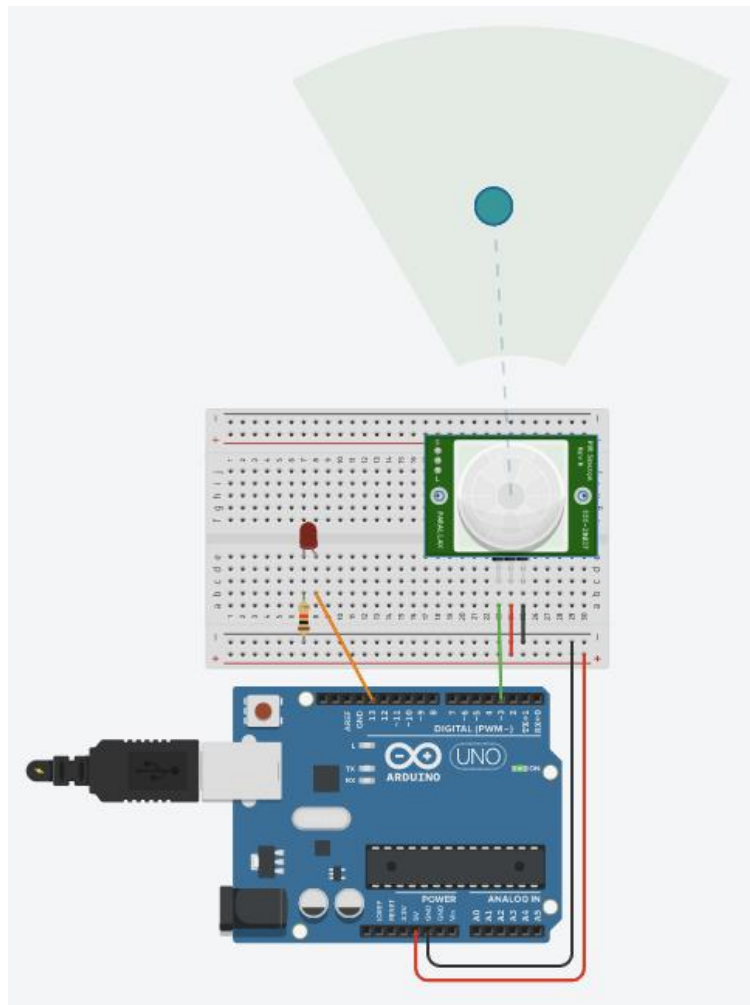


Рисунок 4.6 – Прототип модулю з датчиком руху

Для організації функціонування модулів інших типів розроблено програму, що дозволяє працювати з декількома датчиками. На рисунку 4.7 показано прототип такого модулю, створеного в середовищі Tinkercad з підключеними датчиками освітлення, рівня рідини, газу, руху, температури та вологості. Текст програми наведено в Додатку А.3.

Задля коректної роботи програми та отримання даних про температуру та вологість з датчику DHT11 потрібно підключити бібліотеку DHT.h та Adafruit_Sensor.h.

Після запуску програми починається ініціалізація змінних та портів вводу-виводу. Потім програма переходить в головний цикл у якому вона проводить опитування сенсорів. У таблиці 4.2 наведено опис функцій програми.

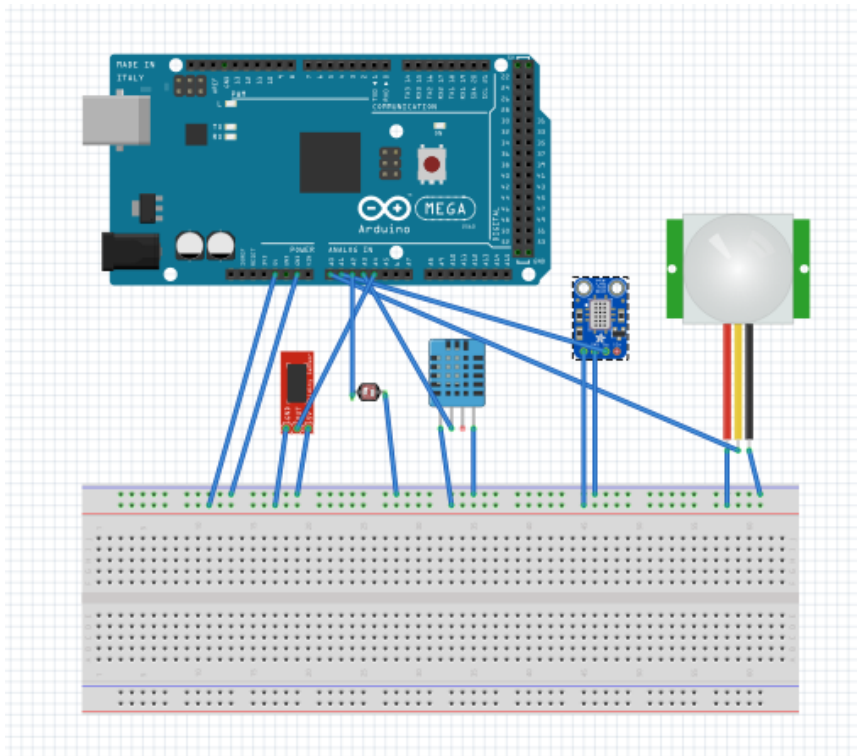


Рисунок 4.7 – Прототип модулю БМСА з групою датчиків

Таблиця 4.2 – Опис функцій програми

Назва функції	Опис функції
Serial.begin(9600)	Відкриває послідовний порт, встановлює швидкість 9600 біт/с
pinMode(ledPin, OUTPUT)	Встановлює режим роботи – вихід для 13 піна
dht.begin()	Запускає датчик вологості та температури за допомогою бібліотеки DHT.h
pinMode(A1, INPUT)	Встановлює режим роботи – вхід
Serial.print()	Передає дані через послідовний порт
dht.readHumidity()	Зчитує вологість
dht.readTemperature()	Зчитує температуру
analogRead(MQ2)	Зчитує значення з датчика газу
digitalWrite(13, HIGH)	Включає світлодіод
digitalWrite(13, LOW)	Вимикає світлодіод

4.4 Розгортання СДА Domoticz на БМСА

4.4.1 Конфігурування системи

Для інтеграції БМСА у системи домашньої автоматизації OpenHAB та Domoticz необхідно реалізувати процедури конфігурації, які дозволять обмінюватися даними між СДА та розробленими вузлами БМСА.

Розроблені вузли БМСА були підключені до середовища Domoticz за допомогою віртуального обладнання. Оскільки в Domoticz існує таке (віртуальне) обладнання, до нього можна приєднати кілька різних об'єктів за допомогою методу "Створити віртуальні прилади". Створений об'єкт може бути налаштований як перемикач, повзунок, дисплей, світло або інший тип. Створені (віртуальні) апаратні засоби відрізняються від інших за індексом ("Idx") (рис. 4.8), який присвоюється їм автоматично під час фази створення.

The image contains two screenshots of the Domoticz web interface. The top screenshot shows a list of virtual devices. The bottom screenshot shows a list of hardware devices.

Idx	Name	Enabled	Type	Address	Port	Data Timeout
18	room_hum	Yes	Dummy (Does nothing, use for virtual switches only) Create Virtual Sensors			Disabled
17	room_temp	Yes	Dummy (Does nothing, use for virtual switches only) Create Virtual Sensors			Disabled
16	broker_MQTT	Yes	MQTT Client Gateway with LAN interface	localhost	1883	Disabled
11	water_heater_set_temp	Yes	Dummy (Does nothing, use for virtual switches only) Create Virtual Sensors			Disabled
5	water_heater_actual_temp	Yes	Dummy (Does nothing, use for virtual switches only) Create Virtual Sensors			Disabled
2	motion_detector	Yes	Dummy (Does nothing, use for virtual switches only) Create Virtual Sensors			Disabled

Idx	Hardware	ID	Unit	Name	Type	SubType	Data	Last Seen
22	Water Heater - set temperature	0014056	1	Water Heater - set temperature	Thermostat	SetPoint	50.0	
26	Room Humidity	14064	1	Room Humidity	Humidity	LaCrosse TX3	Humidity 50 %	
24	Water Heater - actual temperature	14068	1	Water Heater - actual temperature	Temp	LaCrosse TX3	31.8 C	
25	Room Temperature	14069	1	Room Temperature	Temp	LaCrosse TX3	23.6 C	
1	Motion Detector	82001	1	Motion Detector	General	Alert	1	

Рисунок 4.8 – Список обладнання Domoticz

До графічного інтерфейсу Domoticz можна підключитися за допомогою веббраузера на Інтернет-адресу Raspberry Pi (<http://192.168.0.1:8080>) або за допомогою програми Domoticz на пристрої Android. Зазначена IP-адреса є адресою за замовчуванням Raspberry Pi, тоді як фактична адреса залежить від параметрів мережі.

Зв'язок між вузлами БМСА та Domoticz встановлено через брокер Mosquitto MQTT. Для того, щоб відправляти/отримувати теми до/від брокера MQTT, Domoticz має бути підписаний на цього брокера. Підписки виконуються шляхом додавання пристрою "MQTT Client Gateway with LAN interface" до середовища Domoticz. У діалоговому вікні конфігурації повинна бути вказана IP-адреса брокера, порт (за замовчуванням 1883) та інформація для входу брокера MQTT. Domoticz підтримує тільки дві теми MQTT, тобто "domoticz / in" для вхідних і "domoticz / out" для вихідних MQTT повідомлень. Отже, до Domoticz можна підключити лише один вузол БМСА з одним входом та/або одним виходом. Щоб подолати це обмеження, на Raspberry Pi було використано додаткову програму Node-RED. Використовуючи Node-RED, було досягнуто перетворення тем MQTT Domoticz в теми вузлів БМСА і навпаки.

4.4.2 Реалізація MQTT комунікації

Реалізований принцип комунікації MQTT між Domoticz, Node-RED і вузлами БМСА представлено на рисунку 4.9.

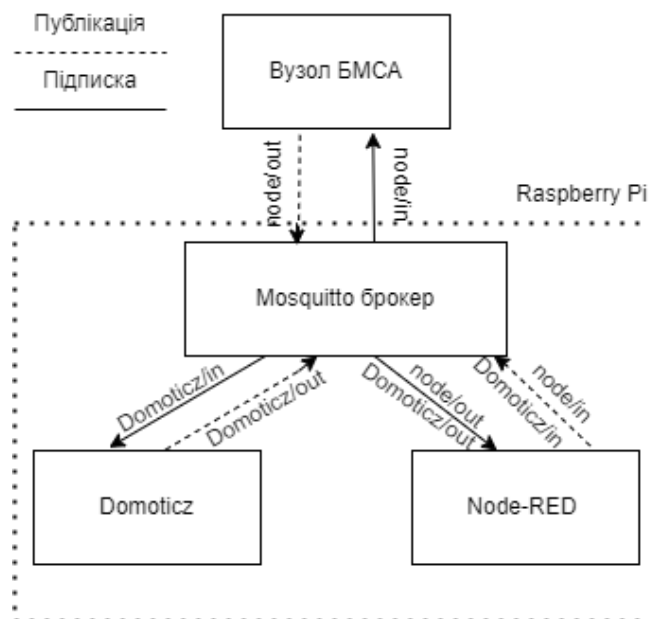


Рисунок 4.9 – Організація взаємодії між Domoticz, Node-RED та вузлами БМСА

При зміні стану об'єкта в середовищі Domoticz генерується і публікується тема «domoticz / out» програмним забезпеченням Domoticz. Приклад такої теми, у форматі JSON (пари ключів і значень), представлений на рисунку 4.10.

```
{ "Battery" : 255, "RSSI" : 12, "description" :
  "", "dtype" : "Temp", "id" : "82025", "idx" :
  25, "name" : "Room Temperature", "nvalue"
  : 0, "stype" : "LaCrosse TX3", "svalue1" :
  "23.8", "unit" : 1 }
```

Рисунок 4.10 – Приклад теми «domoticz / out»

Node-RED підписується на тему "domoticz / out", отримує цю тему від брокера MQTT і відповідно до значення індексу об'єкта ("idx"), генерує відповідну тему для вузла Wi-Fi ("node / in") і публікує її до брокера MQTT. Далі, вузол БМСА отримує тему "node / in" і виконує певні дії. У зворотному напрямку Node-RED створює тему "domoticz / in", відповідно до отриманої вхідної теми вузлів БМСА ("node / out") і публікує згенеровану тему брокеру MQTT. Оскільки Domoticz підписується на тему "domoticz / in", він оновлює стан об'єкта в середовищі Domoticz відповідно до отриманої теми.

4.4.3 Використання Node-RED

Node-RED [70] є інструментом програмування на основі потоків та складається з двох компонентів: вузлів введення, виведення і обробки, а також Потоків. Node-RED забезпечує редактор на основі браузера, що дозволяє легко створювати потоки, використовуючи широкий спектр вбудованих і доданих користувачем вузлів. Створений потік, що обробляє дані, керує пристроями або надсилає сповіщення, може легко розгортатися під час виконання. Крім того, Node-RED також дозволяє створювати графічний користувацький інтерфейс, використовуючи додатковий модуль Node-RED [71].

Node-RED приймає і відправляє (публікує) теми MQTT, використовуючи MQTT вхідні і MQTT вихідні вузли (рис. 4.11).



Рисунок 4.11 – Вихідний та вхідний вузли Domoticz в Node-RED

У діалоговому вікні конфігурації вузлів MQTT (рис. 4.12) повинні бути надані IP-адреса брокера, порт, назва теми та якість обслуговування (QoS).

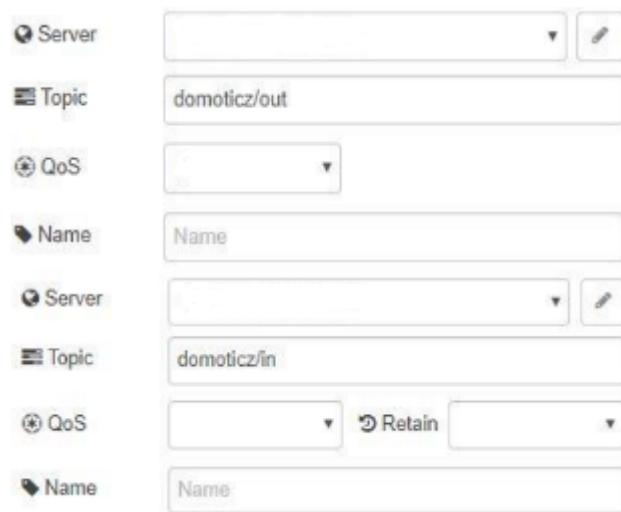


Рисунок 4.12 - Конфігурація вхідних і вихідних MQTT-вузлів у Node-RED

4.4.4 Реалізація алгоритму контролю

У Node-RED було реалізовано алгоритм, що дозволяє здійснювати автоматичне керування водонагрівачем на основі модуля 1 «температура/реле». Алгоритм отримує необхідні значення з інтерфейсу користувача Domoticz, і, виходячи з бажаних і фактичних значень, контролює температуру води, збільшуючи та зменшуючи вхідну потужність. Часовий графік подачі гарячої води, представлений на рис. 4.13, було реалізовано за допомогою налаштувань таймера об'єкта «Водонагрівач–задана температура».

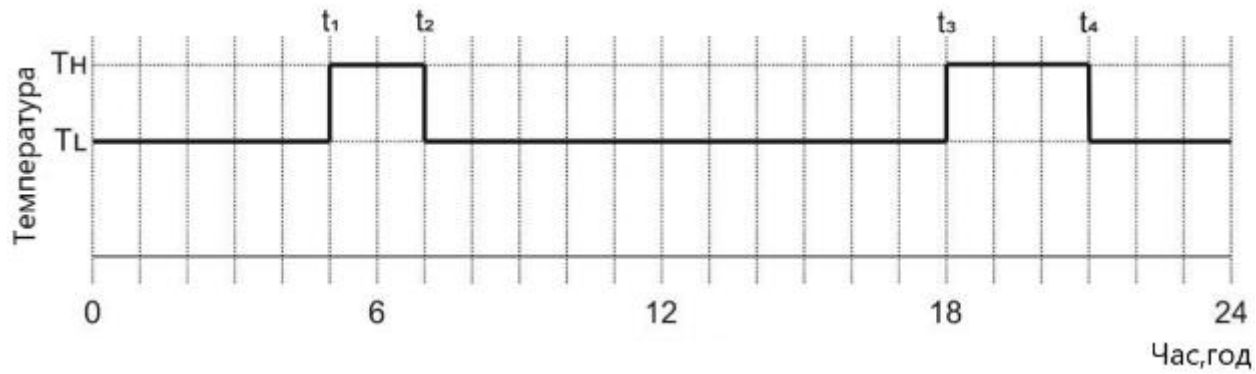


Рисунок 4.13 – Графік подачі горячої води для налаштування роботи вузла БМСА

Відповідно до цих параметрів (рис. 4.14) температура води встановлюється на рівні 60 °С між 5 і 7 годинами вранці та 6 і 9 годинами ввечері, а за межами цих двох періодів температура води повинна бути 40 °С.

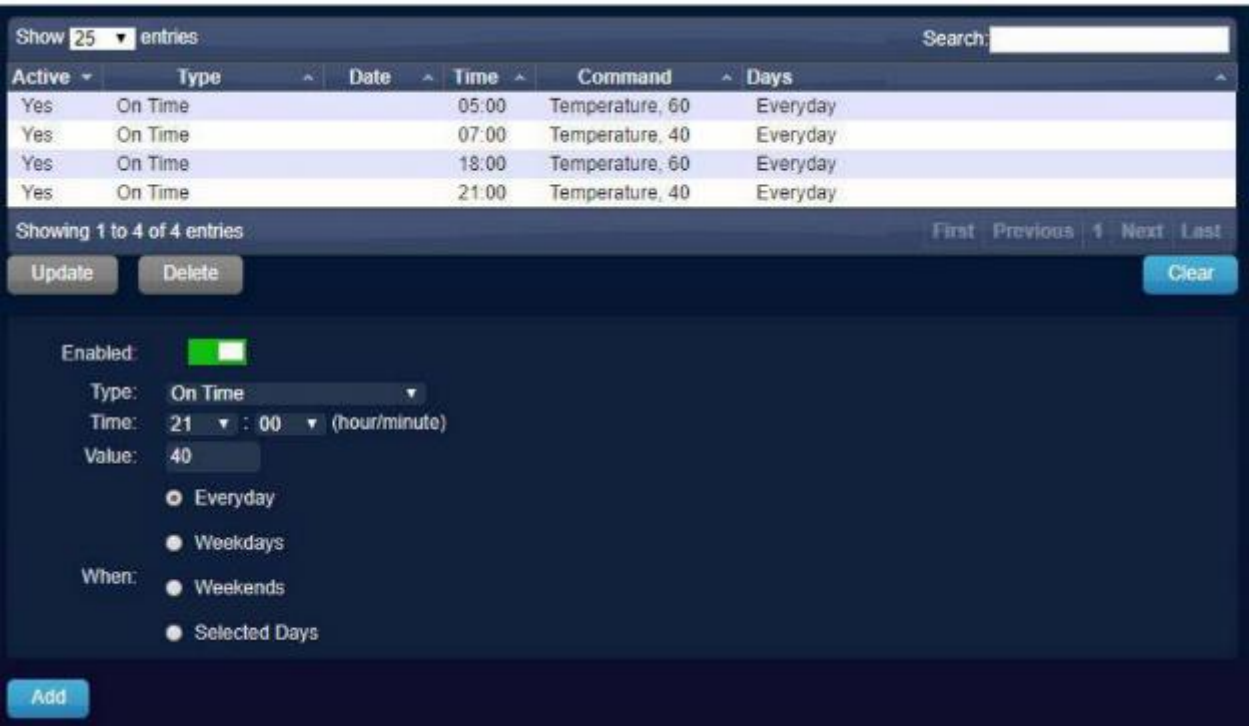


Рисунок 4.14 – Налаштування розкладу в Domoticz

Алгоритм Node-Red, згідно з яким виконує регулювання температури води працює наступним чином. При кожній події бажаної зміни температури води, Node-RED отримує цю температуру (ТН або ТЛ) з середовища Domoticz

через тему "domoticz / out" і зберігає в глобальну змінну Node-RED. Вузол "Inject Temp Request", періодично (кожну хвилину), ініціює потік, який генерує і публікує тему "tr_node / in / t". Коли вузол Wi-Fi отримує цю тему, він вимірює фактичну температуру води і публікує її за допомогою повідомлення "tr_node / out / t". Ця тема, у свою чергу, запускає потік Node-RED, який виконує простий контроль bang-bang (On / Off Controller), що вмикає/вимикає водонагрівач. На основі бажаної та фактичної температур і температурного гістерезису алгоритм розраховує, генерує та публікує тему "tr_node/in/r", яка вмикає/вимикає водонагрівач. Текст програми для функції Node-RED наведено в Додатку А.5.

Інтерфейс системи наведено на рисунку 4.15.



Рисунок 4.15 – Інтерфейс системи Domoticz

4.5 Розгортання СДА на платформі OpenHAB

4.5.1 Конфігурування системи

Зв'язок між вузлами openHAB і Wi-Fi також встановлюється через брокера MQTT. Щоб увімкнути MQTT-зв'язок, до середовища openHAB було додано додатковий компонент MQTT [72]. Прив'язка MQTT дозволяє openHAB діяти як MQTT клієнт, так що openHAB може відправляти/отримувати MQTT-повідомлення до/з брокера MQTT. Перелік створених файлів наведено в таблиці 4.3.

Таблиця 4.3 – Перелік створених файлів

Назва файлу	Призначення
mqtt.cfg	Налаштування параметрів прив'язки MQTT
.sitemap	Містить інформацію про компоненти GUI
.items	Опис типів даних у середовищі openHAB
.rules	Правила автоматизації процесів

Параметри прив'язки MQTT можна налаштувати у файлі mqtt.cfg. Інформація про URL та порт (за замовчуванням 1883) MQTT брокера обов'язково вказуються, в той час як інші параметри, такі як інформація для входу (ідентифікатор користувача та пароль), рівень якості обслуговування та інші параметри є необов'язковими.

Властивості та особливості графічного інтерфейсу користувача (GUI) openHAB задаються у файлах з розширеннями .sitemap та .items. Файл .sitemap містить інформацію про компоненти у графічному інтерфейсі. Один рядок визначення елемента .sitemap створює один відповідний елемент у GUI openHAB (рис. 4.16).

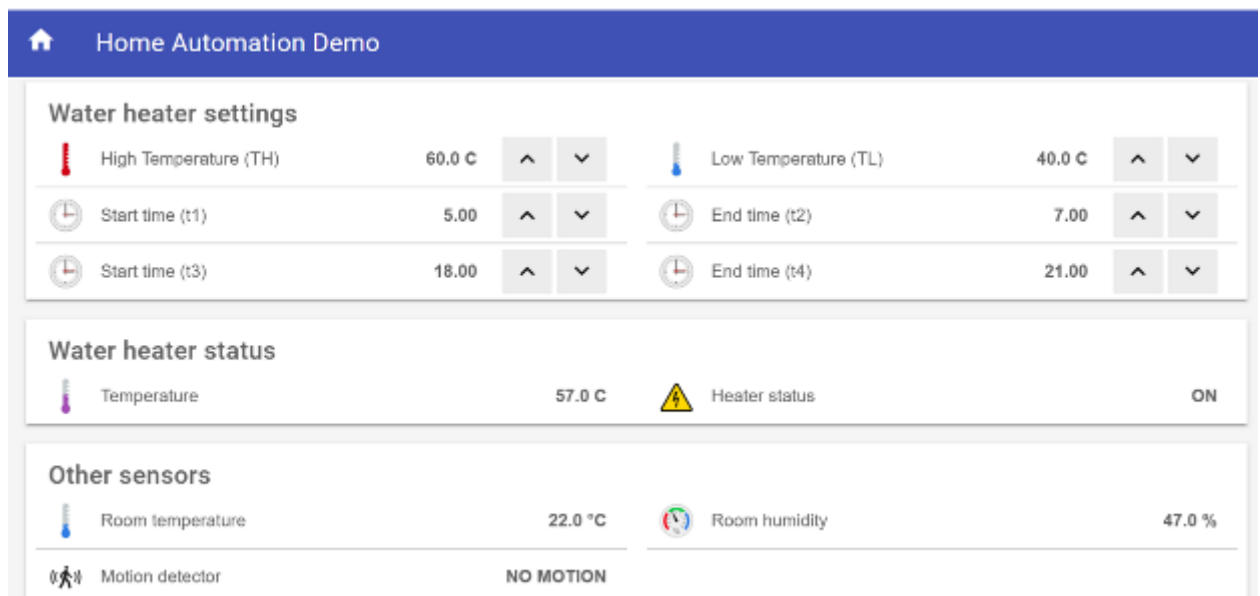


Рисунок 4.16 – Інтерфейс системи OpenHAB

Кожен елемент повинен бути певного типу, наприклад, Switch, Setpoint, Text або інші. Окремий елемент генерує піктограму, описовий текст і елементи стану та/або взаємодії в GUI OpenHAB. Використовуючи команду Frame, декілька елементів однієї теми розрізняються візуально на одній GUI сторінці.

Текст програми, що міститься в файлі .sitemap наведено в Додатку А.5.

Елементи, які зберігаються у файлі .items, представляють собою основні типи даних у середовищі openHAB. Елементи можуть бути прочитані або записані, а також можуть бути пов'язані з каналом прив'язки. Кожен рядок файлу .item являє собою визначення елемента, який включає:

- тип елемента (номер, рядок тощо);
- назву елемента;
- мітку елемента та формат відображення;
- піктограму елемента;
- додаткову конфігурацію зв'язування (між фігурними дужками).

Текст програми, що міститься в файлі .items наведено в Додатку А.6.

Правила, які використовуються для автоматизації процесів OpenHAB, розміщуються у файлі .rules. Кожне правило може бути визване (за часом або подією) і ця подія викликає скрипт, який виконує певні завдання (увімкнення/вимкнення нагрівача, виконання математичних обчислень і т.д.).

Файл правил – це текстовий файл з наступною структурою:

- імпорт;
- декларація змінних;
- правила.

Розділ імпорту містить оператори імпорту, розділ декларації змінних використовується для оголошення різних змінних, тоді як розділ правил містить список правил. Кожне правило має мати унікальне ім'я, умови запуску та блок сценаріїв, які виконуються при виконанні умов запуску (Додаток А.7).

4.5.2 Реалізація алгоритму контролю

У файлі .rules реалізовано алгоритм, що дозволяє здійснювати ручне і автоматичне керування водонагрівачем. Алгоритм отримує бажані значення з інтерфейсу користувача openHAB (рис. 4.17), і, виходячи з бажаних і фактичних значень, контролює температуру води, збільшуючи та зменшуючи вхідну потужність. Згідно з налаштуваннями графічного інтерфейсу і графіком (рис. 4.13), температура води встановлюється на рівні 60 °C між 5 і 7 годинами вранці та 6 і 9 годинами ввечері, а за межами цих двох періодів температура води повинна бути 40 °C. Алгоритм отримує бажані температури (T_H і T_L) і значення часу (t_1 , t_2 , t_3 і t_4) з GUI OpenHAB, а фактичну температуру з вузла БМСА (тема “tr_node / out / T”). Відповідно до поточних часових інтервалів (визначаються t_1 , t_2 , t_3 і t_4), алгоритм визначає необхідну температуру. Виходячи з бажаних і фактичних температур і температурного гістерезису, алгоритм генерує і публікує тему «tr_node / in / r», яка вмикає/вимикає водонагрівач. Публікація тем MQTT реалізується за допомогою командного рядка (команда executeCommandLine). Текст програми файлу .rules наведено в Додатку А.8.

4.6 Висновки до розділу 4

В даному розділі відповідно до мети та завдань роботи розроблено програмне забезпечення БМСА для СДА. В якості платформи СДА використано ПЗ з відкритим вихідним кодом (Domoticz та OpenHAB). Вузли БМСА реалізовано на основі Wi-Fi модулів. Оскільки використано Wi-Fi технологію, немає необхідності в додаткових апаратних засобах зв'язку (наприклад, ZigBee або Z-wave). У більшості випадків також немає потреби в додатковому маршрутизаторі Wi-Fi, оскільки багато вітчизняних користувачів використовують кабельні модеми з вбудованою точкою доступу Wi-Fi.

Хоча розроблені рішення тестувалися тільки з програмним забезпеченням Domoticz і OpenHAB, вони можуть зв'язуватися з іншим ПЗ для домашньої автоматизації, яке підтримує протокол обміну повідомленнями MQTT.

ВИСНОВКИ

В результаті виконання дипломної роботи виконано дослідження технологій домашньої автоматизації, а також структурних та функціональних особливостей сучасних БМСА.

Виконано аналіз вимог до БМСА, вибір її апаратного забезпечення, а також програмних середовищ розробки. Розробка базується на програмному забезпеченні для домашньої автоматизації з відкритим вихідним кодом (openHAB та Domoticz) та апаратному забезпеченні - вузлах сенсорів/актуаторів на основі Wi-Fi модуля ESP8266.

Розроблене програмно-апаратне забезпечення має ієрархічну структуру і складається з центрального комп'ютера (Raspberry Pi 3) та трьох вузлів БМСА. Програмне забезпечення домашньої автоматизації з відкритим вихідним кодом (OpenHAB або Domoticz), що розгорнуте на центральному комп'ютері, здійснює зв'язок з вузлами БМСА через локальну мережу Wi-Fi, використовуючи протокол MQTT. Центральний комп'ютер отримує дані з датчиків на вузлах БМСА, виконує алгоритми обробки даних, а також передає дані на виконавчі пристрої на підключених вузлах БМСА. Впровадження представленого рішення дозволить користувачу контролювати показники мікроклімату, статуси обладнання, дистанційно керувати електроприладами за допомогою різних типів пристроїв (смартфон, ноутбук).

Представлене рішення дозволяє контролювати процеси в СДА, використовуючи тільки локальну мережу. Доступ до мережі Інтернет та хмарних сервісів можливо реалізувати додатково.

Енергоефективний контроль побутової техніки може бути досягнутий за допомогою представлених апаратних і програмних рішень разом з відповідними алгоритмами планування, які вмикають/вимикають електричні прилади, особливо нагрівальні елементи (наприклад, водонагрівачі, електричні радіатори тощо).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Bhatt, A. Cost effective digitization of home appliances for home automation with low-power Wi-Fi devices /A. Bhatt, J. Patoliya // Advances in electrical & electronics, information, communication & bio informatics (AEEICB 2016): 2nd International conference, Chennai, India, 27-28 February 2016: proceedings. – Los Alamitos: IEEE, 2016. – P. 643-648.
2. Smart Android Based Home Automation System Using Internet of Things (IoT) / [M.A. Khan, I. Ahmad, A.N. Nordin et al.] // Sustainability. – 2022. - Vol. 14, № 17. – P.1-17.
3. Smart homes that monitor breathing and heart rate / [F. Adib, H. Z. Mao, Z. Kabelac et al.] // Human factors in computing systems: 33rd Annual conference, Seoul, Republic of Korea, 18 - 23 April 2015: proceedings, 2015. – P.1-10.
4. MQTT in AAL systems for home monitoring of people with dementia // [A. Del Campo, E. Gambi, L. Montanini et al.] // Personal, indoor, and mobile radio communications: 27th Annual international symposium, Valencia, Spain, 4-8 September 2016: proceedings. – Los Alamitos: IEEE, 2016. – P.1-6.
5. Population projections in EU [Electronic resource]. – Access mode: <https://ec.europa.eu/eurostat/statistics-explained/index.php?oldid=497115>.
6. Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects / [F. Portet, M. Vacher, C. Golanski et al.] // Personal and ubiquitous computing. - 2013.- Vol. 17, № 1.- P. 127-144.
7. An OS for Internet of everything: early experience from a Smart home prototype / [J. Cao, L. Xu, A. Raef et al.] // ZTE Communications. – 2017. - Vol. 15, № 4. - P. 12-22.
8. Davidovic, B. A Smart home system based on sensor technology/ B. Davidovic, A. Labus // Electronics and energetics, Facta universitatis-series. – 2016. - Vol. 29, № 3. - P. 451-460.

9. Rana, M. Smart homes for a better living using bluetooth communication based on ATmega microcontroller/ M. Rana, R. Singh // International journal of research in engineering and technology. – 2014. - Vol. 3, № 6. – P. 210-213.
10. Vujovic, V. Raspberry Pi as a sensor web node for home automation / V. Vujovic, M. Maksimovic // Computers & electrical engineering. – 2015. - Vol. 44. - P. 153-171.
11. Mowad, M. A. E.-L. Smart home automated control system using android application and microcontroller,"/ M. A. E.-L. Mowad, A. Fathy, A. Hafez // International journal of scientific & engineering research. – 2014. - Vol. 5, № 5. - P. 935-939.
12. Kodali, R. K. MQTT based Home automation system using ESP8266 / R. K. Kodali, S. Soratkal // IEEE Region 10 Humanitarian technology conference, Agra, India, 21-23 December 2016: proceedings. – Los Alamitos: IEEE, 2016. – P.1-5.
13. Eclipse Mosquitto. An open source MQTT broker [Electronic resource]. – Access mode: <https://mosquitto.org>.
14. Kodali, R. K. A low cost implementation of MQTT using ESP8266 / R. K. Kodali, K. S. Mahesh // Contemporary Computing and Informatics: 2nd International conference, Noida, India, 14-17 December 2016: proceedings. – Los Alamitos: IEEE, 2016. - P. 404-408.
15. The Internet of things for everyone [Electronic resource]. – Access mode: <https://io.adafruit.com/>.
16. Kodali, R. K. Low cost implementation of Smart home automation / R. K. Kodali, K. S. Mahesh // Advances in computing, communications and informatics: International conference, Udupi, Karnataka, India, 13-16 September: proceedings. – Los Alamitos: IEEE, 2017. – P. 461-466.
17. Efficient Smart emergency response system for fire hazards using IoT/ [L. P. Maguluri, T. Srinivasarao, R. Ragupathy et al.] // International journal of Advanced computer science and applications. – 2018. - Vol. 9, № 1. - P. 314-320.

18. Marques, G. An indoor monitoring system for ambient assisted living based on Internet of Things architecture / G. Marques, R. Pitarma// International journal of Environmental research and public health. – 2016. - Vol. 13, № 11. – P. 1-14.
19. A low cost wireless acoustic sensor for ambient assisted living systems / [M. A. Quintana-Suarez, D. Sanchez-Rodriguez, I. AlonsoGonzalez et al.] // Applied sciences-Basel. – 2017. - Vol. 7, № 9. – P. 1-15.
20. An open source "Smart Lamp" for the optimization of plant systems and thermal comfort of offices / [F. Salamone, L. Belussi, L. Danza et al.] // Sensors. – 2016. - Vol. 16, № 3. – P. 1-12.
21. Design and development of a nearable wireless system to control indoor air quality and indoor lighting quality / [F. Salamone, L. Belussi, L. Danza et al.] //Sensors. – 2017. - Vol. 17, № 5. – P. 1-17.
22. Smart fall: accelerometer-based fall detection in a Smart home environment / [D. Sprute, A. Portner, A. Weinitschke et al.] // Inclusive Smart cities and eHealth: 13th International conference on Smart homes and health telematics, Geneva, Switzerland, 10-12 June 2015: proceedings, 2015. - P. 1-13.
23. Dey, S. Home automation using Internet of Thing / S. Dey, A. Roy, S. Das // Ubiquitous computing, electronics & mobile communication: 7th Annual conference, New York City, New York, USA, 20-22 October 2016: proceedings, 2016. – P. 1-6.
24. A visual analytics approach for detecting and understanding anomalous resident behaviors in Smart healthcare / [Z. Liao, L. Kong, X. Wang et al.] // Applied sciences-Basel. – 2017. - Vol. 7, № 3. – P. 1-13.
25. Dawadi, P. N. Automated Ccgnitive health assessment from Smart home based behavior data / P.N. Dawadi, D.J. Cook, M. Schmitter-Edgecombe // Journal of biomedical and health informatics. – 2016. - Vol. 20, № 4. -P. 1188-1194.
26. Suryadevara, N. K. Wireless sensor network based home monitoring system for wellness determination of elderly / N.K. Suryadevara, S.C. Mukhopadhyay// Sensors journal. – 2012. - Vol. 12, № 6. -P. 1965-1972.

27. Collotta, M. A solution based on bluetooth low energy for smart home energy management / M. Collotta, G. Pau // *Energies*. – 2015. - Vol. 8, № 10. - P. 11916-11938.
28. Hernandez, C. Predictive model for detecting MQ2 gases using fuzzy logic on IoT devices / C. Hernandez, S. Villagran, P. Gaona // *Engineering applications of neural networks* / Eds. C. Jayne, L. Iliadis. – Cham: Springer, 2016. - P. 176-185 (*Communications in computer and information science*, vol 629).
29. Espressif Systems. ESP8266 [Electronic resource]. – Access mode: <https://www.espressif.com/en/products/hardware/esp8266ex/overview>.
30. Minaie, A. Wireless sensor networks in Smart and secure homes /A. Minaie, R. Sanati-Mehrizi, K. J. Wheelhouse// *Engineering Education: Annual American society conference & exposition*, Florida, USA, 15 – 19 June 2019: proceedings, 2019. – P.1-12.
31. Integration of wireless sensor networks in the computer science and engineering curricula / [A. Minaie, A. Sanati-Mehrizi, R. Sanati-Mehrizi et al.] // *Engineering Education: Annual American society conference & exposition*, San Antonio, Texas, 10 – 13 June 2012: proceedings, 2012. – P.1-12.
32. Wireless sensor networks and applications / Eds. Y. Li, T. My, W. Weili. - New York: Springer, 2008. - 444 p.
33. Akyildiz, I. F. Wireless sensor networks / I. F. Akyildiz, M. C. Vuran. – Wiley, 2010. – 520 p.
34. Elshafee, A. M. Design and implementation of a Wi-Fi based Home automation system / A. M. Elshafee, K.A. Hamed // *World academy of science, engineering and technology*.- 2012. – Vol. 6. – P. 185-191.
35. Ferro, E. Bluetooth and Wi-Fi wireless protocols: a survey and a comparison / E. Ferro, F. Potorti // *IEEE Wireless communication*, 2005. - Vol. 12, № 1 - P. 12-16.
36. Comparison of IEEE 802.11e and IEEE 802.15.3 MAC / [X. Wang, Y. Ren, J. Zhao et al.] // *Emerging technologies: frontiers of mobile and wireless*

communication: 6th Circuits and systems symposium, Shanghai, China, 31 May 2004 - 02 June 2004: proceedings, 2004. - P. 675-680.

37. Baker, N. ZigBee and Bluetooth: strengths and weaknesses for industrial applications/ N. Baker // IEEE Computing & control engineering, 2005. - Vol. 16, № 2. – P. 20-25.

38. Porcino, D. Ultra-wideband radio technology: potential and challenges ahead / D. Porcino, W. Hirt // IEEE Communication Magazine, 2003. - Vol. 41, № 7. - P. 66-74.

39. Lee, J. S. Performance evaluation of IEEE 802.15.4 for low-rate wireless personal area networks/ J. S. Lee // Transactions on consumer electronics, 2006. - Vol. 52, № 3. - P. 742-749.

40. Lee, J. S. ITRI ZBnode: a ZigBee/ IEEE 802.15.4 platform for wireless sensor networks / J. S. Lee, Y. C. Huang // Systems, man & cybernetics: International conference, Taipei, Taiwan, 8-11 October 2006: proceedings, 2006.- P. 1462-1467.

41. Lee, J. S. A comparative study of wireless protocols: Bluetooth, UWB, ZigBee and Wi-Fi / J. S. Lee, Y. W. Su, C. C. Shen // 33rd Annual conference of the IEEE industrial electronics society, Taipei, Taiwan, 05-08 November 2007: proceedings, 2007. - P. 46-51.

42. OASIS Open. MQTT Version 3.1.1 [Electronic resource]. – Access mode: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>.

43. The Edison is not a Raspberry Pi [Electronic resource]. – Access mode: <https://www.sparkfun.com/news/1603>.

44. Smart fog: fog computing framework for unsupervised clustering analytics in wearable Internet of Things / [D. Borthakur, H. Dubey, N. Constant et al.] // Signal and information processing: IEEE Global conference, Montreal, QC, Canada, 2017, 14-16 November: proceedings, 2017. - P. 472-476.

45. The Raspberry Pi foundation. Raspberry Pi 3 model B [Electronic resource]. – Access mode: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.

46. Deschambault, O. Efficient implementation of the MQTT protocol for embedded systems / O. Deschambault, A. Gherbi, C. Legare // Journal of Information processing systems, 2017. - Vol. 13, № 1. - P. 26-39.
47. Foster, A. Messaging technologies for the industrial Internet and the Internet of Things whitepaper: a comparison between DDS, AMQP, MQTT, JMS and REST: PrismTech corporation [Electronic resource] / A. Foster. – Access mode: <http://www.prismtech.com/sites/default/files/documents/Messaging-Whitepaper-v1-7.pdf>.
48. Pereira, C. Towards efficient mobile M2M communications: survey and open challenges/ C. Pereira, A. Aguiar// Sensors, 2014. - Vol. 14, № 10. - P. 19582-19608.
49. Smart CEI moncloa: an IoT-based platform for people flow and environmental monitoring on a Smart university campus/ [M. Alvarez-Campana, G. Lopez, E. Vazquez et al.] // Sensors, 2017. - Vol. 17, № 12. – P. 1-24.
50. Paving the road toward Smart grids through large-scale advanced metering infrastructures / [G. Lopez, J.I. Moreno, H. Amarís et al.] // Electric power systems research, 2015. – Vol. 120. - P. 194–205.
51. Raza, U. Low power wide area networks: an overview / U. Raza, P. Kulkarni, M. Sooriyabandara // IEEE Commun. Surv. Tutor., 2017. -Vol. 19. - P. - 855–873.
52. Maier, A. Comparative analysis and practical implementation of the ESP32 microcontroller module for the Internet of Things /A. Maier, A. Sharp, Y. Vagapov// Internet technologies and applications: International conference, Wrexham, UK, 12-15 September 2017: proceedings, 2017. - P. 143-148.
53. Espressif Systems. ESP8266 Datasheet [Electronic resource]. – Access mode: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.
54. Espressif_Kelly. ESP8266 Low power solutions. [Electronic resource]. – Access mode: https://www.espressif.com/sites/default/files/9b-esp8266-low_power_solutions_en_0.pdf.

55. Olimex Ltd. MOD-WIFI-ESP8266-DEV module. [Electronic resource]. – Access mode: <https://eu.mouser.com/ProductDetail/Olimex-Ltd/MOD-WIFI-ESP8266-DEV?qs=QGk6feVlqMIuCsvdBN17og%3D%3D>.
56. Maxim Integrated. DS18B20 - Programmable resolution 1-Wire digital thermometer. [Electronic resource]. – Access mode: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.
57. Vigortronix Ltd. 1 Watt AC-DC Converters. [Electronic resource]. – Access mode: <https://www.vigortronix.com/product/1-watt-ac-dc-converter-1-series/>.
58. Adafruit Industries LLC. DHT11, DHT22 and AM2302 Sensors. [Electronic resource]. – Access mode: <https://learn.adafruit.com/dht/overview>.
59. ІЧ датчик руху для Arduino HC-SR501 [Електронний ресурс]. – Режим доступу: <https://arduino.ua/prod193-ik-datchik-dvijeniya-dlya-arduino-hc-sr501>.
60. Arduino IDE vs Light table [Electronic resource]. – Access mode: <https://www.stackshare.io/stackups/arduino-ide-vs-light-table>.
61. Best alternatives of Arduino IDE [Electronic resource]. – Access mode: <https://linuxhint.com/arduino/>.
62. 10 Arduino IDE alternative to start programming [Electronic resource]. – Access mode: <https://www.javacodegeeks.com/2018/08/10-arduino-ide-alternative.html>.
63. Design, implementation and practical evaluation of an IoT Home automation system for fog computing applications based on MQTT and ZigBee-WiFi sensor nodes/ [I. Froiz-Míguez, T.M. Fernandes-Carames, P. Fraga-Lamas et al.] // Sensors, 2018. – Vol. 18, № 8. – P.1-42.
64. Gizmocuz and domoticz community. Domoticz website. [Electronic resource]. – Access mode: <https://www.domoticz.com>.
65. Gizmocuz and domoticz community. Domoticz GitHub repository. [Electronic resource]. – Access mode: <https://github.com/domoticz/domoticz>.

66. Peters, R. E. Domoticz manual. [Electronic resource]/ R. E. Peters. – Access mode: <https://www.domoticz.com/DomoticzManual.pdf>.
67. openHAB Foundation. openHab Website. [Electronic resource]. – Access mode: <https://www.openhab.org>.
68. openHAB Foundation. openHAB GitHub repository. [Electronic resource]. – Access mode: <https://github.com/openhab>.
69. Kim, J.-W. Design and implementation of Smart home based on openHAB/ J.-W. Kim, Y.-J. Kim, // Journal of the Korea society of computer and information, 2017. - vol. 22, № 12. - P. 1-7.
70. JS Foundation. Node-RED website. [Electronic resource]. – Access mode: <https://nodered.org/>.
71. JS Foundation. Node-RED dashboard. [Electronic resource]. – Access mode: <https://flows.nodered.org/node/node-red-dashboard>.
72. openHAB Foundation. openHab MQTT binding. [Electronic resource]. – Access mode: <https://www.openhab.org/addons/bindings/mqtt1/>.

ДОДАТОК А
Текст програми

A.1 Текст програми для роботи з датчиком температури

```
const int Temp1 = 40;
const int Temp2 = 80;
double temp = 0;
void setup()
{
    pinMode(A0, INPUT);
    Serial.begin(9600);
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
}
void loop()
{
    int sensorValue = analogRead(A0);
    // measure temperature in Celsius
    temp = (double) sensorValue / 1024; //find percentage
of input reading
    temp = temp * 5; //multiply by 5V to get voltage
    temp = temp - 0.5; //Subtract the offset
    temp = temp * 100; //Convert to degrees
    Serial.print(temp);
    Serial.println(" C");
    if (temp < Temp1) {
        digitalWrite(2, 0);
        digitalWrite(3, 0);
        digitalWrite(4, 255);
    }
    if (temp >= Temp1 && temp < Temp2) {
        digitalWrite(2, 0);
```

```

    digitalWrite(3, 255);
    digitalWrite(4, 255);
}
if (temp >= Temp2) {
    digitalWrite(2, 255);
    digitalWrite(3, 255);
    digitalWrite(4, 255);
}
delay(500); // Wait for 500 millisecond(s)
}

```

A.2 Текст програми для роботи з датчиком руху

```

int sensorState = 0;
int sensorPin = 3;
int ledPin = 13;
void setup()
{
    pinMode(sensorPin, INPUT);
    pinMode(ledPin, OUTPUT);
    Serial.begin(9600);
}
void loop()
{
    // read the state of the sensor/digital input
    sensorState = digitalRead(sensorPin);
    // check if sensor pin is HIGH. if it is, set the
    // LED on.
    if (sensorState == HIGH)
    {

```

```

    digitalWrite(ledPin, HIGH);
    Serial.println("Sensor activated!");
}
else
{
    digitalWrite(ledPin, LOW);
}
delay(100); // Delay a little bit to improve simulation
performance

```

A. 3 Текст програми для роботи з групою датчиків

```

#include <DHT.h>
#include <DHT_U.h>
#include <Adafruit_Sensor.h>
#define DHTPIN 3 // stores the port number of the
temperature and humidity sensor
DHT dht(DHTPIN, DHT11); // object for working with the
DHT11 sensor
const int MQ2 = A0; // stores the port number of the gas
sensor
const int ledPin = 13; // stores the port number of the
diode
int sensorValue = 0; // stores the value read from the
gas sensor
int x; // variable for flood sensor
int inputPin = 2; // stores the motion sensor port
number
int pirState = LOW; // variable for the motion sensor
int val = 0; // variable for the motion sensor

```

```

int buttonState = 0; // variable for the light sensor

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT); // Set the mode for pin 13
  dht.begin();
  pinMode(A1, INPUT); // water_level
  pinMode(inputPin, INPUT);
// declare sensor as input
  pinMode(buttonPin, INPUT);
}

void loop() {
  Serial.print(dht.readHumidity());
  Serial.print(",");
  Serial.print(dht.readTemperature());
  Serial.print(",");
  sensorValue = analogRead(MQ2);
  Serial.print(sensorValue);
  Serial.print(",");
  x = analogRead(A1); if (x > 100) {
    digitalWrite(13, HIGH); }
  if (x < 100) { digitalWrite(13, LOW); }
  Serial.print(x);
  Serial.print(",");
  val = digitalRead(inputPin);
  if (val == HIGH)
  {
    Serial.print("M_detected,");
    pirState = HIGH; }
  else {
    Serial.print("No_motion,");

```

```

    pirState = LOW;}
buttonState = digitalRead(buttonPin);
if (buttonState == HIGH)
{digitalWrite(ledPin, HIGH);
Serial.print("Light_off,");}
else
{digitalWrite(ledPin, LOW);
Serial.print("Light_on");}
Serial.println("");
}

```

A.4 Текст програми для функції Node-RED

```

// Простий контролер водонагрівача
var SetTemp = global.get("HeaterSetTemp");
var ActualTemp = msg.payload;
var Hysteresis = 2;
msg2 = {};
payload = {};
if (ActualTemp <= SetTemp - Hysteresis) { msg2.payload=1;
}
else if (ActualTemp > SetTemp + Hysteresis) {
msg2.payload=0;
}
return msg2;

```

A.5 Текст програми файлу OpenHAB .sitemap

```

sitemap demo label="Home Automation Demo" {
  Frame label="Water heater settings" {
    Setpoint item=heater_hi_temp minValue=20 maxValue=75
    step=1 icon=temperature_hot
    Setpoint item=heater_lo_temp minValue=10 maxValue=65
    step=1 icon=temperature_cold
    Setpoint item=time_t1 minValue=0.00 maxValue=23
    step=1 Setpoint item=time_t2 minValue=0.00
    maxValue=23 step=1 Setpoint item=time_t3
    minValue=0.00 maxValue=23 step=1 Setpoint
    item=time_t4 minValue=0.00 maxValue=23 step=1
  }
  Frame label="Water heater status" {
    Text item=heater_temp Text item=heater_status
  }
  Frame label="Other sensors" {
    Text item=room_temp icon=temperature_cold
    Text item=room_hum icon=humidity
    Text item=room_motion
  }
}

```

A.6 Текст програми файлу OpenHAB .items.

```

Group All
Number heater_lo_temp "Low Temperature (TL) [%.1f C]"
<temperature>

```

```

Number heater_hi_temp "High Temperature (TH) [%.1f C]"
<temperature>
Number time_t1 "Start time (t1) [%2.2f]" <time>
Number time_t2 "End time (t2) [%2.2f]" <time>
Number time_t3 "Start time (t3) [%2.2f]" <time>
Number time_t4 "End time (t4) [%2.2f]" <time>
Number heater_temp "Temperature [%.1f C]" <temperature>
{mq ="<[mymosqui o:tr_node/out/t:state:default]"}
String heater_status "Heater status [%s]"<energy>
{mq="<[mymosqui o:tr_node/out/s:state:default]"}
Number room_temp "Room temperature [%.1f °C]"
<temperature>{mq="<[mymosquio:th_node/out/t:state:default]
lt]"}
Number room_hum "Room humidity [%.1f %]" <temperature>
mq ="<[mymosqui o:th_node/out/h:state:default]"}
String room_motion "Motion detector [%s]" <motion> {mq
="<[mymosqui o:md_node/out/m:state:default]"}

```

A.7 Текст програми файлу-шаблону OpenHAB .rules

```

// imports
import <java.net.URL>
// variable declarations
Var <VAR_NAME> <VAR_VALUE>
// rules
rule "<RULE_NAME>
when
    <TRIGGER_CONDITION>[or <TRIGGER_CONDITION2> [or... ]]>
then
    <SCRIPT_BLOCK>
end

```

A.8 Текст програми файлу управління OpenHAB .rules

```
// imports
import java.util.Random
// variable declarations
var Timer
// rules
rule "Query water heater temperature"
when
    Time cron "1 * * * * ?"
then
    executeCommandLine("mosqui o_pub -t tr_node/in/t -m
data")
end

rule "Turn water heater on/out? "
when
    Item heater_temp received update
then
    if(((now.getHourOfDay()>= time_t1.state as DecimalType)
    &&
        (now.getHourOfDay()<time_t2.state as DecimalType))
        ((now.getHourOfDay()>= time_t3.state as DecimalType)
        &&
            (now.getHourOfDay()<time_t4.state as DecimalType))) {
    if      ((heater_temp.state as DecimalType)<      =
        ((heater_hi_temp.state as DecimalType) - 2)){
        executeCommandLine("mosqui o_pub -t tr_node/in/r -m
1")
    }
    if      ((heater_temp.state as DecimalType)>
        (heater_hi_temp.state as DecimalType) + 2)){
        executeCommandLine("mosqui o_pub -t tr_node/in/r -m
0")
    }
}
else{
    if      ((heater_temp.state as DecimalType)<
        (water_lo_temp.state as DecimalType) - 2)){
        executeCommandLine("mosqui o_pub -t tr_node/in/r -m
1")
    }
    if      ((heater_temp.state as DecimalType)>
        ((heater_lo_temp.state as DecimalType) + 2)){
        executeCommandLine("mosqui o_pub -t tr_node/in/r -m
0")
    }
}
}
end
```

ДОДАТОК Б
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дипломна кваліфікаційна робота бакалавра

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДОМАШНЬОЇ
АВТОМАТИЗАЦІЇ ДЛЯ БЕЗДРОТОВОЇ МЕРЕЖІ
СЕНСОРІВ ТА АКТУАТОРІВ**

Виконав
ст. гр. КНТ-129

Антон АБАЛЬМАС

Керівник
к.т.н., доцент

Анжеліка ПАРХОМЕНКО

Рисунок Б.1 – Слайд 1



Рисунок Б.2 – Слайд 2

МЕТА ТА ЗАВДАННЯ РОБОТИ

Мета роботи – дослідження та реалізація масштабованого, доступного програмно-апаратного рішення для бездротової мережі сенсорів та актуаторів (БМСА), що забезпечить ефективне функціонування СДА.

Завдання роботи:

- дослідити технології реалізації БМСА;
- виконати аналіз вимог та розробку архітектури СДА на основі БМСА;
- виконати проєктування та конструювання програмного та апаратного забезпечення БМСА;
- розробити конфігурацію СДА на основі БМСА.

3

Рисунок Б.3 – Слайд 3

АНАЛІЗ ПРОТОКОЛІВ ПЕРЕДАЧІ ДАНИХ

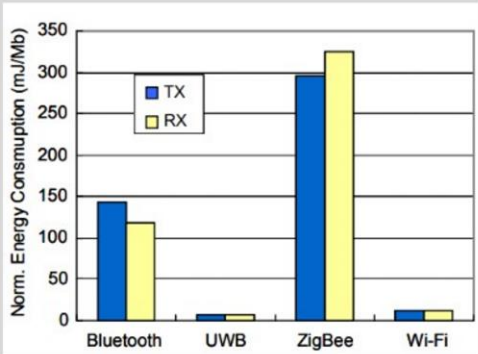
Стандарт	Bluetooth	UWB	ZigBee	Wi-Fi
IEEE специфікація	802.15.1	802.15.3a *	802.15.4	802.11a/b/g
Діапазон частот	2.4 ГГц	3.1-10.6 ГГц	868/915 МГц; 2.4 ГГц	2.4 ГГц; 5 ГГц
Максимальна швидкість сигналу	1 Мб/с	110 Мб/с	250 Кб/с	54 Мб/с
Номинальний діапазон	10 м	10 м	10 - 100 м	100 м
Номинальна потужність передачі	0 - 10 dBm	-41.3 dBm/МГц	(-25) - 0 dBm	15 - 20 dBm
Кількість радіочастотних каналів	79	(1-15)	1/10; 16	14 (2.4 ГГц)
Пропускна здатність каналу	1 МГц	500 МГц - 7.5 ГГц	0.3/0.6 МГц; 2 МГц	22 МГц
Тип модуляції	GFSK	BPSK, QPSK	BPSK (+ ASK), O-QPSK	BPSK, QPSK, COFDM, CCK, M-QAM
Поширення	FHSS	DS-UWB, MB-OFDM	DSSS	DSSS, CCK, OFDM
Максимальна кількість вузлів	8	8	> 65000	2007
Захист даних	16-біт CRC	32-біт CRC	16-біт CRC	32-біт CRC

4

Рисунок Б.4 – Слайд 4

ПОРІВНЯННЯ ЕНЕРГОСПОЖИВАННЯ ПРИ ВИКОРИСТАННІ РІЗНИХ ПРОТОКОЛІВ

Стандарт	Bluetooth	UWB	ZigBee	Wi-Fi
Чіпсет	BlueCore2	XS110	CC2430	CX53111
Робоча напруга, В	1.8	3.3	3.0	3.3
Передача даних (TX), мА	57	~227.3	24.7	219
Прийом даних (RX), мА	47	~227.3	27	215
Швидкість передачі даних, Мб/с	0.72	114	0.25	54

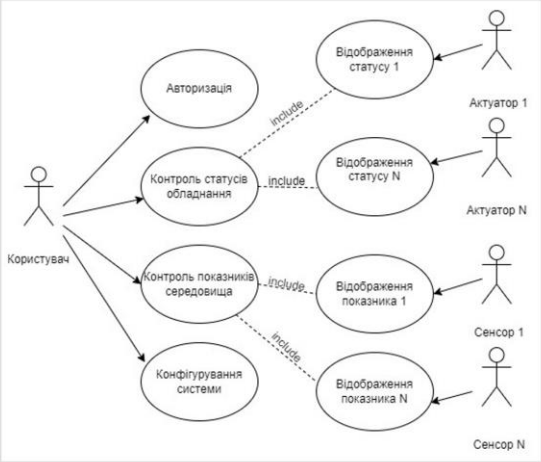


Lee, J. S. A comparative study of wireless protocols: Bluetooth, UWB, ZigBee and Wi-Fi / J. S. Lee, Y. W. Su, C. C. Shen // 33rd Annual conference of the IEEE industrial electronics society, Taipei, Taiwan, 05-08 November 2007: proceedings, 2007. - P. 46-51.

5

Рисунок Б.5 – Слайд 5

ПРОЄКТУВАННЯ ПРОГРАМНО - АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ БМСА



UML діаграма діяльності

6

Рисунок Б.6 – Слайд 6

ПРОЄКТУВАННЯ ПРОГРАМНО - АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ БМСА

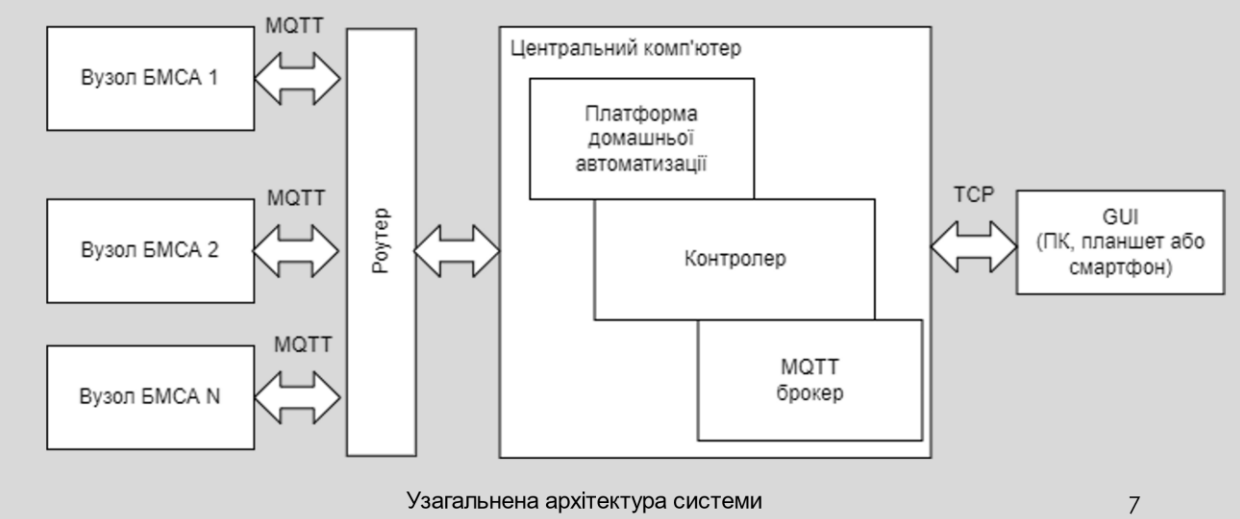


Рисунок Б.7 – Слайд 7

ПРОЄКТУВАННЯ ПРОГРАМНО - АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ БМСА

Характеристика	RaspberryPi B+	Intel Edison	Модуль	ESP32 (ESP-WROOM32)	ESP8266 (ESP8266-12E)	CC32 (CC3220MODSF)	Xbee (XB2B-WF001)
Вартість, грн	1330	2775	Центральний процесор	Tensilica Xtensa LX6 32 біт Dual-Core, 160/240 МГц	Tensilica LX106 32 біт 80 МГц-160 МГц	ARM CortexM4, 80 МГц	-
Відео	HDMI, Direct LCD, -	-	Оперативна пам'ять	520 KB	36 KB	256 KB	-
Аудіо	Стерео вихід	I2S вихід	FLASH пам'ять	2Мб (max 64Мб)	4Мб (max 16Мб)	1Мб (max 32Мб)	-
Flash пам'ять	Мікро SD рознім	SDIO інтерфейс; 4Гб вбудована Flash пам'ять	Програмування	C, C++, Lua, та ін.	C, C++, Lua, та ін.	C (SimpleLink AT та API SDK)	команди
Оперативна пам'ять	512Мб	1Гб	Відкритість	+	+	-	-
Процесор	700МГц ARM1176JZF-S	500МГц двоядерний Atom процесор 100МГц Quark MCU	Bluetooth	4.2 BR/EDR+ BLE	-	-	-
Інтерфейс вводу/виводу	27 пінів на рознімі 0.1"	70 пінів на мікратомному рознімі 0.4 мм	UART	3	2	2	1
Ethernet	10/100 вбудований	-	GPIO	32	17	21	10
Wi-Fi	-	Двоядерний 802.11 (a/b/g/n)	SPI	4	2	1	1
Bluetooth	-	Bluetooth 2.1/4.0	I2C	2	1	1	-
USB	4 порти	1 USB-OTG	ШИМ	8	-	6	-
Інші інтерфейси	SPI, UART, I2C	SPI, UART, I2C, PWM	Розмір, мм	25.5 x 18.0 x 2.8	24.0 x 16.0 x 3.0	20.5 x 17.5 x 2.5	24.0 x 22.0 x 3.0
Енергоспоживання	5В * 600mA (~3Вт)	3.3В - 4.5В @ <1Вт	Вартість, грн	360	240	720	1040
Розміри	85мм x 56мм x 19.5мм	60мм x 29мм x 8мм					

Порівняння та вибір центрального комп'ютера та комунікаційного модуля

Рисунок Б.8 – Слайд 8

ПРОЄКТУВАННЯ ПРОГРАМНО - АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ БМСА

Характеристика	MQTT	CoAP
Стандарт	ISO/IEC PRF 20922	RFC 7252
Парадигма	Видавець-підписник	Запит-відповідь
Кількість повідомлень	16	4
Розмір заголовку	2 байта	4 байта
Протокол транспортного рівня	TCP	UDP
RESTful	-	+

Порівняльна характеристика протоколів обміну повідомленнями

Вузол БМСА	Призначення	Компоненти
1	Керування нагрівальними елементами	Olimex ESP8266-DEV, датчик температури DS18B20 у водонепроникному корпусі 1-канальне реле 16A, перетворювач Vigortronix 220 V до 5V AC/DC
2	Контроль мікроклімату	Olimex ESP8266-DEV, датчик температури та вологості DHT11, перетворювач Vigortronix 220 V до 5V AC/DC
3	Виявлення руху	Olimex ESP8266-DEV, пасивний інфрачервоний (PIR) сенсор, перетворювач Vigortronix 220 V до 5V AC/DC

Опис спроєктованих вузлів БМСА

9

Рисунок Б.9 – Слайд 9

ПРОЄКТУВАННЯ ПРОГРАМНО - АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ БМСА

Характеристика	Arduino IDE	Microsoft Visual Studio	PlatformIO
Простота	+	-	+
Підтримка інших плат	+	-	+
Платформа	Java	C/C++, JavaScript C#	Python
Операційна система	OC Windows, MacOS X i Linux	Windows, macOS i Linux	Windows, macOS, Linux, FreeBSD
Менеджер бібліотек	-	+	+
Підтримка плагінів та ін. інструментів	Processing	HTML, CSS	node.js, typescript, JavaScript
Наявність вебверсії	+	-	-
Додаткові функції	-	+	+

Платформа	Мова	Веб-інтерфейс	Протоколи	Сервіс	API	Плагіни	Документація
OpenHAB	Java	+	+	MQTT	+	+ REST	+
Home-Assistant	Python3	+	+	MQTT	REST/Python/Websocket APIs	+	+
Domoticz	C++	+	+	MQTT	+	+ JSON-based	+
Jeedom	PHP	+	+	MQTT	+ JSON RPC, HTTP-based	+	+ in French
Calaos	C++	+	+	-	+ JSON-based	-	+ in French

Порівняння та вибір середовища програмування та платформи СДА

10

Рисунок Б.10 – Слайд 10



Рисунок Б.11 – Слайд 11

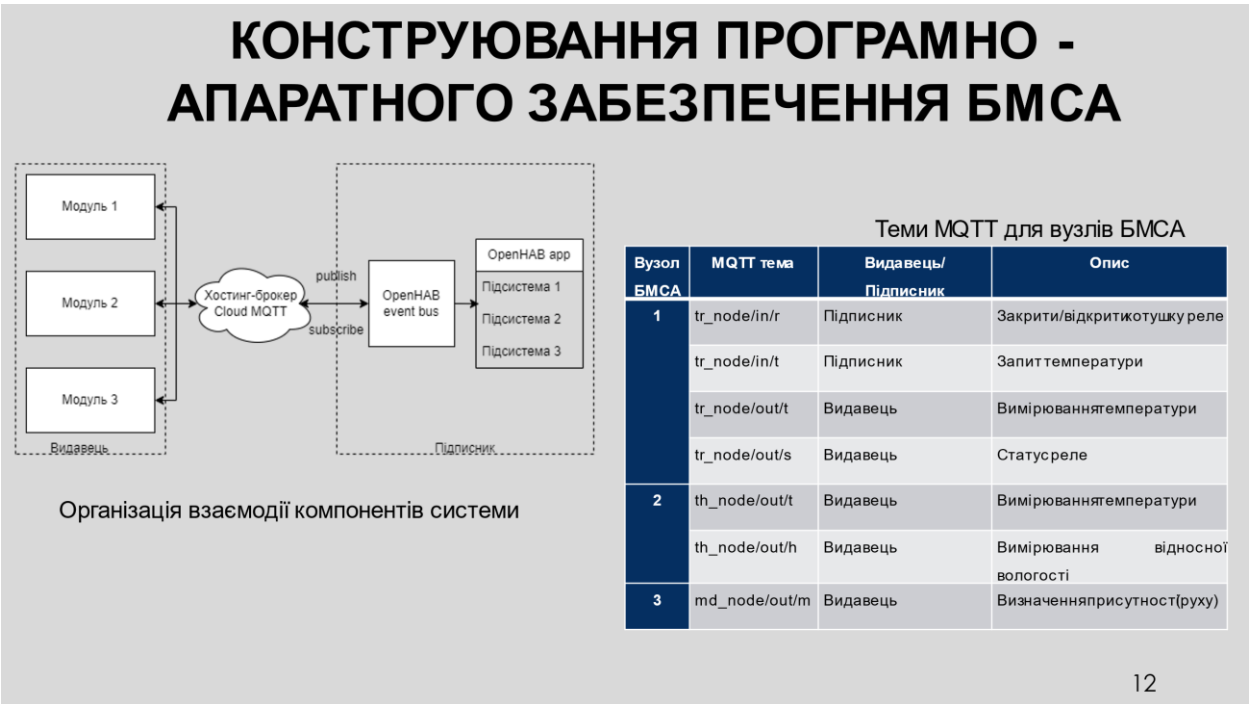
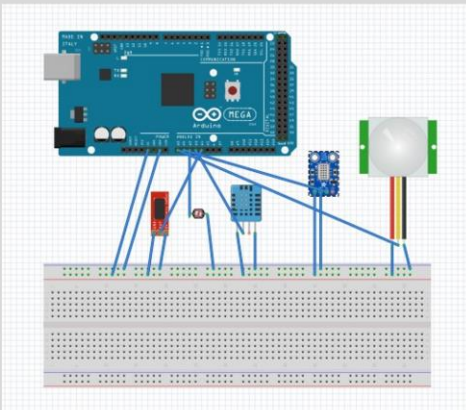


Рисунок Б.12 – Слайд 12

КОНСТРУЮВАННЯ ПРОГРАМНО - АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ БМСА



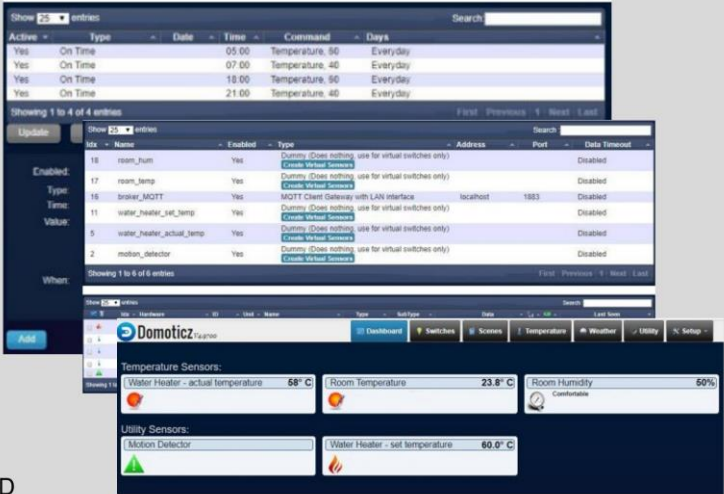
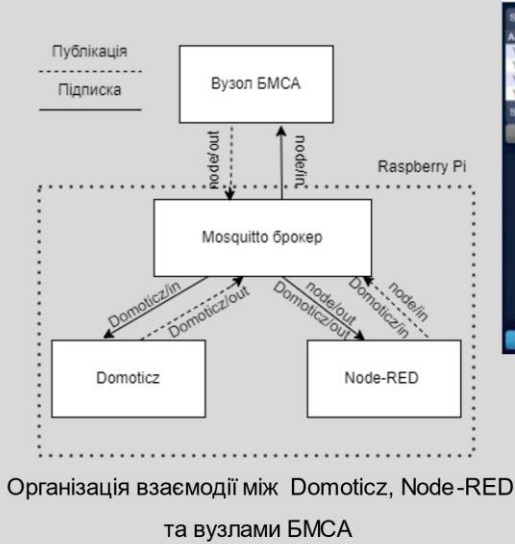
Назва функції	Опис функції
Serial.begin(9600)	Відкриває послідовний порт, встановлює швидкість 9600 біт/с
pinMode(ledPin, OUTPUT)	Встановлює режим роботи вихід для 13 піна
dht.begin()	Запускає датчик вологості та температури за допомогою бібліотеки DHT.h
pinMode(A1, INPUT)	Встановлює режим роботи вхід
Serial.print()	Передає дані через послідовний порт
dht.readHumidity()	Зчитує вологість
dht.readTemperature()	Зчитує температуру
analogRead(MQ2)	Зчитує значення з датчика газу
digitalWrite(13, HIGH)	Включає світлодіод
digitalWrite(13, LOW)	Вимикає світлодіод

Прототип вузла БМСА з підключеними датчиками та опис функцій програми

13

Рисунок Б.13 – Слайд 13

КОНСТРУЮВАННЯ ПРОГРАМНО - АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ БМСА



Інтерфейс Domoticz

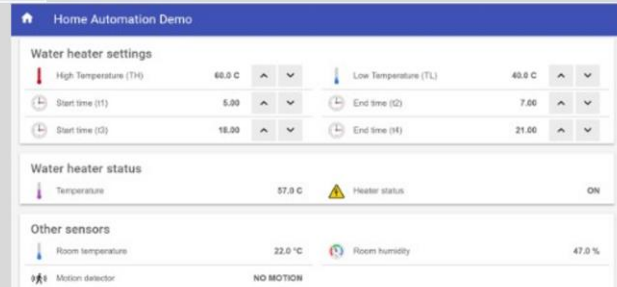
14

Рисунок Б.14 – Слайд 14

КОНСТРУЮВАННЯ ПРОГРАМНО - АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ БМСА

Файл	Призначення
mqtt.cfg	Налаштування параметрів прив'язки MQTT
.sitemap	Містить інформацію про компоненти GUI
.items	Опис типів даних у середовищі OpenHAB
.rules	Правила автоматизації процесів

Розроблені файли та інтерфейс OpenHAB



15

Рисунок Б.15 – Слайд 15

ВИСНОВКИ

- В результаті виконання дипломної роботи розроблено програмно-апаратне забезпечення БМСА, що складається з центрального вбудованого комп'ютера (Raspberry Pi 3) та трьох вузлів БМСА. Воно дозволяє користувачу контролювати показники мікроклімату, статуси обладнання, дистанційно керувати електроприладами за допомогою декількох типів пристроїв (смартфон, ноутбук).
- Представлене рішення забезпечує контроль процесів в СДА, використовуючи тільки локальну мережу. Доступ до Інтернету та хмарних сервісів можливо реалізувати додатково.
- Енергоефективний контроль побутової техніки може бути досягнутий за допомогою представлених апаратних і програмних рішень разом з відповідними алгоритмами планування, які вмикають/вимикають електричні прилади, особливо нагрівальні елементи (наприклад, водонагрівачі, електричні радіатори тощо).

16

Рисунок Б.16 – Слайд 16