

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ МОНІТОРИНГУ
ВОДОПОСТАЧАННЯ В СИСТЕМІ «РОЗУМНИЙ БУДИНОК»
SOFTWARE FOR WATER SUPPLY MONITORING IN SMART
HOUSE SYSTEM

Виконала: студентка 4 курсу, групи КНТ-129

Спеціальності 121 Інженерія програмного

забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

НЕЧАЙ К.В.

(прізвище та ініціали)

Керівник ПАРХОМЕНКО А.В.

(прізвище та ініціали)

Рецензент ЗЕЛЕНЬОВА І.Я.

(прізвище 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)

НЕЧАЙ Ксенії Владиславівни

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для моніторингу водопостачання в системі «Розумний будинок». Software for Water Supply Monitoring in Smart House System

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

(прізвище, 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. Огляд технологій розробки підсистеми моніторингу водопостачання для СРБ. 3. Проектування підсистеми моніторингу водопостачання. 4. Конструювання підсистеми моніторингу водопостачання.

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	Огляд технологій розробки підсистеми моніторингу	3 тиждень	Розділ 2
4	Розробка архітектури системи	4 тиждень	Розділ 3
5	Розробка програмного-апаратного забезпечення підсистеми моніторингу	5 тиждень	Розділ 3
6	Тестування та експериментальне дослідження програми	6 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	7 тиждень	ПЗ, Додатки
8	Захист роботи	8 тиждень	

Студентка

_____ Ксенія НЕЧАЙ
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи)

_____ Анжеліка ПАРХОМЕНКО
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 101 с., 11 табл., 34 рис., 2 дод., 66 джерел.

СИСТЕМА РОЗУМНИЙ БУДИНОК, МОНІТОРИНГ ВОДОПОСТАЧАННЯ, ДАТЧИК РІВНЯ ВОДИ, ПРОГРАМНО-АПАРАТНА ПЛАТФОРМА, ІНТЕРФЕЙС, ЧАТ-БОТ.

Об'єкт дослідження – процес моніторингу водопостачання.

Предмет дослідження – методи та програмно-апаратні засоби моніторингу водопостачання.

Мета роботи – дослідження та розробка програмно-апаратного забезпечення підсистеми моніторингу водопостачання для запобігання витоку води та надлишковому її споживанню в системі Розумний будинок (СРБ).

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування C++, платформа домашньої автоматизації OpenHAB, Arduino Uno, Raspberry Pi під управлінням операційної системи Raspbian, датчик води, реле, світлодіоди, динамік.

Результати. Створено програмне забезпечення, що складається з двох частин – комплексу програм для Arduino Uno та конфігураційних файлів для OpenHAB.

Висновки. Розроблено підсистему моніторингу водопостачання для СРБ, що дозволяє контролювати рівень води, візуалізувати показники та стани в інтерфейсній формі, формувати керуючий вплив на обладнання (включення/виключення водяного насосу) та надсилати сповіщення про надзвичайні ситуації в Telegram.

Галузь використання – системи домашньої, офісної та промислової автоматизації.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 101 pages, 11 tables, 34 figures, 2 appendixes, 66 sources.

SMART HOUSE SYSTEM, WATER SUPPLY MONITORING, WATER LEVEL SENSOR, SOFTWARE/HARDWARE PLATFORM, INTERFACE, CHAT-BOT.

Object of study is the process of water supply monitoring.

Subject of study are methods and hardware/software tools for water supply monitoring.

The goal of the work is development of hardware and software for water supply monitoring subsystem to prevent water leakage and excessive water consumption in Smart House system (SHS).

Materials, methods and technical tools: object-oriented programming, C++ programming language, OpenHAB home automation platform, Arduino Uno, Raspberry Pi under Raspbian operating system, water sensor, relay, LEDs, speaker.

The results. The software was developed that consists of two parts - a set of programs for Arduino Uno and configuration files for OpenHAB.

Conclusions. A water supply monitoring subsystem for SHS has been developed, which allows monitoring the water level, visualizing indicators and states in an interface form, forming a control actions on the equipment (turning on/off the water pump) and sending notifications about emergency situations in Telegram.

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 Дослідження технологій структурної організації системи Розумний будинок.....	10
1.2 Аналіз варіантів реалізації СРБ.....	13
1.2.1 Порівняння реалізацій підсистеми зв'язку СРБ.....	13
1.2.2 Порівняння реалізацій підсистеми управління СРБ.....	19
1.3 Постановка завдань дипломної роботи.....	22
2 Огляд технологій розробки підсистеми моніторингу водопостачання для СРБ.....	23
2.1 Дослідження існуючих систем моніторингу водопостачання.....	23
2.2 Аналіз програмно-апаратних засобів моніторингу водопостачання.....	30
2.3 Висновки до розділу 2.....	34
3 Проєктування підсистеми моніторингу водопостачання.....	35
3.1 Аналіз вимог та розробка архітектури системи.....	35
3.2 Аналіз та вибір апаратних засобів розробки.....	36
3.2.1 Вибір платформ розробки.....	36
3.2.2 Вибір плати Arduino.....	38
3.2.3 Аналіз та вибір датчика рівня води.....	43
3.3 Аналіз та вибір програмних засобів розробки.....	47
3.3.1 Вибір середовища програмування.....	47
3.3.2 Вибір платформи домашньої автоматизації.....	49
3.4 Висновки до розділу 3.....	51
4 Конструювання підсистеми моніторингу водопостачання.....	52
4.1 Розробка підсистеми моніторингу водопостачання.....	52
4.1.1 Моделювання роботи прототипу в середовищі Tinkercad.....	52

4.1.2 Розробка базового модуля.....	53
4.1.3 Розробка модуля автоматичного вмикання/вимикання.....	54
4.1.4 Розробка модуля звукового сповіщення.....	55
4.1.5 Розробка модуля світлового сповіщення.....	57
4.1.6 Алгоритм функціонування підсистеми моніторингу.....	59
4.2 Розробка підсистеми нотифікації на платформі OpenHAB.....	60
4.2.1 Налаштування OpenHAB для роботи на Raspberry Pi.....	62
4.2.2 Налаштування OpenHAB для віддаленого керування і отримання повідомлень.....	66
4.3 Висновки до розділу 4.....	70
Висновки.....	71
Перелік джерел посилань.....	72
Додаток А Текст програми.....	79
Додаток Б Слайди презентації.....	94

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

AODVjr	– Ad hoc On-Demand Distance Vector Routing Junior;
DMPR	– Disjoint Multipath Routing;
GPRS	– General Packet Radio Service;
HSDPA	– High-Speed Downlink Packet Access;
IDE	– Integrated Development Environment;
IoT	– Internet of Things;
LQIR	– Link Quality Indicator Based Routing;
SHS	– Smart House System;
АЦП	– аналого-цифровий перетворювач;
БСМ	– бездротова сенсорна мережа;
МК	– мікроконтролер;
ПЗ	– програмне забезпечення;
ПК	– персональний комп'ютер;
СРБ	– система Розумний будинок;
ШІМ	– широтно-імпульсна модуляція.

ВСТУП

Останнім часом системам Розумний будинок (СРБ) приділяється значна увага завдяки передовим технологіям, які роблять їх ще ефективнішими, а щоденне життя людей ще комфортнішим. Зокрема, розвитку технологій домашньої автоматизації сприяє використання Інтернету речей (IoT) [1].

IoT – це мережа підключених пристроїв, які можна віддалено контролювати через Інтернет. IoT технологія отримала значний розвиток в останні роки і зараз використовується в різних секторах, включаючи домашню автоматизацію, хмарні обчислення, охорону здоров'я, телемедицину, промисловість та ін. [2] - [3]. Бездротові мережі сенсорів та актуаторів, що є частиною IoT, дозволяють підключати інтелектуальні пристрої, які можуть спільно використовувати ресурси або спілкуватися один з одним, що є найважливішою технологією для створення СРБ [4] - [5]. Таким чином, СРБ - це концепція, яка є частиною парадигми IoT, що дозволяє споживачам віддалено контролювати побутову техніку, підключаючи її до Інтернету [3].

Основна перевага домашньої автоматизації полягає у легкому керуванні різноманітним обладнанням та процесами (освітлення, опалення, вентиляція та кондиціонування, використання побутових приладів та т.і.), а також контролі цих процесів з метою економії ресурсів, створення безпечних та комфортних умов проживання мешканців [6].

Але задача подальшого розвитку СРБ залишається актуальною, оскільки високоефективна та результативна СРБ має постійно вдосконалюватись різними технологіями та додатковими компонентами, які дозволять задовільнити всі вимоги мешканців.

Метою дипломної роботи є дослідження та розробка програмно-апаратного забезпечення підсистеми моніторингу водопостачання для запобігання витоку води та надлишковому її споживанню в СРБ.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження технологій структурної організації системи Розумний будинок

Система Розумний будинок (СРБ) – це середовище контролю, зворотного зв'язку, дії побутової техніки та інтелектуального моніторингу на основі запитів мешканців будинку [7].

Базова структура СРБ зображена на рисунку 1.1. Вона складається з трьох основних частин:

- фізичні компоненти або електронні пристрої (датчики або сенсори, виконавчі механізми або актуатори);
- підсистема управління (автоматизована або на основі інтелектуальних технологій);
- підсистема зв'язку (дротова та/або бездротова мережа), яка з'єднує фізичні компоненти з підсистемою управління [1].



Рисунок 1.1 – Узагальнена структура системи Розумний будинок [1]

Можна отримати доступ із зовнішнього середовища до підсистеми управління СРБ через мобільну мережу або мережу Інтернет. У СРБ фізичні компоненти сприймають навколишнє середовище та взаємодіють з підсистемою управління через домашню мережу та підмережі. Підсистема управління СРБ приймає рішення та передає керуючу інформацію до

актуаторів через домашню мережу. Наприклад, датчик газу виявляє витік газу в СРБ і передає це повідомлення підсистемі управління через бездротову мережу. Підсистема управління вирішує вимкнути газовий клапан і передає цю керуючу інформацію приводу, який вимикає/закриває газовий клапан [1].

Роль фізичних компонентів дуже важлива, оскільки вони вимірюють та збирають інформацію та діляться нею з підсистемою управління через домашню мережу. В якості фізичних компонентів використовуються сенсори, мікроконтролери, приводи та інтелектуальні пристрої.

У таблиці 1.1 наведено властивості, режими і сфери застосування датчиків [8] - [9]. Датчики спостерігають за взаємодією мешканців СРБ з такими об'єктами, як двері, вікна, ключі та вся побутова техніка, тобто фактично розпізнають активності повсякденного життя.

Таблиця 1.1 – Властивості, режими і сфери застосування датчиків [8] - [9]

Властивості	Режими	Застосування
Фізичні властивості	Тиск, температура, освітленість, вологість, потік	Охорона здоров'я, енергоефективність
Рух та присутність	Позиція, кут, швидкість, прискорення, напрямок, відстань	Безпека, відстеження місцезнаходження, виявлення падінь
Біохімічні агенти	Тверді речовини, рідини, гази	Моніторинг безпеки та здоров'я, технічне обслуговування басейну
Інші	Інформаційні кнопки, звук, зображення	Використовується для ідентифікації людей і предметів, регулювання гучності, розпізнавання мовлення, розуміння контексту

Підсистема управління отримує інформацію від різних датчиків і класифікує її для різних видів діяльності. Наприклад, дані, зібрані з датчика акселерометра розташовані на тілі, розпізнають дії, які включають повторювані рухи тіла, як-от ходьба, біг тощо.

Ряд моделей машинного навчання використовується для розпізнавання активностей в СРБ [9], зокрема:

- наївні байєсівські класифікатори – ідентифікують активність, яка з найбільшою ймовірністю відповідає набору спостережуваних значень датчика;
- дерева рішень - дозволяють вивчити логічний опис діяльності.

Ймовірнісна послідовність подій датчика кодується за допомогою марковських моделей, байєсівських мереж і умовних випадкових полів. Використання нейронних мереж або нечіткої логіки дозволяє розробляти автоматизовані методи прийняття рішень і контролю [1].

Підсистема зв'язку використовується для обміну інформацією між фізичними компонентами та підсистемою управління СРБ на основі дротового або бездротового зв'язку. Популярними бездротовими технологіями є Bluetooth, Wi-Fi, ZigBee [10].

Bluetooth - це перший популярний бездротовий інтерфейс з низькою пропускною здатністю для СРБ. Підсистема зв'язку на його основі є недорогою, простою в реалізації, але працює лише на коротких відстанях.

Бездротова технологія ZigBee стала цілком придатною для використання в СРБ завдяки низькій вартості, низькому енергоспоживанню та легкій інтеграції [1]. Але, Zigbee має невисоку швидкість передачі даних та низьку стабільність мережі, а також високу вартість обслуговування [1].

За останні кілька років вимоги до пропускної здатності СРБ різко зросли, завдяки чому з'явилася технологія бездротової локальної мережі Wi-Fi на основі IEEE 802.11. Вона охоплює весь будинок, але споживає більше енергії та забезпечує низький рівень безпеки. Оскільки Wi-Fi зараз встановлено в більшості будинків, набагато простіше придбати пристрої, які мають вбудовані можливості Wi-Fi. Крім того, Wi-Fi є дешевим варіантом, він простий у використанні та підключенні [1].

Порівняння вищезазначених чотирьох технологій побудови бездротової сенсорної мережі (БСМ) для СРБ наведено в таблиці 1.2 [1].

Таблиця 1.2 – Порівняльний аналіз бездротових технологій [1]

	Смуга частот	Швидкість біт/сек	Енерго-споживання	Безпека	Відстань, м
Bluetooth	2.4G	1 M	>10mW	+	10
Wi-Fi	2.4G/5G	11-54 M	>10mW	-	200
ZigBee	868/915M, 2.4G	20-250 K	<10mW	+	100

1.2 Аналіз варіантів реалізації СРБ

Порівняльний аналіз варіантів СРБ може виконуватись з точки зору особливостей реалізації підсистеми управління та підсистеми зв'язку.

При реалізації підсистеми зв'язку використовуються різноманітні технології, протоколи та топології організації мережі СРБ. При реалізації підсистеми управління використовуються різноманітні методи та процедури моніторингу станів та контролю процесів функціонування обладнання та пристроїв в будинку або квартирі [1].

1.2.1 Порівняння реалізацій підсистеми зв'язку СРБ

В таблиці 1.3 наведено аналіз особливостей реалізації підсистеми зв'язку з точки зору застосовуваних алгоритмів маршрутизації та мережевих технологій [11] - [24].

В роботі [11] запропоновано Розумну систему управління енергоспоживанням для СРБ. Ця система розподіляє та призначає різні завдання домашньої мережі відповідним компонентам. Вона може інтегрувати інформацію з сенсорів та контролювати різні домашні пристрої споживачів, такі як світильники, газові клапани, штори, телевізор та кондиціонери за підтримки бездротової мережі сенсорів та актуаторів.

Таблиця 1.3 – Порівняльний аналіз варіантів реалізації підсистем зв’язку СРБ [11] - [24]

Аспект реалізації	Задача	Пристрої	Домашня мережа	Підсистема управління	Зовнішня мережа	Мережевий алгоритм
1	2	3	4	5	6	7
Алгоритми маршрутизації	Розумна система управління енергоспоживанням [11]	Сенсори, актуатори	ZigBee	Екстрактор інформації	Інтернет	Jess / DMPR
	Розробка вузла та координатора для СРБ [12]	Вузол, координатор	ZigBee	Архітектура на основі механізму зв’язку сокетів протоколу TCP/IP	GPRS	Dijkstra
	Інтелектуальна СРБ на основі БСМ [13]	Сенсори, актуатори	ZigBee	-	Інтернет	LQIR
	Удосконалений алгоритм маршрутизації, що поєднує дерево кластерів і AODV Junior алгоритм [14]	-	ZigBee	-	-	Алгоритм виявлення та підтримки маршрутизації
Мережеві технології	Спостереження за людьми похилого віку вдома [15]	Інфрачервоні датчики, температури та вологості		Підтримуюча векторна машина	-	Алгоритм класифікації
	Інтелектуальна багатосенсорна мережа, орієнтованої на управління за допомогою мобільного пристрою [16]	Магнітний датчик, фотодіод, мікрофон, датчики руху та вібрації	Wi-Fi, Zigbee	Ультрамобільний ПК	Wi-Fi	Моніторинг

Продовження таблиці 1.3

1	2	3	4	5	6	7
Мережеві технології	Мобільна система моніторингу стану здоров'я [17]	Носимий датчик пульсу у вигляді кільця, смартфон	Bluetooth	-	HSDPA, Wi-Fi, Wimax, GPRS	-
	Проектування та впровадження архітектури домашньої автоматизації [18]	Датчики освітленості та диму	Zigbee	-	Wi-Fi	-
	Динамічна інтелектуальна система управління будинком [19]	-	Android	Аналіз поведінки користувача	GSM, WiFi	Класифікатор з використанням БД
	Внутрішній моніторинг для літніх людей [20]	Сенсорний блок для електроприладів, ліжка, води	Zigbee	Моніторинг	-	-
	Реалізація електронної системи охорони здоров'я [21]	Сенсор пульсу, тиску, пожежі	BCM	-	Інтернет	Змішаний алгоритм позиціонування
	СРБ з цифровим дверним замком [22]	Сенсори температури, газу, пожежі	Zigbee	-	Інтернет	-
	Домашній простір на основі BCM [23]	Сенсори температури, тиску, освітленості	Zigbee	Проміжне ПЗ, орієнтоване на діяльність	-	-
	Підсистема безпеки для СРБ [24]	Сенсори диму, газу, температури, біосенсор	Zigbee	Відправка СМС	GSM	-

Мережа складається з трьох компонентів:

- збору даних – збирає дані з сенсорів та надає їх компонентам прийняття рішень;
- інтелектуальної обробки – аналізує поведінку;
- управлінням послугами – адаптивно вибирає правильні домашні послуги на основі поточного стану.

Також автори запропонували новий протокол маршрутизації Disjoint Multipath Routing (DMPR) для підвищення продуктивності сенсорних мереж Zigbee [11].

Автори [12] запропонували архітектуру гібридної мережі на основі топології «зірка», використовуючи мобільні пристрої для забезпечення ефективних послуг БСМ. Ця архітектура складається з чотирьох компонентів:

- координатор мережі Zigbee, що є відповідальним за зв'язок;
- вузол Zigbee – складається з датчиків та бездротового модуля Zigbee,
- GPRS мережа – передає зібрані дані в центр моніторингу через мережу GPRS та Інтернет;
- центр моніторингу – керує даними, що генеруються всією мережею Zigbee.

В роботі також представлено удосконалений алгоритм Dijkstra, продуктивність якого оцінюється за допомогою моделювання [12].

Інтелектуальну систему контролю будинку, яка розділяє та призначає різні завдання домашньої мережі відповідними компонентами представлено в дослідженні [13]. Завдяки підтримці активних сенсорних мереж, які складаються з сенсорів та актуаторів, інформація отримується для контролю різних домашніх пристроїв споживачів. Новий протокол маршрутизації Link Quality Indicator based Routing (LQIR) розробляється для підвищення продуктивності активних сенсорних мереж [13].

В роботі [14] продемонстровано проєкт розумної домашньої мережі на основі технології Zigbee. Більш зручний та надійний бездротовий зв'язок для досягається за допомогою вдосконаленого алгоритму маршрутизації, який

будується на комбінації кластерного дерева та алгоритму Ad hoc on-demand distance vector junior (AODVjr) [14].

Автори [15] розглянули СРБ з функціями спостереження за людьми похилого віку, з інфра-червоними датчиками присутності (контроль розташування), контактами дверей (для контролю за деякими об'єктами), датчиком температури та вологості в ванній кімнаті та мікрофонами (звукова класифікація та розпізнавання мови). Дані, зібрані з різних датчиків, використовуються для класифікації кожного часового кадру певного виду діяльності щоденного життя, які раніше було визначено (гігієна, використання туалету, їжа, відпочинок, сон, спілкування та одягання/роздягання). Це робиться за допомогою векторних машин підтримки [15].

В роботі [16] автори реалізували систему моніторингу на основі Zigbee технології та мережі GPRS. Представлено вдосконалений алгоритм маршрутизації, заснований на алгоритмі Dijkstra для підтримки маршрутизації з кількома стрибками [16].

Автори [17] представили мобільну систему моніторингу здоров'я з інтеграцією смартфона з датчиком типу кільця. За допомогою Bluetooth результати вимірювань передаються на смартфон, де користувач може контролювати власний пульс або температуру. Вимірювані дані передаються на віддалений сервер через мобільний зв'язок з використанням таких технологій як High-Speed Downlink Packet Access (HSDPA), Wi-Fi, WiMax, General Packet Radio Service (GPRS) тощо. Віддалений сервер також відстежує положення персони, що моніториться, в режимі реального часу [17].

В статті [18] реалізовано систему домашньої автоматизації на основі Zigbee та мережі Wi-Fi, які інтегруються через загальний домашній шлюз, що забезпечує простий та гнучкий інтерфейс користувача та віддалений доступ до СРБ [18].

Новий тип інтелектуальної системи домашньої автоматизації [19] використовує Android смартфон в якості домашнього шлюзу для аналізу

поведінки користувачів з метою економії енергії та автоматичного вимкнення невикортовуваних пристроїв.

Автори [20] створили систему моніторингу для догляду за людьми похилого віку за допомогою технології Zigbee та декількох високоточних але недорогих датчиків.

Розробники [21] рекомендували застосування БСМ для цілодобового постійного моніторингу щоденної діяльності людей похилого віку. Зокрема, пропонується змішаний алгоритм позиціонування для визначення місця, де знаходиться літня людина. Це допомагає системі визначити діяльність людини та в подальшому приймати рішення про її стан здоров'я. Ця система забезпечує два типи основних потреб:

- визначення нетипових подій та оповіщення лікаря про надзвичайні ситуації через автоматичний дзвінок, SMS та електронну пошту;
- визначення щоденних вимог, таких як прийом ліків, вимкнення водонагрівача тощо.

Автори [22] розробили розумну систему цифрового блокування дверей для домашньої автоматизації. Ця система складається з мережі датчиків та приводів із цифровим замком дверей як базової станції. Система блокування дверей складається з RFID-зчитувача для автентифікації користувачів; сенсорного екрану, модуля двигуна для відкриття та закриття дверей, датчика для виявлення стану всередині будинку, модуля зв'язку та модуля управління для управління іншими модулями. Перевага цієї системи полягає в тому, що вона може бути легко встановлена без будь-якої додаткової інфраструктури [22].

В [23] реалізовано СРБ, щоб відслідковувати об'єкти та зміни їх станів, встановлені на основі взаємодії користувача з ними. БСМ побудована на основі протоколу зв'язку з Zigbee, включає 42 повсякденних об'єкта та 81 датчик контролю зміни стану восьми типів.

Віддалену систему безпеки будинку на основі Zigbee представлено в роботі [24]. БСМ включає: центральний вузол з мікроконтролером та модулем

GSM; вузлом збору даних; вузлом управління пристроєм та мобільним телефоном. Ця система надсилає сигнал тривоги на центральний вузол БСМ, коли виникає нетипова ситуація. Центральний вузол надсилає коротке повідомлення про тривогу користувачам через модуль GSM та мережу GSM. Також користувач може керувати різними пристроями, підключеними до блоку управління пристроєм через SMS [24].

1.2.2 Порівняння реалізацій підсистеми управління СРБ

В таблиці 1.4 наведено аналіз особливостей реалізації підсистеми управління СРБ з точки зору задач, використовуваних компонентів, засобів зв'язку, підсистеми управління та інтерфейсу користувача [25] - [30].

Автори [25] запропонували прогнозування участі користувачів у реальних сценаріях управління енергоспоживанням для СРБ. Для вирішення проблеми збалансування попиту на енергію та генерації використовуються зовнішні сигнали, що відображають стан сітки низької напруги. Представлено два алгоритми прогнозування для оцінки майбутньої поведінки СРБ: модель типу дня та ймовірнісний підхід, заснований на напівмарківській моделі першого порядку [25].

Оскільки люди дотримуються деяких конкретних моделей поведінки у своєму житті, автори [26] запропонували ідею класифікації видів діяльності домашнього користувача для прогнозування подій в СРБ. У цій роботі запропоновано багатоагентну систему для відстеження дій користувача для виокремлення завдань та статусів домашніх пристроїв [26].

Розумна домашня енергетична система, що представлена в [27] складається з розумного лічильника, розумної розетки/вимикача, контролера електромережі, розумного інтерактивного терміналу та інших розумних пристроїв.

Таблиця 1.4 – Порівняльний аналіз варіантів реалізації підсистем управління СРБ [25] - [30]

Задача	Пристрої	Домашня мережа	Підсистема управління	Зовнішня мережа	Контрольний алгоритм
Прогнозування взаємодії користувача в енергоменеджменті розумного будинку [25]	-	-	Метод дерева рішень	-	Гібридний алгоритм прогнозування (модель типу дня та напівмарківська модель першого порядку)
Дослідження психологічних особливостей домашнього користувача за допомогою мультиагентної системи [26]	-	-	Мультиагентна система	-	Алгоритм кластеризації подій в СРБ
Дружнє управління енергією в СРБ [27]	Розумні лічильники, вимикачі та ін.	Zigbee	На основі сітки	Розумний інтерактивний термінал	Програмована логічна інтегральні схема
Застосування агентної системи для СРБ [28]	Термодатчики	-	Агент BDI з використанням Jadex	-	Агент Jadex – ієрархічна декомпозиція цілей – ієрархія плану цілей
Енергоефективна СРБ [29]	Розумні пристрої	Бездротові розумні лічильники	Hydra - проміжне середовище	-	Hydra – управління подіями
Професійна, адаптивна, нечітка СРБ [30]	Датчик освітленості	-	Нечіткий процес управління	-	Зміна вагових коефіцієнтів, а також додавання та видалення правил

В роботі розроблено та проаналізовано основні функції, наведено структурні схеми, представлено комунікаційну технологію ZigBee та її інтелектуальний енергетичний профіль, а також наведено кілька типових транзакцій на основі профілю ZigBee Smart Energy [27].

Автори [28] розробили підсистему управління для СРБ із такими характеристиками агента, як цілеспрямованість, надійність і здатність до вивчення поведінки. Вони описують повний процес від вибору найбільш підходящого агентського підходу для СРБ до деталізації переходу від специфікації системи до впровадження.

СРБ з інтеграцією функцій енергоефективності [29] створена на основі проміжного середовища, яке полегшує інтелектуальний зв'язок різномірних вбудованих пристроїв через накладену мережу P2P. Звичайні пристрої, доступні в приватних домогосподарствах, і інтегровані бездротові засоби вимірювання електроенергії для отримання доступу до даних про споживання енергії взаємопов'язані. Ці дані використовуються для моніторингу та аналізу спожитої енергії на рівні пристрою в реальному часі. Поєднання технічно складної СРБ і водночас прозорих, інтуїтивно зрозумілих інтерфейсів користувача, що показують інформацію про енергоспоживання (вартість електроенергії, джерело енергії, споживання в режимі очікування) має потенціал для досягнення мети щодо створення енергоефективної СРБ [29].

Автори [30] розробляють ідею змінити домашнє середовище, щоб воно завжди відповідало бажанням та потребам користувача. Приводи середовища керуються проактивно, щоб система завжди могла передбачати вимоги користувача. Таким чином, користувачеві не доведеться турбуватися про керування обладнанням [30].

1.3 Постановка завдань дипломної роботи

Як показали проведені дослідження, в сучасних СРБ використовуються різні типи датчиків, наприклад, температури, газу, освітленості тощо. Поширеними технологіями для організації домашньої мережі є ZigBee через низьке енергоспоживання та Wi-Fi через розповсюдженість. Інтернет, GSM, мобільна мережа використовуються як зовнішня домашня мережа доступу. Оскільки особливості СРБ залежать відконкретної реалізації, жодна система не може вважатися кращою за інші.

Таким чином СРБ дозволяє створити комфортні та безпечні умови проживання мешканців будинків, квартир, а також працівників офісів. Але, майбутні перспективи в цій галузі зосереджені на розробці СРБ, яка дозволить заощаджувати ресурси, а відповідно й кошти. Тому, програмні та апаратні засоби, що дозволяють моніторити споживання ресурсів, зокрема води, постійно розвиваються та потребують подальшого удосконалення. Таким чином, тема дипломної роботи є актуальною.

Мета роботи – дослідження та розробка програмно-апаратного забезпечення підсистеми моніторингу водопостачання для запобігання витoku води та надлишковому її споживанню в СРБ.

Завдання роботи:

- дослідження технологій розробки сучасних IoT систем для моніторингу водопостачання;
- аналіз вимог та розробка структури підсистеми моніторингу водопостачання СРБ;
- вибір та обґрунтування апаратних та програмних засобів розробки;
- конструювання програмного забезпечення підсистеми моніторингу водопостачання;
- інтеграція підсистеми моніторингу водопостачання в СРБ.

2 ОГЛЯД ТЕХНОЛОГІЙ РОЗРОБКИ ПІДСИСТЕМИ МОНІТОРИНГУ ВОДОПОСТАЧАННЯ ДЛЯ СРБ

Вода є цінним ресурсом, який потрібно розумно витратити. Однією з найбільших проблем є втрата побутової води, що відбувається у формі надмірного використання, витоків, а іноді й недбалості. Щодня середнє домогосподарство використовує багато води, що призводить до її значного витрачання. Тому, розробка підсистеми моніторингу споживання води є актуальною задачею.

2.1 Дослідження існуючих систем моніторингу водопостачання

В роботі [31] представлено систему моніторингу споживання води, яка відключає постачання води, коли крани не використовуються (тобто, коли крани не відкриті). Пристрій контролює і перевіряє різницю тиску між входом (нормальний тиск водопостачання) і виходом (вхідна труба будинку). Коли різниця тиску однакова, основний клапан відключається, якщо тиск на вході перевищує тиск на виході, клапан відкривається. Якщо тиск на вході менше, ніж на виході, користувач сповіщується про низький тиск у лінії подачі. Процес контролюється мікроконтролером (рис. 2.1), який також відповідає за включення води, коли датчики на вході та виході визначають різницю тиску. Представлений пристрій також використовує зчитувач Total Dissolved Solids (TDS), який визначає загальну кількість розчинених твердих речовин у водопровідній воді; зчитувач кислотно-лужного балансу питної води (Potentia hydrogeni (pH)) і датчик температури, які вимірюють якісні характеристики води. Таким чином, він дозволяє контролювати використання води та її якість, зокрема можливо керувати клапаном і встановлювати часові обмеження, щоб запобігти випадковій втраті води. Система також надає інформацію про низький тиск води, що подається (місця, де використовуються пневматичні насоси).

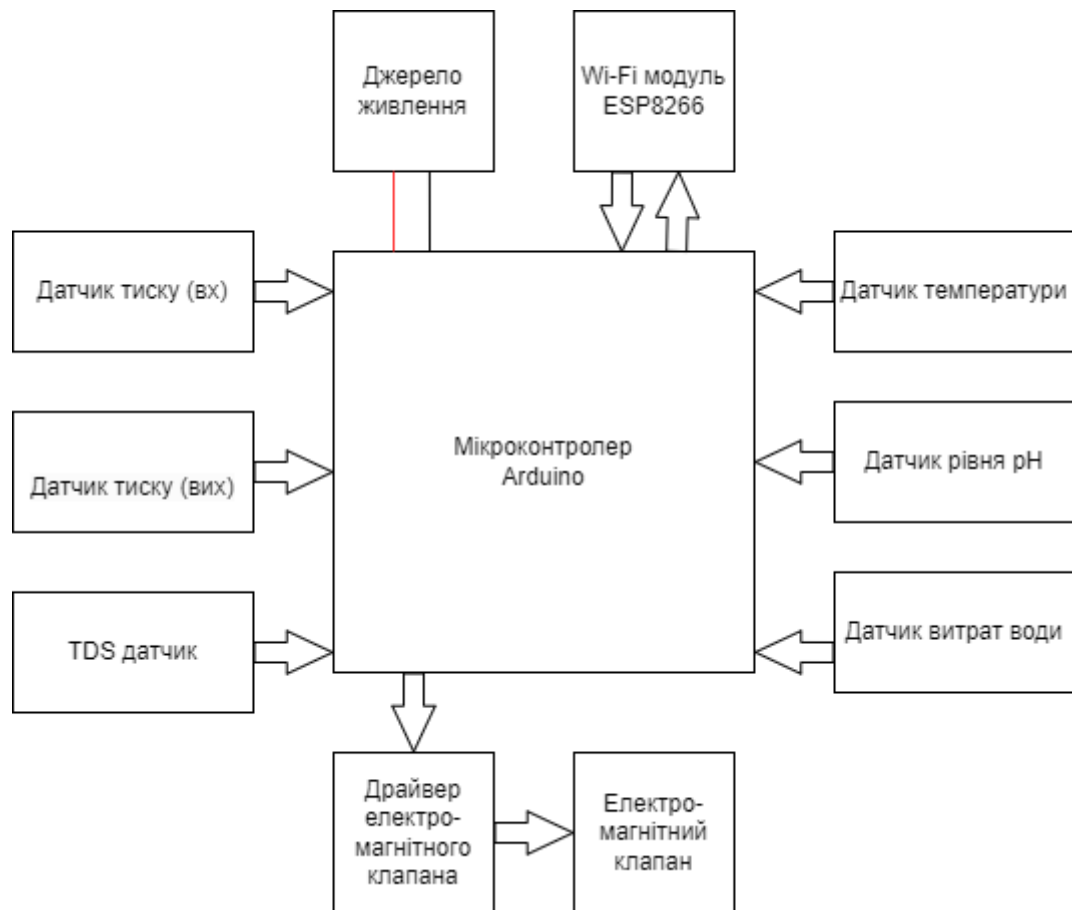


Рисунок 2.1 – Структура системи контролю водопостачання [31]

Цей пристрій підключається до домашнього Wi-Fi за допомогою модулю ESP8266 або подібного, а мікроконтролером для прототипу є Arduino або подібні плати. Пристрій надає дані про поточне споживання води, тиск, TDS води, температуру, значення рН і графік щоденного використання.

Пристрій має вбудований блок живлення, який подає 5 В на мікроконтролер і 12 В на драйвер електромагнітного клапана. Літій-полімерна батарея вмикається у разі відключення електроенергії.

Датчик TDS оновлюється 50 разів кожену секунду, а два датчики тиску оновлюються 100 разів кожену секунду. Клапан, який використовується, запускається мікроконтролером та розрахований на використання до 5 бар.

Мікроконтролер зчитує значення з датчика TDS, двох датчиків тиску, датчика температури, датчика рівня рН і датчика витрати води. Після цього він надсилає дані через модуль ESP8266 через Wi-Fi та мережу Інтернет на

телефон користувача. Таким чином, можливо контролювати пристрій, а також переглядати всі значення щодо якості та споживання води.

За допомогою цього пристрою можливо обмежити споживання води та повністю вимкнути воду, коли користувач не вдома. Система, дає користувачам уявлення про кількість літрів води, які використовуються щодня, а також її можна налаштувати залежно від способу життя користувача.

Автори роботи [32] реалізували автоматичне керування включенням/виключенням водяного насоса, крана та душу. Вони розглянули ситуацію, коли вода потрапляє спочатку в бак і потрібно контролювати її рівень та не допускати переповнення баку шляхом автоматичного перемикання системи залежно від рівня води, тому немає ймовірності переливання води.

З іншого боку, можливо, що води в баку недостатньо і користувач вимушений чекати певний період часу, коли вода підніметься до певного рівня і лише тоді буде забезпечено водопостачання.

В роботі реалізовано дві функції моніторингу та управління водопостачанням:

- автоматична подача води - водопровідні крани та душ автоматизовані за допомогою електричного електромагнітного клапана, інфрачервоного датчика та реле (рис. 2.2) [32];
- автоматичне перемикання водяного насоса - датчики провідності використовуються, щоб визначити рівень води в баку.

Один із датчиків розміщено на верхньому рівні, після якого вода переллється, а інший – на нижньому рівні, після чого потрібно наповнити резервуар. Водяний насос запускається через реле, і коли обидва датчики вказують на наявність відповідного рівня води, насос вимикається, щоб запобігти переповненню баку, а також втраті води (рис. 2.3) [32].

Система моніторингу водопостачання Dripless, що представлена в [33] складається з бездротового вимірювального вузла, вебсервера та мобільного застосунку.

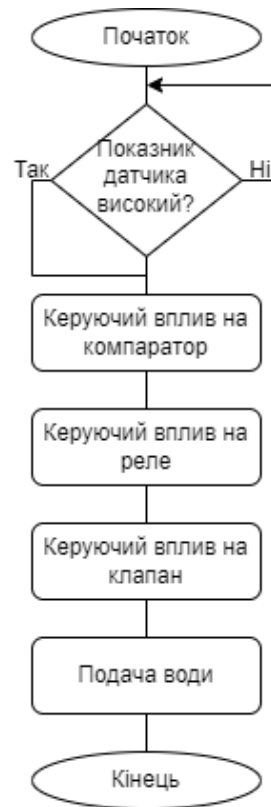


Рисунок 2.2 – Схема алгоритму контролю подачі води в кран [32]



Рисунок 2.3 – Схема алгоритму контролю подачі води в бак [32]

Вимірювальний вузол встановлюється на головному вводі води в будинок і налаштований через серверну програму для забезпечення бездротового підключення через Wi-Fi та зв'язок GSM. Як показано на рисунку 2.4, до складу системи входять пристрій для вимірювання витрати води, датчик тиску для вимірювання тиску у водопроводі, електромагнітний клапан для перекриття води та гідроелектрогенератор, який використовується для підзарядки акумуляторів. Вона також містить мікроконтролер (AVR), бездротовий модуль Wi-Fi і модуль GSM для організації зв'язку [33].

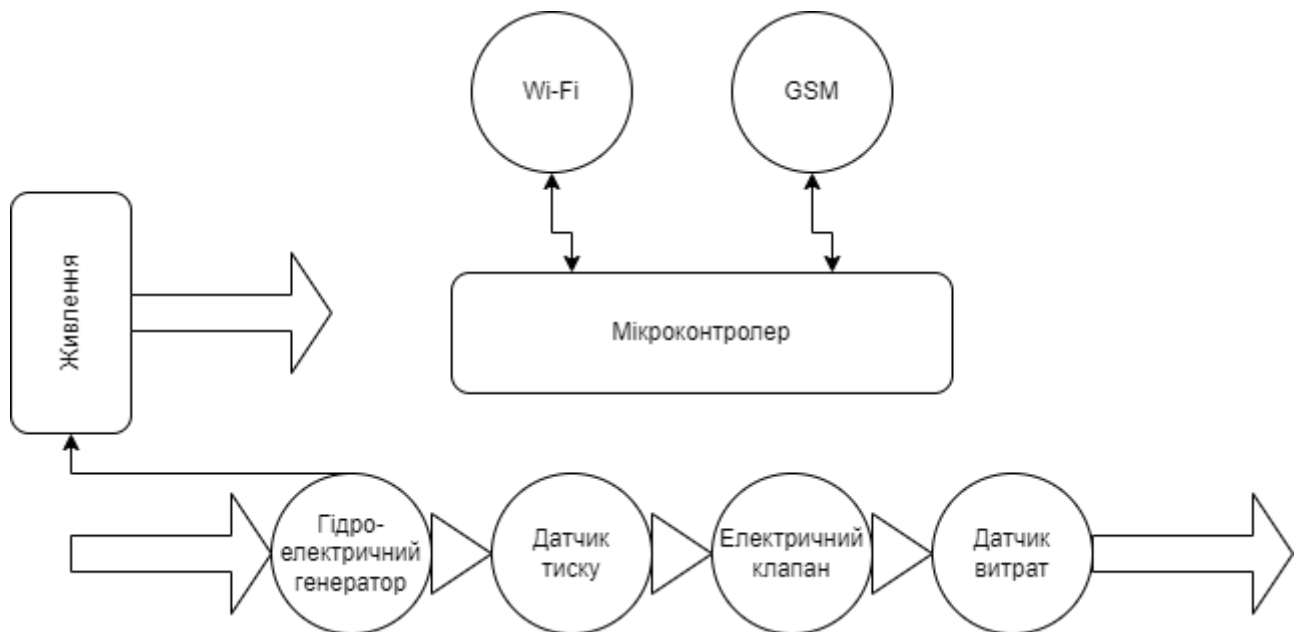


Рисунок 2.4 – Структура системи моніторингу водопостачання [33]

Статистика даних про споживання. Зазвичай споживання води відслідковується лише шляхом щомісячної фіксації та складання рахунку. й Запропонована в [33] система моніторингу дозволяє відстежувати споживання води в мобільному застосунку та на вебпанелі на основі щоденних, тижневих та місячних моделей споживання. На рисунку 2.5 показано кілька скріншотів мобільного застосунку для відстеження споживання в реальному часі та статистики використання води.

Запобігання витоку води. Головною особливістю представленої системи є її здатність виявляти ненормальні моделі використання та видавати



Рисунок 2.5 – Скріншоти мобільного застосунку для відстеження споживання води [33]

сповіщення про витік безпосередньо користувачеві через мобільні служби. Також є функція дистанційного відключення клапана з мобільного застосунку. На рисунку 2.6 показано скріншоти мобільного застосунку для таких функцій [33].

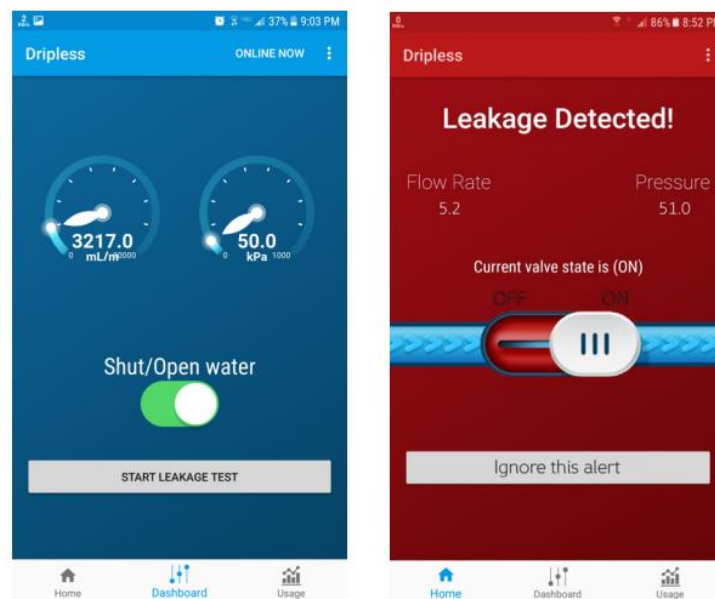


Рисунок 2.6 – Скріншоти мобільного застосунку для відстеження протікання води та нотифікації [33]

Індивідуальні поради щодо економії води. Аналізуючи моделі щоденного використання, Dripless може надсилати персоналізовані поради щодо того, як зменшити споживання води та заощадити кошти.

Водозберігаючі цілі та соціальна конкуренція. Система допомагає користувачеві встановити ціль споживання на наступний місяць і повідомляє йому, чи він досягає своєї мети, також може організовувати соціальні змагання, де користувачі отримують бали за кожен літр та змагаються з конкурентами.

Захист водяного насоса. Моніторинг роботи водяного насоса та відстеження створюваного тиску для виявлення будь-якої несправності в роботі водяного насоса або напірного бака [33].

Dripless використовує захищені з'єднання з надійними серверами, він також обмежує прямий зв'язок із бездротовою мережею лише під час першого налаштування, потім його можна повторно налаштувати через сервер. Щоб уникнути підключення адаптера живлення поруч із основним джерелом води, Dripless був розроблений таким чином, щоб забезпечити достатню потужність, оскільки він заряджає свою акумуляторну батарею за допомогою потоку води, а потім повторно використовує заряд акумуляторів для живлення, коли потоку води немає. Планування зондування та передачі було оптимізовано, щоб зменшити споживання енергії під час простою за допомогою режиму енергозбереження [33].

У статті [34] автори розробили мобільний застосунок, що дозволяє користувачеві моніторити та контролювати використання води за допомогою графічного інтерфейсу мобільного онлайн-застосунку, зокрема коли спостерігаються незвичайні показники використання води.

В роботі [35] представлено дистанційну систему вимірювання показників якості води та моніторингу потоку води в реальному часі. Автори використовують бездротову мережу датчиків на основі ZigBee для реалізації розумної системи контролю та моніторингу споживання води.

Автори статті [36] представили система рівномірного розподілу та моніторингу водопостачання, а також вимірювання якості води, що використовується для бездротових мереж.

В роботі [37] представлено систему на основі мікроконтролера та пов'язаних з ним компонентів для автоматичного контролю рівня води зі зворотним зв'язком. За допомогою цього рішення можна контролювати рівень води в баку, вмикати водяний насос, коли резервуар порожній, і вимикати той самий насос, коли резервуар заповнений, без будь-якої участі людини. Завдяки цьому виключається випадкова втрата води та раптове припинення водопостачання.

Стаття [38] присвячена розробці ефективної системи моніторингу води на основі БСМ. Концепція моніторингу води в розумних будинках/офісах передбачає три різні способи моніторингу:

- моніторинг рівня води;
- моніторинг забруднення води;
- моніторинг витоку води через несправність водопроводу.

Система [39] складається з програмованого логічного контролера, моніторів піску, які працюють у поєднанні з зондами, які зчитують дані в кількох місцях, і відвідними клапанами, які безпосередньо працюють на основі вихідних даних зондів моніторів піску.

2.2 Аналіз програмно-апаратних засобів моніторингу водопостачання

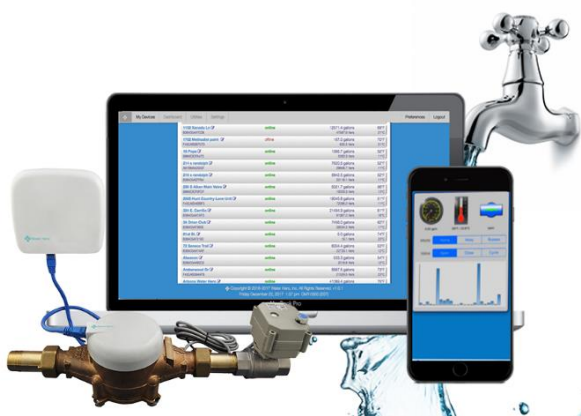
Як показали проведені дослідження, на ринку пропонуються декілька IoT систем, які надають послуги з моніторингу водопостачання. Найбільш відомі представлено на рисунку 2.7 [40] - [43].



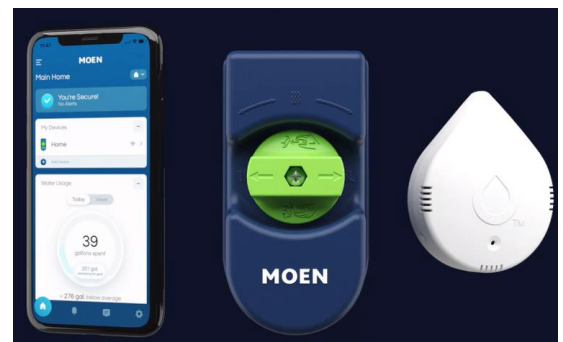
а



б



в



г

а - D-Link; б – Driblet; в – Water Hero; г - Flo

Рисунок 2.7 – IoT системи з моніторингу водопостачання [40] - [43]

Пристрій D-Link [40] – це Wi-Fi датчик води, який виявляє витік води. Він встановлюється в підвалі, де виявляє воду на кабелі датчика, після чого лунає сигнал тривоги та надсилається сповіщення на смартфон. Цей пристрій легко встановити, але він не має багатьох функцій, оскільки це просто датчик води. Він не може виміряти споживання води та допомогти зменшити його або виявити будь-яку несправність водяного насоса. Крім того, в ньому відсутній модуль GSM для спілкування з користувачем за допомогою SMS. Ціна пристрою в США становить близько 100 доларів.

Driblet [41] – це невеликий пристрій, який можна приєднати до будь-якої водопровідної труби, де він збирає дані про те, скільки води використовує домогосподарство, а також температуру та тиск води. Модуль Wi-Fi надсилає всі дані на смартфон користувача та представляє споживання води

користувачем на доступних графіках. Застосунок Dribblet також дозволяє користувачам встановлювати цілі щодо економії води та відстежувати їх прогрес. Оскільки він живиться від тиску води, Dribblet ніколи не потрібно заряджати – достатньо встановити один раз.

Система сигналізації Water Hero [42] використовує існуючу мережу Wi-Fi домовласника для збору інформації про споживання води в хмарі. Проста інформація, надана користувачем, дозволяє Water Hero почати пошук незвичайних ситуацій, які можуть свідчити про витік. Коли він виявляє ненормальне споживання води, власник будинку отримує попереджувальне повідомлення за допомогою текстового повідомлення. Якщо домовласник не вкаже інше, Water Hero активує моторизований клапан, щоб перекрити воду, захищаючи будинок.

Flo [43] – це система безпеки та управління водою на основі Wi-Fi та пристрою, що встановлений на магістральній лінії водопостачання. Система використовує програму для iOS або Android, яка контролює параметри збереження, моніторингу та безпеки. Постійно контролюючи потік, тиск і температуру води в домі, Flo виявляє аномалії, щоб допомогти уникнути таких проблем, як прорив труб. Flo має датчики тиску води, які використовують машинне навчання для проактивного виявлення витоків. Система може автоматично перекривати подачу води у відповідь на прорив труби, затоплення або інші порушення. Щоб заощаджувати воду, система визначає тиск води, швидкість потоку та температуру та вивчає моделі використання власником будинку. У застосунку Flo можливо перевірити поточне споживання води, порівняти з історичними даними та встановити цілі збереження.

В таблиці 2.1 представлено порівняння характеристик розглянутих IoT пристроїв та систем моніторингу водопостачання [33], [40] - [43].

Таблиця 2.1 – Порівняльна характеристик IoT пристроїв та систем моніторингу водопостачання [33], [40] - [43]

Система	Dripless	D-Link	Driblet	Water Hero	Flo
Простота встановлення	Помірна – встановити прямо в трубу, де вода входить до дому	Легко – датчик на підлозі	Складно - підключити до всіх водопровідних труб	Помірна – встановити прямо в трубу, де вода входить до дому	Помірна – встановити прямо в трубу, де вода входить до дому
Водоспоживання та статистика	+	-	+	+	+
Цілі споживання та поради	+	-	+	-	+
Мобільний застосунок	+	-	+	-	+
Вебінтерфейс	+	-	-	+	-
Підключення	Wi-Fi & GSM	Wi-Fi	Wi-Fi	Ethernet	Wi-Fi
Виявлення витoku	+	+	-	+	+
Метод виявлення	ІІІ	Визначення води на підлозі	-	Виявлення перевищення витрати води	Виявлення перевищення витрати води
Дія на витік	Відключення води та сирена	Сирена	-	Відключення води	Відключення води
Моніторинг насосу	+	-	-	-	-
Джерело живлення	Самокерований	Розетка АС	Самокерований	Розетка змінного струму	Розетка змінного струму

2.3 Висновки до розділу 2

Як показали проведені дослідження, на сьогоднішній день на ринку представлені апаратні та програмні рішення для моніторингу водопостачання в СРБ.

Основні задачі, що вирішують ці підсистеми наступні:

- виявлення або запобігання витoku води;
- контроль споживання води;
- автоматична подача води в кран;
- контроль наповнення баків з водою;
- контроль якості води;
- надання статистичних даних щодо споживання води за день, тиждень, місяць;
- контроль тиску в трубопроводі;
- віддалене керування водопостачанням;
- формування порад щодо споживання води;
- контроль роботи водяного насосу та іншого обладнання;
- налаштування системи контролю відопостачання відповідно до звичок мешканців.

Більшість розглянутих підсистем базуються на використанні мікроконтролера, датчиків тиску, температури, рівня та якості води. Як домашня мережа поширена БСМ на основі технології Wi-Fi. Оповіщення про позаштатні ситуації надсилаються на смартфон користувача.

3 ПРОЄКТУВАННЯ ПІДСИСТЕМИ МОНІТОРИНГУ ВОДОПОСТАЧАННЯ

3.1 Аналіз вимог та розробка архітектури системи

Розроблювана підсистема моніторингу водопостачання призначена для роботи у складі СРБ для виявлення та запобігання витoku води з нотифікацією користувачу, що дозволить заощаджувати ресурси та уникнути надзвичайних ситуацій.

Функціональні вимоги:

- контроль рівня води;
- автоматичне вмикання/вимикання подачі води;
- звукова та світлова індикація інформації про рівень води.

Нефункціональні вимоги:

- доступна елементна база;
- інтеграція в СРБ на платформі OpenHAB;
- нотифікація через Telegram про надзвичайні ситуації.

На основі аналізу вимог розроблено архітектуру системи, що представлена на рисунку 3.1.

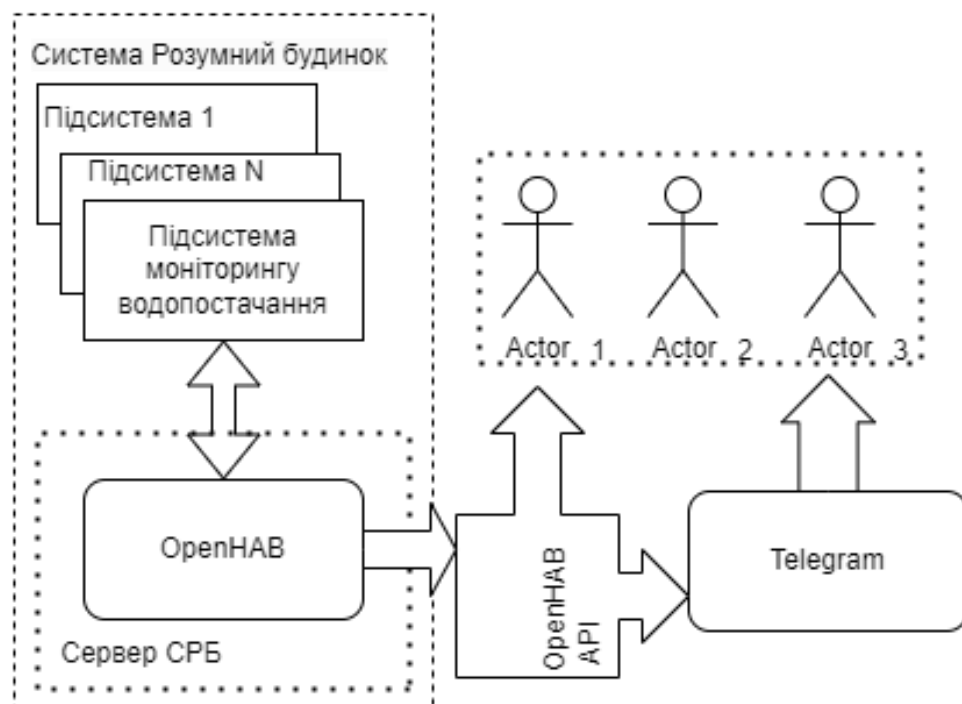


Рисунок 3.1 – Архітектура СРБ з підсистемою моніторингу водопостачання

До складу типової СРБ входять підсистеми для моніторингу різноманітних показників мікроклімату, освітленості, енергоспоживання, безпеки тощо, а також управління процесами використання побутових приладів.

Підсистема моніторингу водопостачання виконує функцію контролю рівня води та забезпечує своєчасне увімкнення/вимкнення подачі води.

Підсистема моніторингу передає дані щодо рівня води та статусу водяного насосу в OpenHAB для візуалізації та інформування користувача через інтерфейс СРБ, а також формування повідомлення користувачу в Telegram у випадку надзвичайної ситуації (рівень води вище або нижче заданого, аварійна ситуація з водяним насосом).

3.2 Аналіз та вибір апаратних засобів розробки

3.2.1 Вибір платформ розробки

Як показав проведений огляд найбільш популярних платформ розробки вбудованих систем, кожна з них має свої функціональні та структурні особливості, переваги та недоліки.

Raspberry Pi – маленька, потужна, недорога комп'ютерна плата. Може працювати в режимі сервера для обробки невеликого внутрішнього або вебтрафіку. Python - це основна мова ядра в операційній системі Raspbian. Підтримує USB 2.0 дозволяє доповнювати цей мінікомп'ютер великою кількістю периферійних пристроїв. Можливе розширення та зв'язок з мережевими пристроями через адаптер локальної мережі [44] - [45].

Arduino – це добре відома платформа електроніки з відкритим кодом, яка є центральним блоком керування багатьма вбудованими програмами керування, починаючи від інтерактивного одягу до робототехніки та 3D-принтерів. Arduino отримує команди від головного пристрою (майстра) через USB-з'єднання, виконує потрібні операції та відповідає йому. Відповідь може містити лише підтвердження або запитувані дані [46].

BeagleBone Black – це недорога платформа розробки, яка підтримується спільнотою для розробників і любителів. Дозволяє завантажити Linux менш ніж за 10 секунд і почати розробку менш ніж за 5 хвилин за допомогою лише USB-пристрою кабель [47].

Udoo – це сімейство одноплатних комп'ютерів із відкритим кодом і аксесуарів, що поєднує світ Arduino та фізичних обчислень із програмуванням Android/Linux. Плати Udoo можуть бути використані і як міні-ПК для споживчого використання, і як вбудовані системи для виробників, а також для стартапів обладнання та підприємств, які прагнуть створювати швидкі прототипи без початкових витрат на виробництво [48].

Порівняльну характеристику розглянутих платформ розробка наведено в таблиці 3.1 [44] - [48].

Таблиця 3.1 – Порівняння платформ розробки [44] - [48]

Плата/ Характеристика	Raspberry Pi	Arduino	BeagleBone Black	Udoo
CPU	ARM, BCM2835	ATMEGA8, ATMEGA168, ATMEGA328, ATMEGA128 0	ARM, Cortex-A8	MX6Quad, 4xARM Cortex, A9 core Atmel SAM3X8E, ARM Cortex- M3 CPU
RAM	256-512 Мб	16-32 Кб	512Мб	1Гб
LAN(Mbit)	10/100	-	10/100	10/100/1000
Аналогові входи	0	6	6	14
ОС	Raspbian;OSMC; OpenELEC;Windows IoT Core; RISC OS.	Linux, Windows, MAC	Linux, Angstrom	Ubuntu, Android, Linux
Живлення	5V/USB	5V	5V	6-15V
Засоби програмування	C, C++, Java, Python	C, C++, Java, Python	Android	Android, C, C++, Java

Таким чином, існує багато різних мікроконтролерів і плат розробки, але основна перевага Arduino, що робить її кращим вибором для реалізації прототипу підсистеми моніторингу водопостачання – це доступність, оскільки апаратне забезпечення Arduino дешевше, ніж інші мікроконтролери та крім того, Arduino IDE можна завантажити безкоштовно.

В той же час, в якості сервера для розгортання платформи домашньої автоматизації можна рекомендувати Raspberry Pi [49].

Raspberry Pi 4 - мініатюрний одноплатний комп'ютер, який може повноцінно замінити десктопний ПК. Платформа підходить для розробки застосунків, може стати «мозком» для робота чи розумного будинку, медіацентром, вузлом розподіленої обчислювальної системи чи промисловим контролером. Raspberry Pi працює на різноманітних операційних системах, таких як: Raspbian; OSMC; OpenELEC; Windows IoT Core; RISC OS.

В Raspberry Pi 4 використовується однокристальна система Broadcom BCM2711. Кристал включає 4-ядерний 64-бітний процесор Cortex-A72 (ARM v8) з частотою 1,5 ГГц і графічний процесор GPU VideoCore VI з частотою 500 МГц. За даними виробника, система на новій архітектурі стала на 50% швидше, ніж минулі покоління Raspberry Pi, а також в ній збільшено оперативну пам'ять до 4 ГБ. На рисунку 3.2 зображено компоненти моделі Raspberry Pi 4 та їх розташування на платі [49].

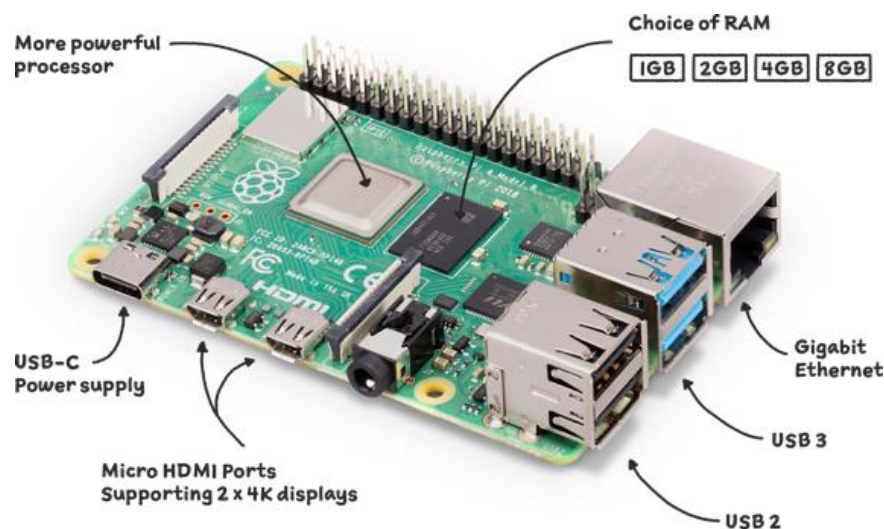


Рисунок 3.2 – Компоненти Raspberry Pi 4 [49]

3.2.2 Вибір плати Arduino

Arduino – це електронна платформа з відкритим вихідним кодом із спрощеним апаратним і програмним забезпеченням, що робить навчання легким для новачків, але достатньо гнучким, щоб використовувати його професіоналам. Апаратне забезпечення Arduino має відкритий код, що означає, що користувачі можуть створювати похідні плати або створювати новий продукт на основі дизайну Arduino. Засновники опублікували проекти та документацію плат на своєму вебсайті [50].

Різні варіанти Arduino мають одну спільну рису: мікроконтролер або інтегральну схему, яка служить мозком плати. Більшість плат Arduino складаються з 8-розрядних мікросхем від компанії ATmel, але плати з набагато вищою продуктивністю використовують 32-розрядні мікроконтролери, такі як ARM Cortex-M0+ і ARM Cortex-M3 від ARM і Intel Quark від Intel [50].

На платі є USB порт, який може бути як портом живлення, так і портом даних, гніздо для джерела живлення, світлодіодний індикатор живлення, кнопка скидання, регулятор напруги та світлодіоди TX/RX. Також є набір позначених контактів (пінів) для 5 В, 3,3 В, GND, аналогові, цифрові, ШІМ та AREF. Ці контакти в основному використовуються для підключення плат розширення або шилдів до Arduino для додаткових функцій, таких як підключення до мережі, РК-дисплей і джойстики. Піни також можна використовувати для кріплення компонентів з макетної плати для створення прототипів [50].

Існує багато різновидів плат Arduino, розроблених для різних типів користувачів і проектів. Arduino Uno, наприклад, підходить для початківців і простих проектів завдяки 14 контактам цифрового вводу/виводу. З іншого боку, Arduino Mega використовується для складних проектів із 54 контактами введення/виведення для взаємодії. Lilypad — це креативна версія Arduino, яку можна піддавати впливу води [50].

Всі плати можна умовно поділити на три основні категорії: плати початкового рівня, плати з розширеним функціоналом та плати Інтернету речей (рис. 3.3) [51].



Рисунок 3.3 – Категорії плат Arduino [51]

Плати Arduino початкового рівня є найкращим вибором для студентів. У цій категорії більшість плат мають низьку тактову частоту або обмежену кількість портів введення/виведення. Всі ці плати живляться від 8-розрядних мікроконтролерів. Більшість із них легко вивчати та створювати проєкти. На ринку доступні різноманітні модулі та плати розширення, особливо орієнтовані на ці плати базового рівня.

До них відносяться: Arduino UNO R3, Arduino Nano, Arduino PRO Mini, Arduino Leonardo, Arduino Micro, Arduino Nano Every, Arduino Mega2560 Rev3 [52]. Порівняльну характеристику плат цієї категорії наведено в таблиці 3.2 [51].

Плати Arduino з розширеними функціями призначені для проєктів, де потрібні розширені функції та висока продуктивність. До них відносяться: Arduino Nano 33 BLE, Arduino MKR Zero, Arduino UNO WIFI Rev2, Arduino Due, Arduino MKR Vidor 4000, Arduino Zero [51].

Плати IoT - це типи плат Arduino, які розроблені для підтримки технологій IoT, від простого комутатора до всіх побутових приладів: Arduino Nano 33 IoT, Arduino MKR FOX 1200, Arduino MKR WAN 1300/1310, Arduino MKR GSM 1400, Arduino MKR WIFI 1010, Arduino MKR NB 1500, Arduino Nano RP2040 Connect, Portenta H7 [51].

Таким чином, вибір плати Arduino повністю залежатиме від задач проєкту. Для початківця та проєкту, що не потребує великої обчислювальної потужності чи вбудованого зв'язку, можна вибрати найпопулярніші Arduino UNO або Arduino Nano, оскільки вони дешеві та прості в програмуванні, зокрема Arduino UNO підтримується відомими програмами моделювання такими як TINKERCAD, Simul та інші, тому саме цю плату було обрано для реалізації підсистеми моніторингу водопостачання в СРБ (рис. 3.4) [53].

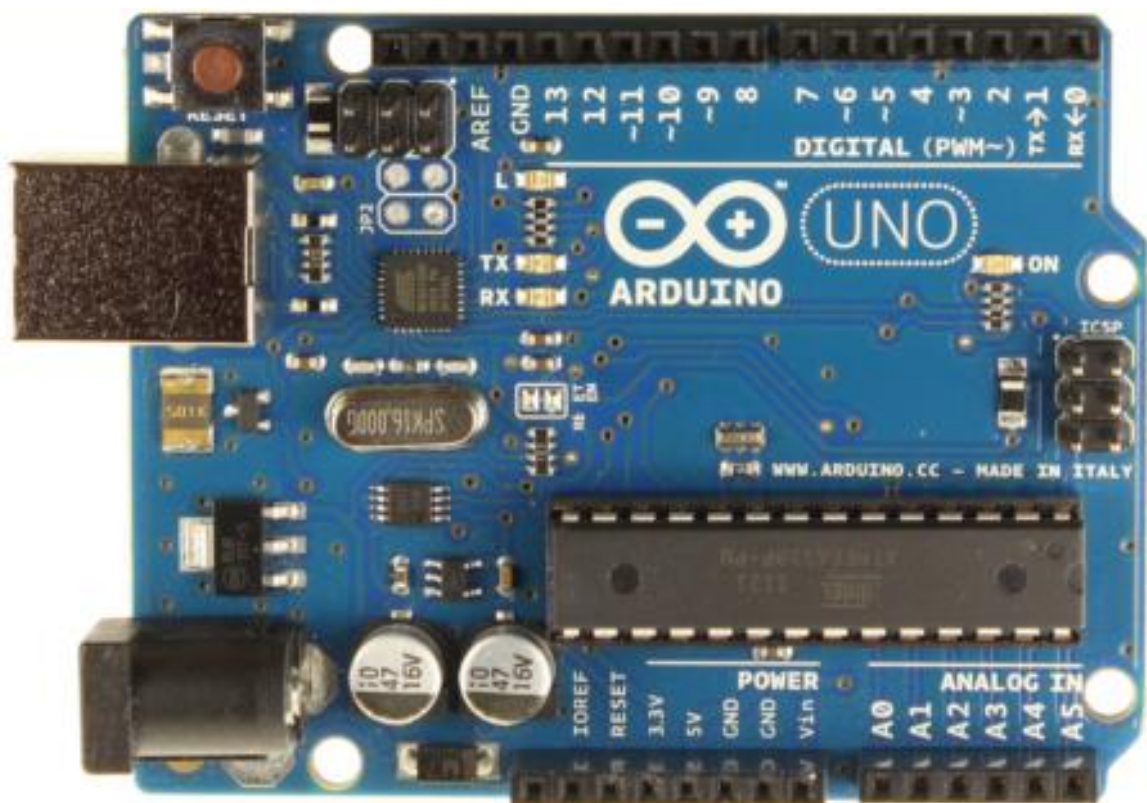


Рисунок 3.4 – Плата Arduino UNO [53]

Таблиця 3.2 – Порівняння плат Arduino початкового рівня [51]

Плата/ характеристика		Arduino UNO R3	Arduino Nano	Arduino PRO Mini	Arduino Leonardo	Arduino Micro	Arduino NanoEvery	Arduino Mega256 0 Rev3
Мікроконтролер		ATmega328 p	ATmega328p	ATmega328p	ATmega32u 4	ATmega32u4	ATmega4809	ATmega2 560
Тип USB		USB-B	Mini-B USB	Mini-B USB	Micro USB (USB-B)	Micro USB	Micro USB	USB-B
Піни вводу/виводу	Цифрові In/Out	14	14	14	20	20	14	54
	Аналогові In	6	8	8	12	12	8	16
Протоколи зв'язку		UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI
Тактова частота		16МГц	16МГц	16МГц	16МГц	16МГц	20МГц	16МГц
Flash пам'ять		32Кб	32Кб	32Кб	32Кб	32Кб	32Кб	32Кб
Живлення	Напруга	7-12 В	7-12 В	7-12 В	7-12 В	7-12 В	7-12 В	7-12 В
	Струм	20мА	20мА	20мА	10мА	10мА	20мА	20мА

3.2.3 Аналіз та вибір датчика рівня води

Як показали проведені дослідження, класифікацію датчиків рівня рідини можна виконати за декількома критеріями (рис. 3.5).

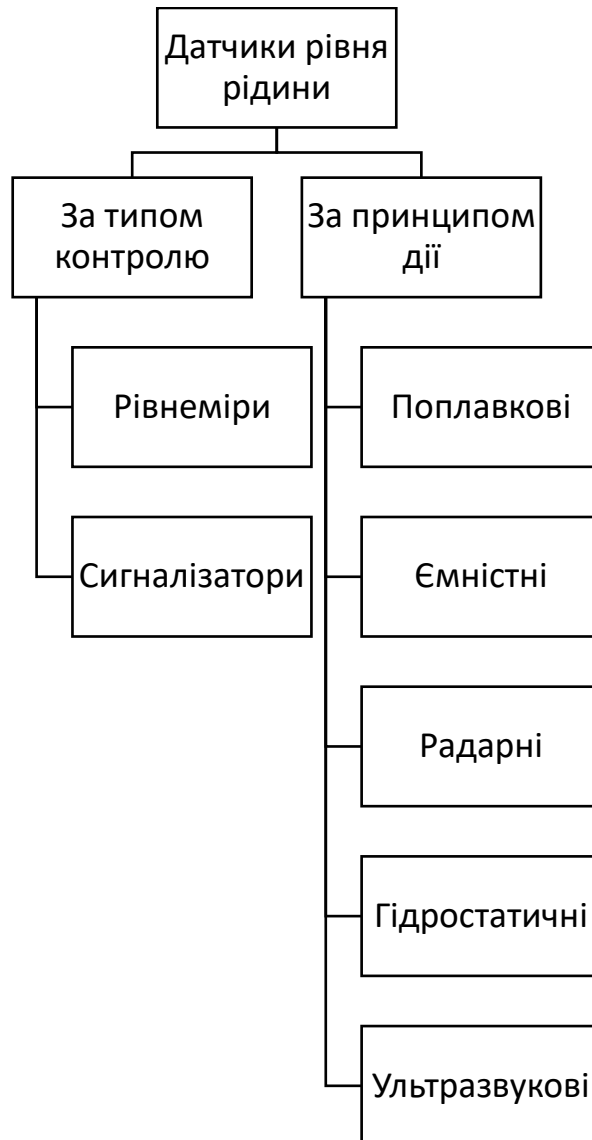


Рисунок 3.5 – Класифікація датчиків рівня рідини

За типом контролю – це рівнеміри та сигналізатори, тобто такі що забезпечують постійне вимірювання поточного рівня рідини, або контролюють лише певний рівень, наприклад, максимальний або мінімальний.

За принципом дії вони бувають таких типів [54]:

- поплавкові;
- ємнісні;

- радарного типу;
- гідростатичні;
- ультразвукові.

Ємнісний датчик працює як конденсатор, що змінює ємність при змінах в діелектрику. У таких пристроях немає рухомих елементів, тому вони довговічні. Під час експлуатації їх слід занурювати у рідину, а також компенсувати суттєву температурну залежність [54].

Поплавковий датчик рівня води може випускатися в декількох конфігураціях, для дискретного і безперервного сигналу (тобто демонструвати абсолютний рівень, або дискретний, тобто досягнення певного рівня). Дискретні поплавкові пристрої застосовуються для покрокових вимірів або як елемент автономного регулювання для систем з постійним рівнем рідини. Датчики, що видають безперервний сигнал, відносяться до магнітострикційних. Тут передбачений поплавець із постійним магнітом, завдяки чому вироби забезпечують підвищену точність [54].

Конструкція радарного датчика ґрунтується на радіолокації. Частота випромінювання локатора тут змінюється за лінійною ознакою. Сигнал обробляється у цифровому форматі, що нейтралізує сторонні шуми через поверхневі хвилювання рідини. У такому датчику немає рухомих елементів, вони не контактують з рідким середовищем, відрізняються високою точністю та універсальністю, проте досить дорого коштують [54].

Ультразвуковий датчик рівня рідини - це конструкція, що складається з генератора та приймача ультразвукових хвиль. Випромінювання проходить середовище і за швидкістю його розповсюдження обчислюється відстань до поверхні рідини [54].

Гідростатичні датчики вимірюють тиск стовпа рідини, а її рівень визначають залежно [54].

Електродні – застосовуються в рідких електропровідних середовищах. Складаються з двох електродів, які контролюють необхідний рівень рідини.

Третій електрод є аварійним і використовується при перевищенні заданих параметрів і включення режиму відкачування [54].

Критерії вибору датчика [55]:

- тип рідини, її щільність, рівень небезпеки для людини;
- матеріал та габарити ємності (наприклад бак, резервуар);
- вид контролю – перевірка макс/мін рівня рідини або постійне відстеження рівня;
- точність вимірювання.

При виборі промислових датчиків кількість критеріїв може бути розширено, для задач домашньої автоматизації достатньо враховувати габарити резервуара і тип контролю.

Датчик рівня рідини для Arduino (рис. 3.6) призначений для визначення рівня рідини в різних ємностях, де недоступний візуальний контроль, з метою попередження перенаповнення ємності рідиною через критичну позначку. Досліджуваний датчик можна використовувати для вимірювання рівня води (наприклад, у пральних машинах, баках), контролю відстійника, виявлення дощу чи витоку, а також факту затоплення приміщення [56].

Даний датчик містить ряд з п'яти живлячих та п'яти чутливих доріжок, які чергуються таким чином, що між кожними двома живлячими доріжками є одна чутлива доріжка. При зануренні ці доріжки з'єднуються водою між собою [56].

Також на платі розташовані (рис. 3.6):

- індикатор живлення, який загоряється під час подачі на плати напруги живлення;
- вихід Signal – це аналоговий вихід, який буде підключений до одного з аналогових входів плати Arduino;
- вихід VCC забезпечує живлення датчика (рекомендується напруг від 3,3 до 5 В);
- GND - земля [56].

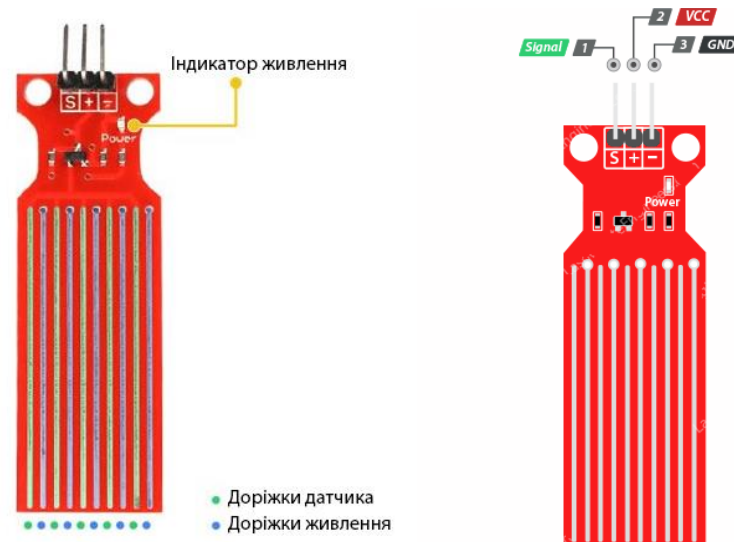


Рисунок 3.6 – Конструкція датчика води для Arduino [56]

Робота датчика рівня води базується на тому, що набір відкритих паралельних провідників діє як змінний резистор (потенціометр), опір якого змінюється в залежності від рівня води. Зміна опору відповідає відстані верхівки датчика до поверхні води. Опір обернено пропорційно висоті води [56]:

- чим більше води, в яку занурений датчик, тим краще провідність, і тим нижчий опір;
- чим менше води, в яку занурений датчик, тим гірша провідність, і тим вищий опір.

Датчик відповідно до опору видає вихідну напругу, вимірюючи яку ми можемо визначити рівень води [56].

Оскільки в якості платформи розробки було обрано плату Arduino, тому цей датчик є оптимальним варіантом для створення та дослідження прототипу підсистеми моніторингу водопостачання СРБ.

3.3 Аналіз та вибір програмних засобів розробки

3.3.1 Вибір середовища програмування

Програмування для обраних плат може здійснюватись різними мовами: Java, C++, Python.

Java – це мова програмування, яку розробники використовують для створення різноманітних програм. Java також має вебплагін, який дозволяє запускати програми у браузері [57].

C++ – це мова об'єктно-орієнтованого програмування, що є розширенням мови C. Тому можна кодувати C++ у «стилі C» або «об'єктно-орієнтованому стилі» [58].

Python – широко використовувана динамічна мова програмування високого рівня загального призначення з інтерпретацією. Її філософія дизайну підкреслює читабельність коду, а синтаксис дозволяє програмістам виражати концепції меншою кількістю рядків коду, ніж це можливо в таких мовах, як C++ або Java. Мова надає конструкції, призначені для програм як малого, так і великого масштабу [59].

Порівняльну характеристика цих трьох мов програмування наведено в таблиці 3.3.

Таблиця 3.3 – Порівняння мов програмування для обраних плат [57] - [59]

Критерій	Python	Java	C++
Швидкодія	+	++	++
Час розробки	+	++	+++
Довжина коду	+	++	+++
Методи відлагодження	+	+	+

Щодо швидкодії, Python має нижчу швидкодію ніж Java та C++. Але програмування на Python зазвичай потребує менше часу ніж програмування на Java та особливо на C++. Кількість рядків коду програми на C++ набагато

більша ніж на Python та Java. Методи відлагодження для всіх мов достатньо прості.

Тим не менш, основною мовою програмування, яка використовується для кодування Arduino, вважається C++. Arduino надає офіційне інтегроване середовище розробки Arduino Integrated Development Environment (IDE), що має мінімалістичний інтерфейс та дозволяє легко створювати, компілювати та завантажувати код на плату Arduino. Це універсальне програмне забезпечення, сумісне з MAC, Windows і Linux. Як і апаратне забезпечення, програмне забезпечення Arduino має відкритий для розробки код, що дозволяє полегшити роботу як новачкам, так і досвідченим користувачам. [60].

Багато користувачів, особливо початківці, зазвичай вибирають Arduino IDE для програмування мікроконтролера. Тим не менш, інші платформи, схожі на Arduino IDE, наприклад PlatformIO, також можна використовувати для програмування Arduino. Порівняльну характеристику цих двох середовищ розробки наведено в таблиці 3.4.

Таблиця 3.4 – Порівняння середовищ розробки для Arduino [60] - [61]

Критерій	Arduino IDE	PlatformIO
Зрозумілість	+	+
Підтримка	+	+
Мова	Java	Python
Операційна система	Windows, OS X, Linux	Windows, macOS, Linux, FreeBSD
Бібліотечний менеджер	-	+
Використання плагінів	-	+
Доступність вебверсії	+	-
Відкритість	Open-source	Open-source

Аналіз показав, що Arduino IDE має вебверсію, а також надає велику бібліотеку кодів, які користувачі можуть використовувати, змінювати або доповнювати, щоб уникнути розробки з нуля. Це полегшує навчання початківцям та заощаджує час професіоналам [62].

Тому, в якості середовища розробки програмного забезпечення для Arduino доцільно обрати саме оригінальне середовище від розробника Arduino IDE.

3.3.2 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. Але Home Assistant [63] та OpenHAB [64] - [65] є найпотужнішими учасниками спільноти.

Обидва ці проекти мають однакову мету та об'єднують велику кількість пристроїв. Використання OpenHAB вимагає знання команд введення, а процес інтеграції пристрої досить складний і тривалий. З іншого боку, Home Assistant легше у використанні, але потребує значної роботи для налаштування. Мобільні програми, здається, мають меншу гнучкість і виглядають досить складними та складними, особливо для новачків. Domoticz надає достатню кількість функцій; більшість його налаштувань виконується через Інтернет, а плагіни реалізовано для збільшення можливостей ПЗ. Інтерфейс програми нелегко зрозуміти на основі інформації про налаштування, яка пропонується Domoticz. Дві французькі компанії, які є основними учасниками екосистеми домашньої автоматизації з відкритим кодом, це Calaos і Jeedom. На жаль, більшість їх форумів і спільнот лише французькою мовою, що часто є мовним бар'єром [66].

Платформи можна порівняти за декількома ознаками, включаючи мову програмування, яка використовується для розробки, API, кількість розроблених плагінів і протоколів, кількість і різноманітність документації.

У таблиці 3.5 наведено порівняльний аналіз найважливіших систем домашньої автоматизації з відкритим кодом у їхніх наборах функцій [66].

Таблиця 3.5 – Порівняльний аналіз платформ домашньої автоматизації [66]

Платформа	API	Мова програмування	Кількість протоколів	MQTT	Вебінтерфейс
OpenHAB	REST	Java	+	+	+
Domoticz	JSON	C++	-	+	+
Jeedom	HTTP, JSON	PHP	+	-	+
Home Assistant	Python, REST	Python	+	+	+

OpenHAB – це популярна відкрита платформа домашньої автоматизації, незалежна від технологій, яка працює як центр керування СРБ. OpenHAB здійснює зв'язок з «розумними» пристроями, виконує задані користувачем дії та надає вебсторінки із заданою користувачем інформацією, а також задані користувачем інструменти для взаємодії з усіма пристроями. Для досягнення цієї мети OpenHAB сегментує та розділяє певні функції та операції [64] - [65].

До переваг OpenHAB відносяться:

- здатність інтегрувати безліч пристроїв та систем. OpenHAB включає інші системи домашньої автоматизації, «розумні» пристрої та інші технології в єдине рішення;
- забезпечення єдиного інтерфейсу користувача та загального підходу до правил автоматизації для всієї системи, незалежно від кількості задіяних виробників і підсистем;
- надає найбільш гнучкий інструмент, що дозволяє втілити практично будь-яке бажання в області домашньої автоматизації [64] - [65].

Тому, в якості платформи для розгортання СРБ на мінікомп'ютері Raspberry Pi можна рекомендувати саме OpenHAB.

3.4 Висновки до розділу 3

В цьому розділі відповідно до завдань дипломної роботи виконано аналіз функціональних та нефункціональних вимог до проєктованої підсистеми моніторингу водопостачання СРБ, показано її місце в архітектурі СРБ.

Проведено аналіз апаратних та програмних засобів розробки.

Обґрунтовано вибір платформи Raspberry Pi в якості серверу СРБ, а також Arduino в якості плати для підключення сенсорів для моніторингу водопостачання.

Виконано огляд існуючих сенсорів води та обрано датчик для Arduino.

В якості мови програмування для Arduino обрано C++, як середовище розробки – Arduino IDE, а як платформу домашньої автоматизації - OpenHAB.

4 КОНСТРУЮВАННЯ ПІДСИСТЕМИ МОНІТОРИНГУ ВОДОПОСТАЧАННЯ

Розроблюване програмне забезпечення інтегрується в СРБ та об'єднує дві складові: підсистему моніторингу на основі Arduino, що складається з набору модулів та підсистему нотифікації на основі конфігураційних файлів для платформи OpenHAB та месенджеру Telegram (рис.4.1). В наступних підрозділах описано реалізацію складових ПЗ.



Рисунок 4.1 – Структура програмного забезпечення

4.1 Розробка підсистеми моніторингу водопостачання

4.1.1 Моделювання роботи прототипу в середовищі Tinkercad

Створений приклад моделювання підсистеми моніторингу на основі датчика вологості в середовищі Tinkercad наведено на схемі (рис. 4.2). Для

того, щоб працювати з датчиком вологості у режимі симуляції в середовищі Tinkercad, необхідно натиснути на датчик та на області, що з'явилася, змінювати положення бігунка вологості курсором миші. У наведеному прикладі, під час підвищення вологості більше, ніж на 50%, світлодіод починає світитися. Код програми наведено в Додатку А.1.

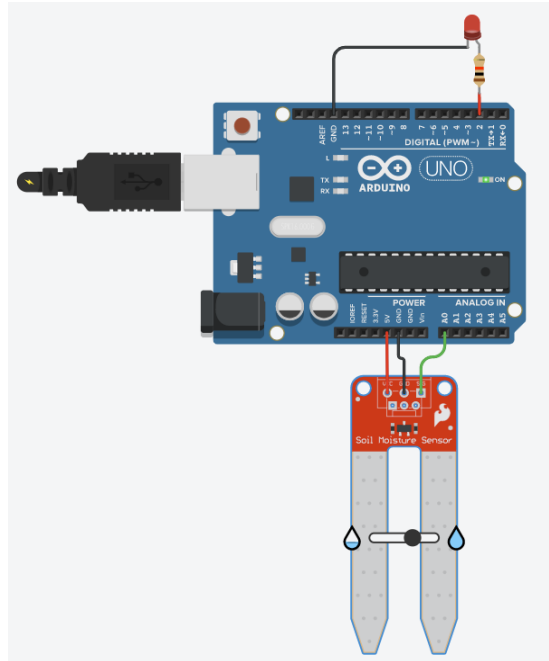


Рисунок 4.2 – Дослідження роботи датчика вологості в Tinkercad

4.1.2 Розробка базового модуля

Схему підключення для базового модуля наведено на рисунку. 4.3.

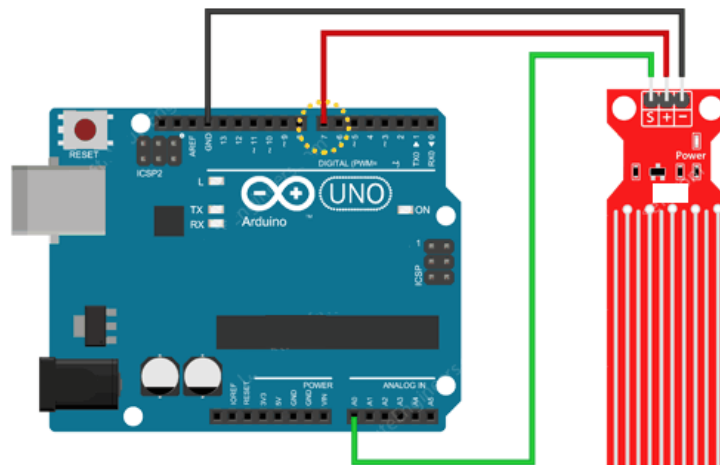


Рисунок 4.3 – Схема підключення для базового модуля

Текст програми наведено в Додатку А.2.

Результат роботи програми (рис. 4.4) можна побачити у вікні монітора послідовного порту Arduino. Значення дорівнює 0, коли датчик нічого не торкається; приблизно 420, коли датчик занурений у воду частково, приблизно 520 коли датчик занурений повністю.



Рисунок 4.4 – Виведення результатів роботи програми у монітор послідовного порту

4.1.3 Розробка модуля автоматичного вмикання/вимикання

Для автоматичного вмикання/вимикання подачі води необхідно додати в схему модуль реле 5В 10А та постійний резистор 10 КОм потужністю 0,25 Вт.

5В 1-канальний модуль реле використовує 5-20мА для спрацьовування, тобто керується безпосередньо з пінів мікроконтролера Arduino або інших. Включається логічним нулем, вимикається логічною одиницею. На модулі представлено два світлодіоди: червоний показує наявність напруги живлення, зелений сигналізує про спрацьовування реле (рис. 4.5).



Рисунок 4.5 – Реле [56]

Вигляд реального прототипу модуля автоматичного вмикання/вимикання подачі води наведено на рисунку 4.6. Текст програми наведено в Додатку А.3.

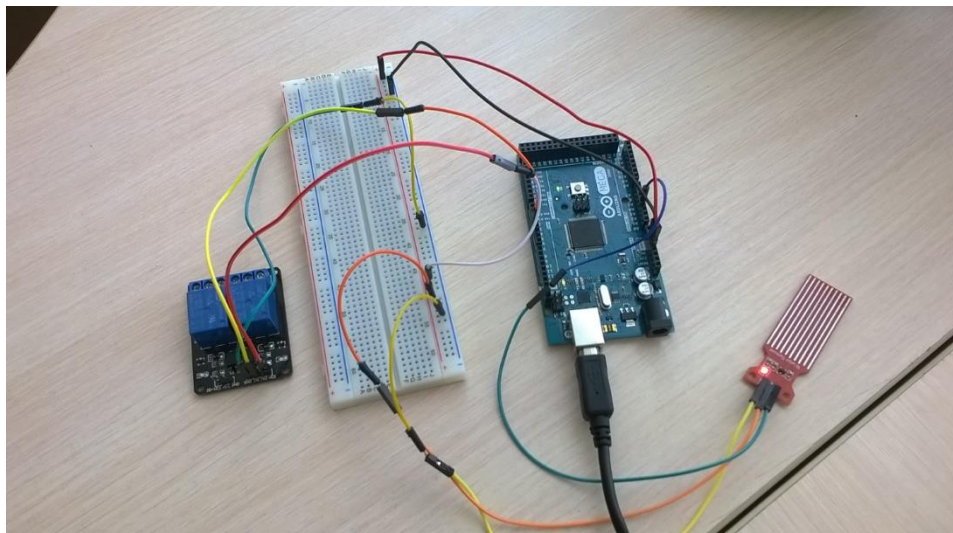


Рисунок 4.6 – Прототип модуля автоматичного вмикання/вимикання води

4.1.4 Розробка модуля звукового сповіщення

Алгоритм звукової сигналізації затоплення приміщення передбачає, що при зануренні датчика у воду сигналізація видає три види звукових сигналів (невелике затоплення, середній рівень, критичний рівень), що відповідає трьом рівням води. Для реалізації сповіщення, необхідно до цифрового виходу підключити п'єзовипромінювач - електроакустичний пристрій відтворення звуку, а також до цифрового виводу Arduino через транзистор підключити динамік для забезпечення достатнього рівня гучності.

Для організації звукового сповіщення необхідно додати в схему динамік 8 Ом, резистор 500 Ом, транзистор KT503є. Схему підключення наведено на рисунку. 4.7.

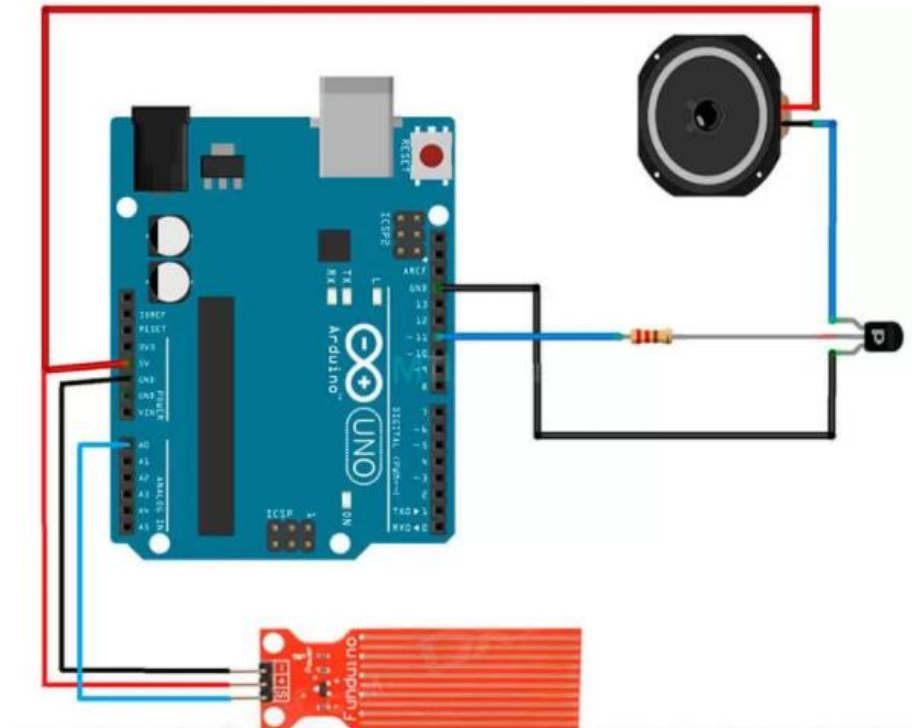


Рисунок 4.7 – Схема підключення для модуля звукового сповіщення

Аналоговий вихід датчика підключений до аналогового входу Arduino, який представляє собою аналого-цифровий перетворювач (АЦП) з роздільною здатністю 10 біт, що дозволяє на виході отримувати значення від 0 до 1023. Значення аналогових сигналів на аналоговому вході Arduino для трьох рівнів занурення були визначені так: > 400 - мінімальне занурення; > 500 - середній рівень занурення; > 600 - велике занурення. Відповідно для кожного рівня занурення на динаміці відтворюється звуковий сигнал різної частоти: мінімальне занурення - 293 Гц (нота ре 1 октави); середній рівень занурення - 466 Гц (нота сі-бімоль 1 октави); велике занурення - 587 Гц (нота ре 2 октави). При відсутності занурення звуковий сигнал на динаміці не відтворюється.

Текст програми наведено в Додатку А.4.

4.1.5 Розробка модуля світлового сповіщення

За основу можна взяти схему на рис. 4.3 та додати до неї світлодіоди (рис. 4.8).

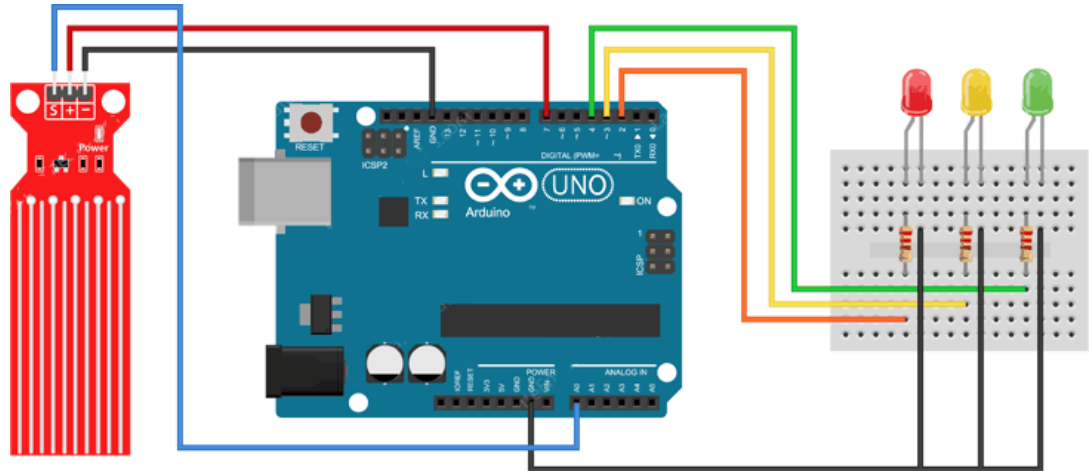


Рисунок 4.8 – Схема підключення для модуля світлового сповіщення

Текст програми з коментарями наведено в Додатку А.5. В програмі оголошено дві змінні `lowerThreshold` і `upperThreshold`, що становлять порогові рівні:

- значення, що нижче за нижній поріг, включає червоний світлодіод;
- значення, що вище за верхній поріг, включає зелений світлодіод;
- значення, що знаходиться між ними, включає жовтий світлодіод.

Опис базових функцій ПЗ підсистеми моніторингу наведено в таблиці 4.1.

Таблиця 4.1 – Функції програми

Функція	Опис
1	2
<code>setup()</code>	виконується лише раз при старті програми і дозволяє задати початкові параметри
<code>loop()</code>	виконується періодично, доки плата не буде вимкнена
<code>pinMode(pin, mode)</code>	Функція <code>pinMode()</code> встановлює режим портів: <code>pin</code> – вивід, з яким ми збираємося працювати; <code>mode</code> - як повинен працювати вказаний вивід: працювати на вихід (OUTPUT) чи вхід (INPUT), або INPUT_PULLUP

Продовження таблиці 4.1

1	2
<code>analogRead(pinA)</code>	Повертає ціле значення в межах від 0 до 1023, що є пропорційним напрузі, поданій на аналоговий вхід, номер якого передається як в якості параметра <code>pinA</code> функції.
<code>analogWrite(pin, value)</code>	Встановлює аналогове значення ШІМ для виведення. Використовується для плавної зміни кольору світлодіода або швидкості обертання мотора. Піни з підтримкою PWM позначені на платі символом тильда (~). Для Arduino Uno виводи мають номери 3, 5, 6, 9, 10 та 11.
<code>digitalRead(pin)</code>	Зчитує показання з цифрового виводу. Повертається HIGH (високе, 1) або LOW (низьке, 0):
<code>digitalWrite(pin, value)</code>	Функція <code>digitalWrite()</code> приймає два параметри, але не повертає значення. Відповідно, <code>pin</code> — номер цифрового порту, на який ми надсилаємо сигнал; <code>value</code> – значення, яке ми відправляємо на порт. Для цифрових пінів можливі два значення: або HIGH (високе, 1) або LOW (низьке, 0)
<code>delay(ms)</code>	Призупиняє роботу програми на заданий проміжок часу, а саме певну кількість мілісекунд.
<code>Serial.begin(speed)</code>	Задає послідовне з'єднання і швидкість передачі даних, що вимірюється в біт за секунду
<code>Serial.available()</code>	Приймає ту кількість символів, що доступні для зчитування з Serial порту, тобто ті, що вже отримані та записані в його буфер.
<code>Serial.read()</code>	Зчитує черговий доступний байт з буфера послідовного з'єднання
<code>Serial.write()</code>	Записує дані до послідовного порту; дані надсилаються як байт або послідовність байт.
<code>Serial.flush()</code>	Очищає буфер послідовного порта
<code>Serial.println (val, format)</code>	Друкує дані, що надходять із серійного порту у вигляді ASCII-тексту: <code>val</code> – значення для друку; <code>format</code> - формат даних, що виводяться.
<code>tone(pin, frequency, duration)</code>	Генерує звук заданої частоти на вказаному піні. Можна виводити лише одну ноту під час виконання: <code>pin</code> – виведення плати для відтворення звуку; <code>frequency</code> - частота звуку в Гц; <code>duration</code> - тривалість у мілісекундах.

4.1.6 Алгоритм функціонування підсистеми моніторингу

Узагальнений алгоритм функціонування підсистеми моніторингу водопостачання наведено на рисунку 4.9.



Рисунок 4.9 – Узагальнений алгоритм роботи підсистеми моніторингу

Підсистема дозволяє контролеру активувати водяний насос на основі показників рівня води, які отримані від датчика, а також увімкнути звукову або світлову індикацію рівня води.

Блок 1. Початок.

Блок 2. Задання граничних значень показчиків рівня води.

Блок 3. Запитування даних з датчиків

Блок 4. Контроль зчитаних з датчиків даних на відповідність заданим граничним величинам.

Блок 5. Якщо виміряні значення не відповідають заданим показникам рівня води, активується робота насоса.

Блок 6. Якщо виміряні значення не відповідають заданим показникам рівня води, вмикається звукове/світлове сповіщення.

Блок 7. Якщо зчитування даних з датчика виконується циклічно, то перехід на Блок 3, інакше - на Блок 8.

Блок 8. Кінець.

4.2 Розробка підсистеми нотифікації на платформі OpenHAB

Узагальнений алгоритм розробки підсистеми нотифікації представлено на рисунку 4.10.

Блок 1. Початок.

Блок 2. Налаштування OpenHAB. Передбачає встановлення клієнта та налаштування сервера на Raspberry Pi.

Блок 3. Підключення до Telegram . Передбачає встановлення аддонів OpenHAB, зокрема “Telegram Binding”, що надає можливість комунікації з месенджером Telegram через створених ботів цієї платформи.

Блок 4. Підключення до myopenhab. Надає можливість віддаленого керування за допомогою мобільного застосунку OpenHAB.

Блок 5. Віддалене керування та отримання повідомлень. Надає можливість отримувати інформацію про показники та статуси в інтерфейсну форму OpenHAB, в мобільний застосунок OpenHAB та в Telegram.

Блок 5. Кінець.



Рисунок 4.10 – Узагальнений алгоритм розробки підсистеми нотифікації

4.2.1 Налаштування OpenHAB для роботи на Raspberry Pi

Для встановлення OpenHAB на Raspberry Pi з заздалегідь встановленою ОС Raspbian слід виконати послідовність дій та команд (за допомогою терміналу), що описані нижче.

Крок 1. Завантажити ключ:

```
wget -qO - 'https://bintray.com/user/downloadSubjectPublicKey?username=openhab' | sudo apt-key add -
```

Крок 2. Встановити репозиторій OpenHAB:

```
echo "deb http://dl.bintray.com/openhab/apt-repo stable main" | sudo tee /etc/apt/sources.list.d/openhab.list
```

Крок 3. Оновити системні файли: `sudo apt-get update`

Крок 4. Встановити OpenHAB: `sudo apt-get install openhab-runtime`

Крок 5. Додати користувача до групи: `sudo usermod -aG openhab pi`

Крок 6. Запустити OpenHAB: `sudo systemctl enable openhab`

Крок 7. Встановити доповнення для роботи з GPIO:

```
sudo apt-get install openhab-addon-io-gpio
```

```
sudo apt-get install openhab-addon-binding-gpio
```

Крок 8. Встановити доповнення для роботи з портами:

```
sudo apt-get install openhab-addons-binding-serial
```

Крок 9. Задати дані користувача:

```
sudo nano /etc/default/openhab
```

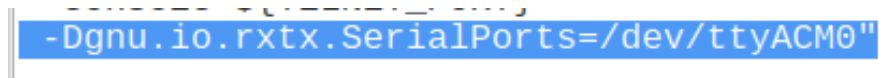
```
sudo nano /usr/lib/systemd/system/openhab.service
```

Крок 10. Перезавантажити systemd: `sudo systemctl daemon-reload`

Крок 11. Додати порт, який використовуватиметься з Arduino:

```
sudo nano /usr/share/openhab/bin/openhab.sh
```

В розділі «JAVA_ARGS_DEFAULT» додати в кінець рядок як на рисунку 4.11.



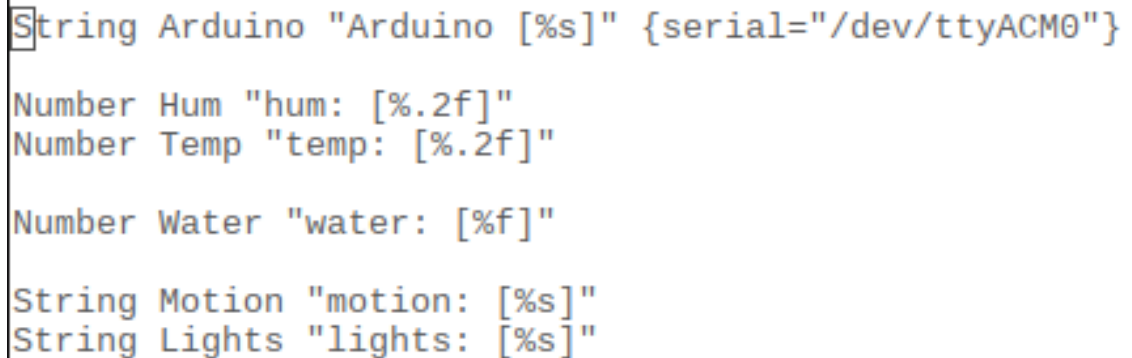
```
-Dgnu.io.rxtx.SerialPorts=/dev/ttyACM0"
```

Рисунок 4.11 – Додавання порту

Крок 12. Перевантажити OpenHAB: `sudo service openhab restart`

Крок 13. Написати код для отримання даних від Arduino (рис.4.12):

`sudo nano /etc/openhab/configurations/items/home.items`

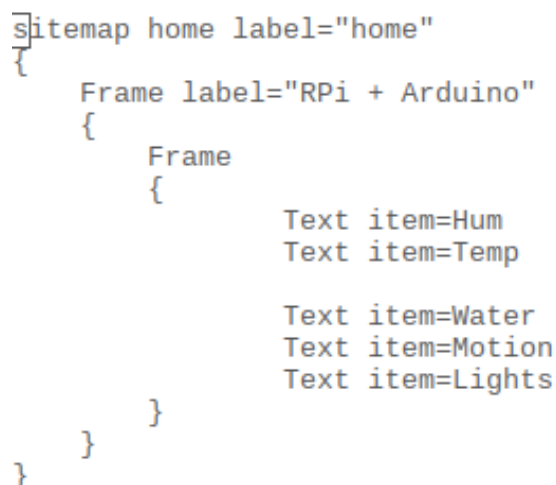


```
String Arduino "Arduino [%s]" {serial="/dev/ttyACM0"}
Number Hum "hum: [%2f]"
Number Temp "temp: [%2f]"
Number Water "water: [%f]"
String Motion "motion: [%s]"
String Lights "lights: [%s]"
```

Рисунок 4.12 – Код отримання даних від Arduino

Крок 14. Створити інтерфейс для відображення даних (рис.4.13):

`sudo nano /etc/openhab/configurations/sitemaps/home.sitemap`



```
sitemap home label="home"
{
    Frame label="RPi + Arduino"
    {
        Frame
        {
            Text item=Hum
            Text item=Temp

            Text item=Water
            Text item=Motion
            Text item=Lights
        }
    }
}
```

Рисунок 4.13 – Створення інтерфейсу для відображення даних

Крок 15. Перевантажити OpenHAB: `sudo service openhab restart`

Крок 16. Знайти процес Java та запам'ятати його номер: `top` (рис. 4.14).

```
pi@raspberrypi:~ $ top
top - 12:49:29 up 1:58, 2 users, load average: 1.09, 0.53, 0.29
Tasks: 146 total, 1 running, 145 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.9 us, 0.4 sy, 0.0 ni, 98.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 949576 total, 37712 free, 263564 used, 648300 buff/cache
KiB Swap: 102396 total, 102396 free, 0 used. 599936 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 3349 pi        20   0   8100  3244  2780 R   2.0   0.3   0:00.24 top
 3234 pi        20   0 365756 91912 12496 S   1.3   9.7   0:46.82 java
```

Рисунок 4.14 – Відображення процесів

Крок 17. Зупинити процес Java. Наприклад, номер Java процесу – 3234, тоді: `kill 3234`

Крок 18. Перейменувати файл `openhab_default.cfg` у `openhab.cfg` в каталозі `/etc/openhab/configurations/` (рис.4.15).

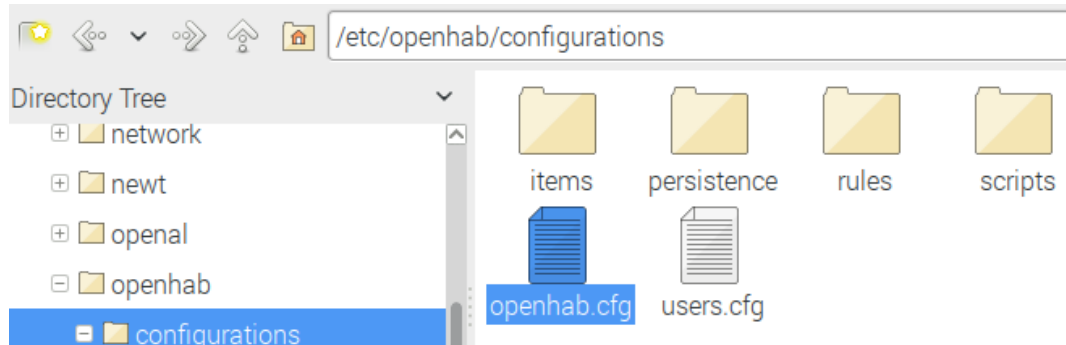


Рисунок 4.15 – Файл перейменовано

Крок 20. Створити правило OpenHAB для керування датчиками (рис.4.16): `sudo nano etc/openhab/configurations/rules/home.rules`

Крок 21. Підключити Arduino із завантаженим скетчем.

Крок 22. Надати повний доступ до порту, що буде використовуватися Arduino: `chmod 777 /dev/ttyACM0`

Крок 23. Перейти до каталогу openHab: `cd /usr/share/openhab/bin`

Крок 24. Запустити файл `.sh`: `sudo ./openhab.sh`

```

import java.io.*
rule "Arduino"
when
    Item Arduino received update
then
    var String ArdUpd=Arduino.state.toString().split("\n")

    for (String s:ArdUpd){
        var String[] ar=s.split(",")

        var Double h = new Double(ar.get(0))
        var Double t=new Double(ar.get(1))

        var Double w=new Integer(ar.get(2))

        var String m=new String(ar.get(3))
        var String l=new String(ar.get(4))

        Hum.postUpdate(h)
        Temp.postUpdate(t)

        Water.postUpdate(w)
        Motion.postUpdate(m)
        Lights.postUpdate(l)

    }
end

```

Рисунок 4.16 – Код правила

Крок 25. Перейти за адресою: RaspberryPi-IP:8080/openhab.app?sitemap=home (рис.4.17).

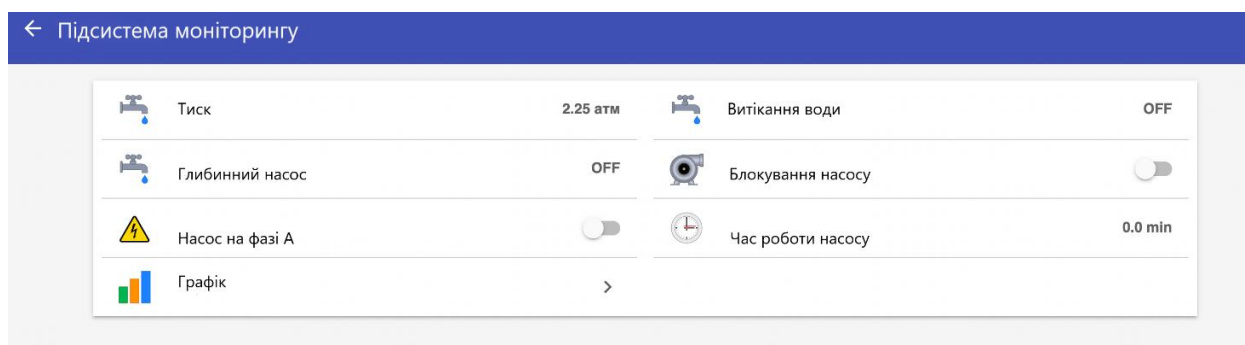


Рисунок 4.17 – Сторінка OpenHAB

4.2.2 Налаштування OpenHAB для віддаленого керування і отримання повідомлень

Серед основних додаткових можливостей OpenHAB для віддаленого керування і отримання повідомлень було обрано: підключення до Telegram; підключення myopenhab.

Підключення до Telegram дозволить відправляти повідомлення клієнтам Telegram з бот-клієнта. Для підключення ботів до OpenHAB потрібні маркери авторизації і ідентифікатори чатів. Згідно Telegram Bot API щоб отримати токен авторизації і ідентифікатор чату потрібно виконати наступні кроки:

Крок 1. Створити бота і отримати токен:

- у клієнтському застосунку Telegram почати чат з BotFather;
- написати / newbot для створення бота, ввести ім'я бота і отримати токен авторизації.

Крок 2. Створити чат з обраним користувачем і отримати ідентифікатор чату.

- почати чат з новим ботом і відправити йому повідомлення;
- відкрити браузер і відправити запит виду: <https://api.telegram.org/bot<token>/getUpdates> (де <token> це щойно отриманий токен авторизації);
- переглянути JSON результат і в ньому знайти значення поля result [0] .message.from.id - це і буде ідентифікатор чату.

Крок 3. Підключити цих ботів до OpenHAB, для чого потрібно додати інформацію в файл openhab.cfg, наприклад, для підключення двох ботів потрібно дописати в файл таку інформацію:

```
telegram: bots = bot1, bot2
telegram: bot1.chatId = 22334455
telegram: bot1.token = xxxxxxxxxxxx
telegram: bot2.chatId = 654321
telegram: bot2.token = yyyyyyyyyyyy
```

Крок 4. Після цього можна відправляти повідомлення з OpenHAB (рис.4.18).

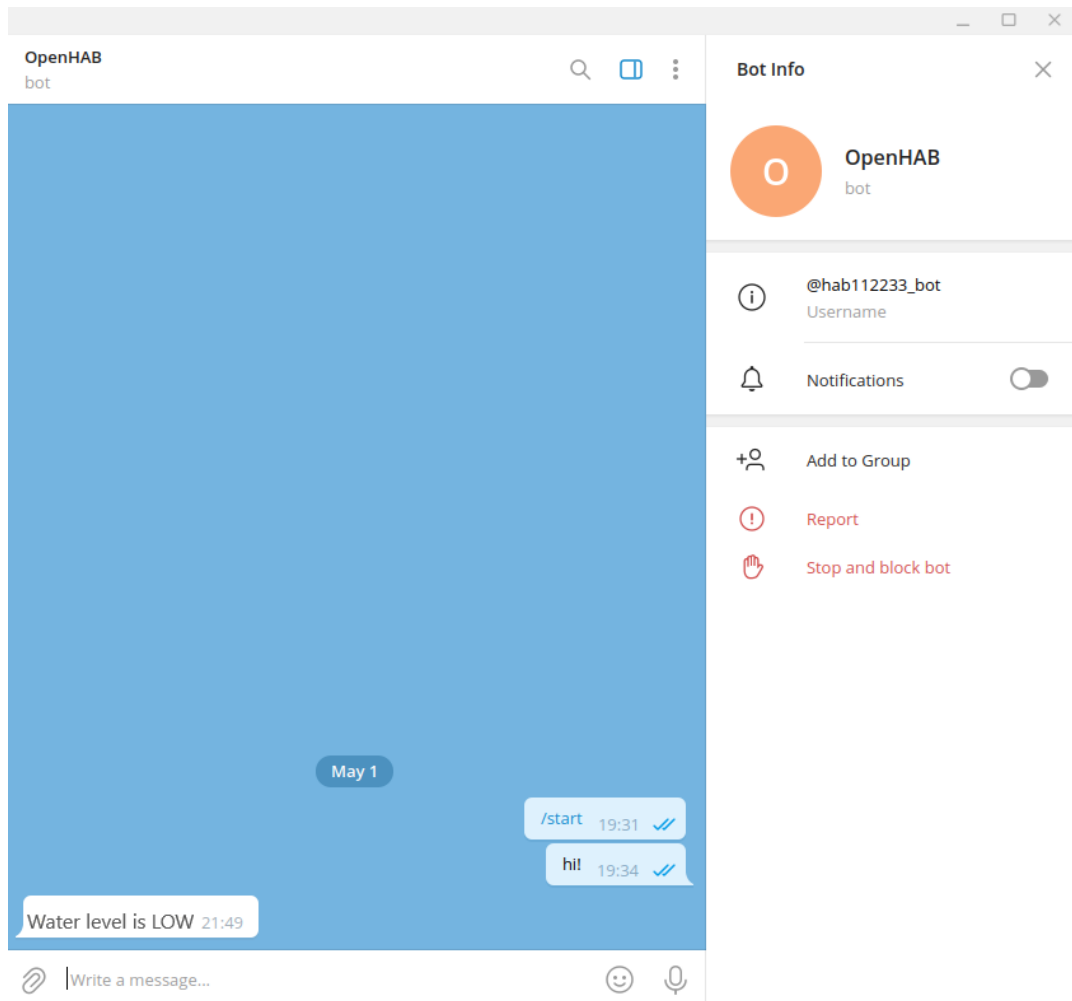


Рисунок 4.18 – Чат-бот

Для цього в .rules потрібно додати правило такого виду:

```
rule "Send telegram with Fixed Message"
when
    Item Foo changed
then
    sendTelegram ( "bot1", "item Foo changed")
end
```

Для підключення до myopenhab в OpenHAB знаходяться два згенерованих файли, один з яких містить UUID користувача, а другий випадковий секретний ключ. Вони розташовані в директорії webapps / static

всередині директорії OpenHAB з іменами `uuid` і `secret`. Далі потрібно перейти на сторінку реєстрації: <https://myopenhab.org/login> і ввести інформацію з файлів `uuid` і `secret` у відповідних полях. Після реєстрації для віддаленого доступу можна користуватися посиланням виду:

<https://myopenhab.org/openhab.app?sitemap=yoursitemap>

Оскільки `my.openHAB` його не передає дані аутентифікації через Інтернет з'єднання з `openHAB` з міркувань безпеки, потрібно відключити локальну аутентифікацію, для можливості відправлення запитів до `openHAB`. Для цього необхідно в файлі `openhab.cfg` встановити `security: option = EXTERNAL` і налаштувати `security: netmask =` на маску локальної підмережі. Після цього `my.openHAB` матиме можливість робити запити, в той час як, інші зовнішні запити будуть проходити перевірку автентичності користувачів задану у файлі `users.cfg`.

Завдяки підключенню системи до `myopenhab` було протестовано віддалене керування обладнанням для контролю водопостачання за допомогою мобільного застосунку OpenHAB (рис. 4.19).

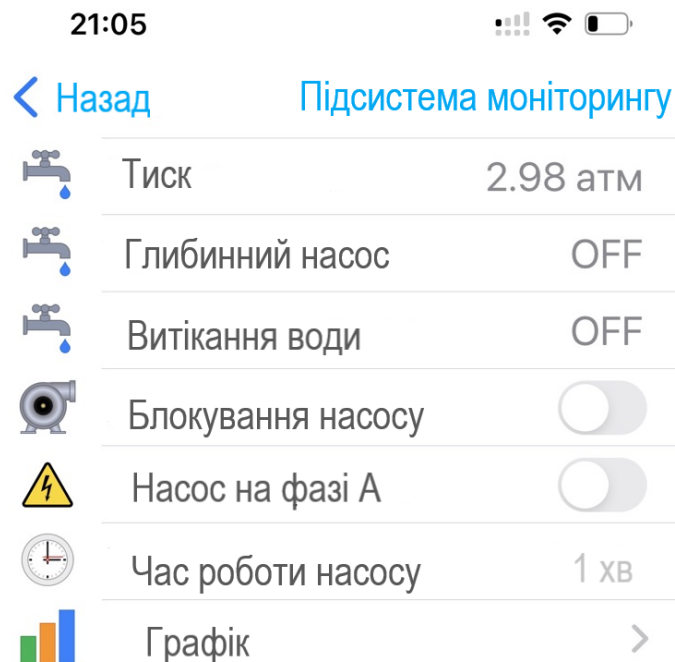


Рисунок 4.19 – Відображення інформації в мобільному застосунку OpenHAB

На рисунку 4.20 наведено приклади графіків, що будуються на основі даних та засобами OpenHAB та ілюструють показники процесу та статуси обладнання водопостачання в СРБ.

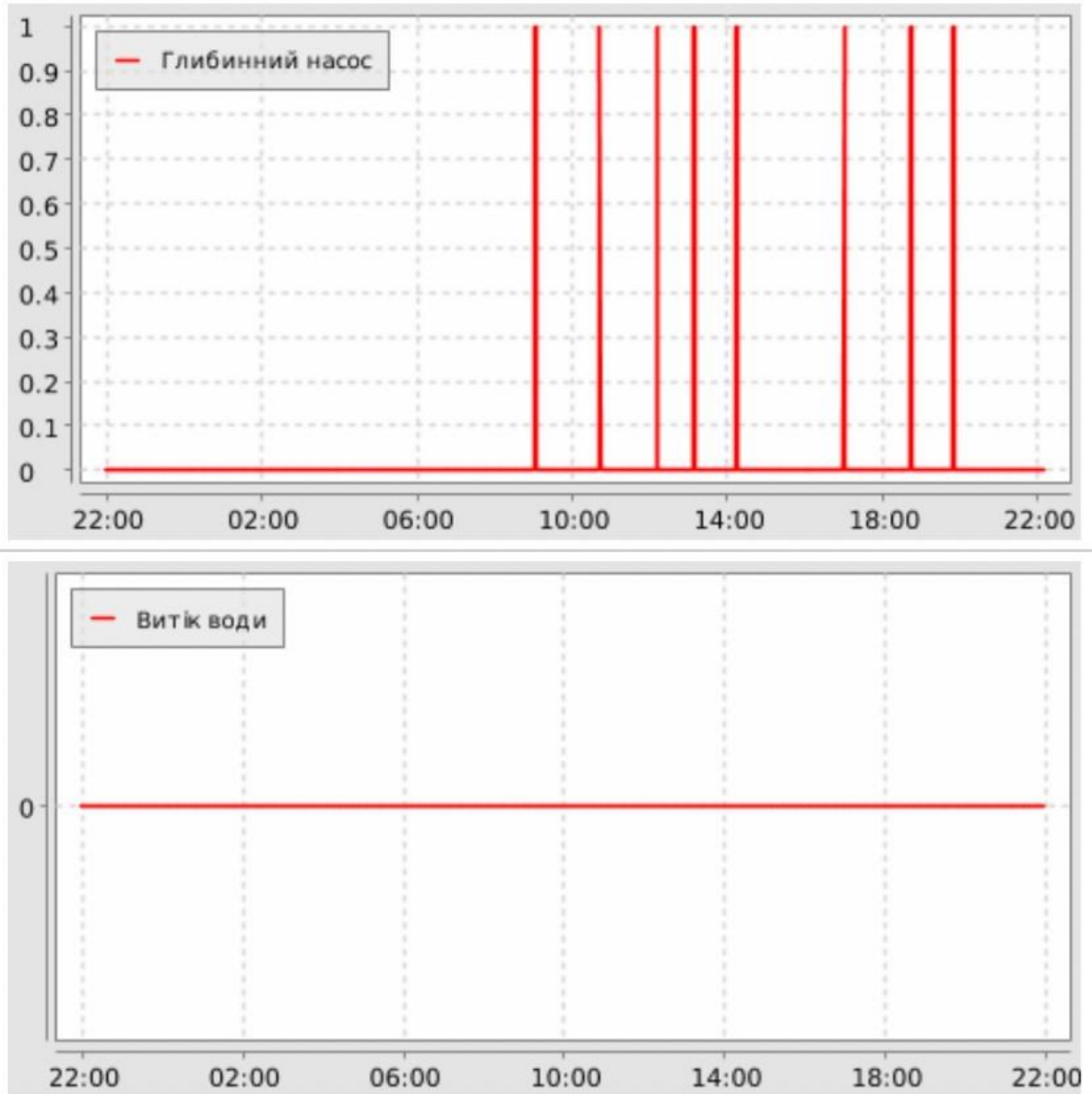


Рисунок 4.20 – Графіки, що ілюструють показники процесу та статуси обладнання водопостачання на основі даних OpenHAB

4.3 Висновки до розділу 4

У цьому розділі відповідно до мети та задач дипломної роботи виконано конструювання програмного забезпечення підсистеми моніторингу водопостачання та її інтеграція в СРБ на платформі OpenHAB.

Вхідними даними для програми при роботі системи являються дані отримані від датчика води, що підключений до плати Arduino.

Вихідними даними є:

- повідомлення на Telegram Messenger про події внаслідок виконання правил;
- відображення інформації щодо моніторингу в мобільному застосунку OpenHAB;
- відображення інформації щодо моніторингу в користувацькому інтерфейсі OpenHAB;
- керуючі сигнали до водяного насосу, що надсилаються за допомогою реле.

ВИСНОВКИ

В результаті виконання дипломної роботи виконано проєктування та конструювання програмного забезпечення підсистеми моніторингу водопостачання та її інтеграція в СРБ на платформі OpenHAB.

Виконано аналіз вимог до проєктованої підсистеми моніторингу водопостачання для СРБ, розроблено архітектуру системи, в якій визначено місце створюваної підсистеми та її взаємодію з зовнішніми та внутрішніми компонентами СРБ. Обґрунтовано вибір апаратних та програмних засобів розробки. Показано, що в якості серверу СРБ доцільно обрати мінікомп'ютер Raspberry Pi, а в якості контролера для роботи з датчиків рівня води – Arduino. Як програмні середовища розробки обрано Arduino IDE та OpenHAB.

Розроблене ПЗ складається з двох частин – набору програм для Arduino та конфігураційних файлів для OpenHAB, в які було додано правила роботи з підсистемою моніторингу водопостачання. Розроблена підсистема нотифікації передбачає декілька видів сповіщень: звукове, світлове, в інтерфейсну форму, в мобільний застосунок OpenHAB, в месенджер Telegram.

Практична цінність розробки полягає в тому, що впровадження підсистеми моніторингу водопостачання дозволить уникнути витоків води та надлишкового її споживання, за умови забезпечення комфортного користування водними ресурсами.

Проєкт надає можливості для майбутнього розширення системи, тобто додавання інших датчиків та пристроїв управління, а також зміну інтерфейсу та підключення додаткових сервісів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Kavitha, R. Smart Home systems using wireless sensor network – a comparative analysis / R. Kavitha, G. M. Nasira, N. Nachamai// International journal of computer engineering & technologies. – 2012. – Vol.3, № 3. – P.94-103.
2. Debnath, B. Smart switching system using bluetooth technology/, B. Debnath, R.Dey, S. Roy // Artificial Intelligence: Amity international conference, Dubai, United Arab Emirates, 4–6 February 2019: proceedings. – Los Alamitos: IEEE, 2019. – P. 760-763.
3. Internet of Things, a vision of digital twins and case studies / [A. Murthy, M. Irshad, S.M. Noman et al.] // IoT and Spacecraft Informatics. Aerospace Engineering. - Amsterdam: Elsevier, 2022. - P. 101–127.
4. Javeed, D. SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT/ D. Javeed, T. Gao, M.T.Khan// Electronics. – 2021. - Vol.10, № 8. – P. 1-16.
5. Zamora-Izquierdo, M.A. An integral and networked home automation solution for indoor ambient intelligence / M.A. Zamora-Izquierdo, J. Santa, A.F. Gomez-Skarmeta // IEEE Pervasive Computing. – 2010. - Vol. 9, № 4. – P. 66–77.
6. 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.
7. Bhide, V.H. i-learning IoT: an intelligent self learning system for home automation using IoT / V.H. Bhide, S.Wagh // Communications and signal: International conference, Melmaruvathur, India, 2–4 April 2015: proceedings. – 2015.- P. 1763-1767.
8. Cook, D. J. How Smart are our environments? An updated look at the state of the art / D. J. Cook, S. K. Das // Parvasive and mobile computing. – 2007. - Vol.3. - P. 53-73.

9. Cook, D. J. Ambient intelligence: technologies, applications, and opportunities / D. J. Cook, J. C. Augusto, V. R. Jakkula // *Parvasive and mobile computing*. – 2009. - Vol.5, № 4. - P 277-298.
10. Liu, W. Application of ZigBee wireless sensor network in Smart Home system / W. Liu, Y. Yan // *International Journal of Advancements in Computing Technology*. – 2011. - Vol. 3, № 5. - P 154 - 160.
11. Han, D.M. Design and implementation of Smart Home energy management systems based on ZigBee / D.M. Han, J.H. Lim// *IEEE Transactions on Consumer electronics*. -2010. - Vol. 56, № 3. – P. 1417-1425.
12. Design and implementation of a wireless sensor network for Smart Homes / [M. Xu, L. Ma, F. Xia et al.] // *Mobile cyber-physical systems: International workshop in conjunction with UIC2010, IEEE, Xi'an, China, 26 - 29 October 2010: proceedings*. – 2010. - P. 1-5.
13. Suh, C. Design and implementation of intelligent Home control systems based on active sensor networks/ C. Suh, Y.-B. Ko// *IEEE Transactions on Consumer electronics*.- 2008. - Vol. 54, № 3. – P. 1177-1184.
14. An improved routing algorithm of Zigbee wireless sensor network for Smart home system / [D. Zhong, W. Ji, Y. Liu et al.] // *Automation, robotics and applications: 5th International conference, Wellington, New Zealand, 6-8 December 2011: proceedings*. – 2011. – P.346-350.
15. Fleury, A. SVM-based multi-modal classification of activities of daily living in health Smart Homes: sensors, algorithms and first experimental results / A.Fleury, M.Vacher, N.Noury// *IEEE Transation on Information technology in biomedicine*.- 2010. - Vol. 14, № 2. -P. 274 – 283.
16. Design and implementation of a wireless sensor network architecture using smart mobile devices/ [B. Koo, K. Han, J. J. Park et al.] // *Telecommunication systems*. – 2013. – Vol. 52. – P. 2311–2320.
17. Wu, Y.-C. A mobile health monitoring system using RFID ring-type pulse sensor/ [Y.-C. Wu, P.-F. Chen, Z.-H. Hu et al.] // *Dependable, autonomic and*

secure computing: 8th IEEE International conference, Chengdu, China, 12-14 December 2009: proceedings. – 2009. - P. 317-322.

18. A ZigBee-based Home automation system / [K. Gill, S.-H. Yang, F. Yao et al.] // IEEE Transactions on Consumer electronics. – 2009. - Vol. 55, № 2. – P. 422-430.

19. Bian, J. The new intelligent Home control system based on the dynamic and intelligent gateway/ J. Bian, D. Fan, J. Zhang// Broadband network and multimedia technology: 4th IEEE International conference, Shenzhen, China, 28-30 October 2011: proceedings. – 2011. - P. 526-530.

20. Gaddam, A. Trial & experimentation of a Smart Home monitoring system for elderly / A. Gaddam, S. C. Mukhopadhyay, G. S. Gupta // Instrumentation and Measurement Technology: IEEE International conference, Hangzhou, China, 10-12 May 2011: proceedings. – 2011. – P.1-6.

21. Wireless sensor network based e-health system – implementation and experimental results / [H. Yan, H. Huo, Y. Xu et al.] // IEEE Transactions on Consumer electronics. - 2010. -Vol. 56, № 4. – P. 2288-2295.

22. Tae, Y. Smart digital door lock for the Home automation / Y. Tae, P. P. Sthapit, J.-Y. Pyun // TENCON 2009: IEEE Region 10 conference, Singapore, 23-26 January 2009: proceedings. – 2009. - P. 1-6.

23. Surie, D. Wireless sensor networking of everyday objects in a Smart Home environment / D. Surie, O. Laguionie, T. Pederson // Intelligent sensors, sensor networks and information: International conference, Sydney, NSW, Australia, 15-18 December 2008: proceedings. – 2008. - P. 189-194.

24. Krishna, S. Design and implementation of Home security system based on WSNs and GSM technology / S. Krishna, J. Ravindra// International journal of engineering science and technology. - 2012. - Vol.2, № 4. - P. 139-142.

25. Bao, K. User behavior prediction for energy management in Smart homes / K. Bao, F. Allerdin // Fuzzy systems and knowledge discovery: 8th International conference, Shanghai, China, 26-28 July 2011: proceedings. – 2011. - P.1335-1339.

26. Human activity classification for Smart home: a multiagent approach / [M. R. Alam, M. B. I. Reaz, M. A. M. Ali et al.] // Industrial electronics and applications: IEEE Symposium, Penang, Malaysia, 03-05 October 2010: proceedings. – 2010. - P. 511-514.

27. Research and thinking of friendly Smart Home energy system Based on Smart power / [Y. Zhao, W. Sheng, J. Sun et al.] // Electrical and Control Engineering: International Conference, Yichang, China, 16-18 September 2011: proceedings. – 2011. - P. 4649-4654.

28. Reinisch, C. Agent based control in the Smart Home / C. Reinisch, W. Kastner // 37th Annual conference of the IEEE Industrial electronics society, Melbourne, VIC, Australia, 07-10 November 2011: proceedings. – 2011. – P. 334-339.

29. The energy aware Smart Home/ [M. Jahn, M. Jentsch, C. R. Prause et al.] // Future information technology: 5th International conference, Busan, Korea (South), 21-23 May 2010: proceedings. – 2010. – P. 1-8.

30. Vainio, A.-M. Proactive fuzzy control and adaptation methods for Smart Homes / A.-M. Vainio, M. Valtonen, J. Vanhala // IEEE Intelligent Systems. - 2008. - Vol. 23, № 2. - P. 42-49.

31. Swapnadip, P. Smart Home water monitor and control system [Electronic resource] / P. Swapnadip. – Access mode: <https://www.iotchallengekeysight.com/2019/entries/smart-water/114-0510-144643-smart-home-water-monitor-and-control-system>.

32. Ehsan, M. A. Smart Home automation and security system / M. A. Ehsan, M. I. Mahfuj, G. M. I. Hossain // 10.13140/RG.2.2.19635.84000.- P.1-3.

33. Abdelfattah, I. Dripless: a smart household water monitoring system / I. Abdelfattah, M. F. Abdelkader, A. Ghuniem // Leveraging technologies for growth: 2nd Europe – Middle East – North African regional conference of the International telecommunications society (ITS), Aswan, Egypt, 18-21 February, 2019: proceedings. – 2019. – P.1-12.

34. Kumar, K. Monitoring and controlling tap water flow at homes using Android mobile application / K. Kumar, M. A.Mahmoud // International journal of communication sciences and disorders. -2017. - Vol. 1, № 2. – P. 30-38.
35. Rasin, Z. Application and evaluation of highpower Zigbee based wireless sensor network in water irrigation control monitoring system / Z. Rasin, H. Hamzah, M.S. Mohd Aras // Industrial electronics & applications: 2009 IEEE Symposium, Kuala Lumpur, Malaysia, 04-06 October 2009: proceedings. – 2009. - P. 548-551.
36. Zulhani, R. Water quality monitoring system using ZigBee based wireless sensor network/ R. Zulhani, A. M. Rizal // International journal of Engineering & technology. – 2012. – Vol. 9, № 10. – P. 24-28.
37. Ebere, E. V. Microcontroller based automatic water level control system / E. V. Ebere, O. F. Oladipo // International journal of innovative research in Computer and communication engineering. -2007. - Vol.1, № 6. - P. 1390-1396.
38. Deepiga, T. Smart water monitoring system using wireless sensor network at home/office / T.Deepiga, A.Sivasankari // International research journal of Engineering and technology. – 2015. -Vol. 2, № 4. – P. 1305-1314.
39. Richerand, F. A. Smart water discharge and monitoring system / F. A.Richerand. - US 8,834,723 B1; published on Sep. 16, 2014. – 6 p.
40. myD-Link Home Wi-Fi water sensor DCH-S160 [Electronic resource]. – Access mode: <https://eu.dlink.com/uk/en/products/dch-s160-mydlink-home-water-sensor>.
41. DRIBLET Smart water meter [Electronic resource]. – Access mode: <https://elemental.green/dribblet-smart-water-meter/>.
42. Water Hero P-100 leak detection & automatic water shut off system [Electronic resource]. – Access mode: <https://waterhero.com/p100leakdetectionautoshutoff/>.
43. Flo Smart water monitor and shutoff [Electronic resource]. – Access mode: <https://meetflo.com>.

44. Raspberry Pi as Internet of Things hardware: performances and constraints / [M. Maksimovic, V. Vujović, N. Davidović et al.] // Electrical, electronic and computing engineering: 1st International conference, Vrnjačka Banja, Serbia, June 2-5 2014: proceedings. –2014. – P.1-6.
45. Communication within Cloudlet using the Raspberry/ [N. Kortas, H. Azzabi, A. Ben Arbia, J.B. Tahar et al.] // Procedia Computer Science, 2015. – Vol.73. – P. 193-198.
46. An inexpensive Arduino-based LED stimulator system for vision research/ [P. Teikari, R. P. Najjar, H. Malkki, et al.] // Journal of neuroscience methods. – 2012. - Vol. 211, 2. - P. 227-236.
47. Coley, G. Beaglebone black system reference manual / G.Coley. – Dallas: Texas Instruments, 2013. -108 p.
48. Palazzetti, E. Getting Started with UDOO/ E. Palazzetti // Packt Publishing Ltd, 2015. - P. 206.
49. Raspberry Pi 4 [Electronic resource]. – Access mode: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>.
50. Navarrete G. What is Arduino and why is it used? [Electronic resource] / G. Navarrete. – Access mode: <http://surl.li/fkvne>.
51. Jobit, J. Types of Arduino boards – Quick comparison on specification and features [Electronic resource] / J. Jobit. – Access mode: <https://circuitdigest.com/article/different-types-of-arduino-boards>.
52. Jones, B. Which programming languages can you use with Arduino? [Electronic resource] / B. Jones. – Access mode: <https://www.makeuseof.com/tag/programming-languages-can-use-arduino/>.
53. Arduino Uno [Electronic resource]. – Access mode: <https://doc.arduino.ua/ru/hardware/Uno>.
54. Основні типи датчиків рівня [Електрон. ресурс]. - Режим доступу: <https://ukrspecavtomat.com.ua/osnovnyie-tipyi-datchikov-urovnya/>.

55. Як вибрати датчик рівня рідини [Електрон. ресурс]. - Режим доступу: <https://simpletech.com.ua/ua/info-ua/datchyky-rivnia/kak-vybrat-datchik-urovnya-zhidkosti-ua>.
56. Датчик рівня рідини [Електрон. ресурс]. - Режим доступу: <https://arduino.ua/ru/prod644-datchik-yrovnya-jidkosti>
57. Savitch, W. Java: An introduction to Problem Solving and Programming / W. Savitch. - Pearson Education, 2018.- 1056 p.
58. Ortega, J. M. An introduction to C++ and numerical methods/ Ortega, J. M., A. S. Grimshaw. - Oxford niversity Press, 1998. – 288 p.
59. Guzdial, M. J. Introduction to computing and programming in Python, a multimedia approach / M. J. Guzdial, B. Ericson. - Prentice Hall Press, 2009. – 432 p.
60. Coburn, J. Getting Started With Arduino: A Beginner's Guide. [Electronic resource] / J. Coburn. – Access mode: <https://www.makeuseof.com/tag/getting-started-with-arduino-a-beginners-guide/>.
61. PlatformIO [Electronic resource]. – Access mode: <https://platformio.org/>.
62. Ardino IDE. [Electronic resource]. – Access mode: <https://www.arduino.cc/en/main/software>.
63. Home-Assistant official web page. [Electronic resource]. – Access mode: <http://www.home-assistant.io/>.
64. OpenHAB Documentation. [Electronic resource]. – Access mode: <https://www.openhab.org/docs/tutorial/>.
65. Getting started with OpenHAB home automation on Raspberry Pi [Electronic resource]. – Access mode: <https://www.makeuseof.com/tag/getting-started-openhab-home-automation-raspberry-pi/>.
66. 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.

ДОДАТОК А
Текст програми

A.1 Тексти програм для Arduino

A.1.1 Програма для моделювання роботи прототипу в середовищі Tinkercad

```
int led = 2;
int moistureThres = 50;
const int MIN_SENSOR_VALUE = 0;
const int MAX_SENSOR_VALUE = 876;

void setup()

{
    Serial.begin(9600);
    pinMode(led, OUTPUT);
    pinMode(A0, INPUT);
    //pinMode(2, OUTPUT);
}

void loop()

{
    int moisture = analogRead(A0);
    int moisturePercent = map(moisture, MIN_SENSOR_VALUE,
MAX_SENSOR_VALUE, 0, 100);
    Serial.println(moisturePercent);
    if (moisturePercent > moistureThres)
    {
        digitalWrite(led, HIGH);
    }
    else
```

```

{
    digitalWrite(led, LOW);
}
delay(100);
}

```

А.1.2 Текст програми базового модуля підсистеми моніторингу

```

// оголошення пінів Ардуіно, до яких підключено виводи
датчика
#define sensorPower 7
#define sensorPin A0
int val = 0; // визначення змінної для зберігання
поточного значення рівня

void setup()

{
    pinMode(sensorPower, OUTPUT); // налаштування виводу
D7 для живлення датчика на вихід
    digitalWrite(sensorPower, LOW); // встановлення на
виводі D7 низького рівня, щоб на датчик спочатку не
подавалося живлення
    Serial.begin(9600); // налаштування послідовного
зв'язку з комп'ютером
}

void loop()

{

```

```

    int level = readSensor(); // періодичний виклик функції
    щоб отримати значення і вивести його
    Serial.print("Water level: ");
    Serial.println(level);
    delay(1000);
}
int readSensor()// функція для зчитування показів
поточного рівня води
{
    digitalWrite(sensorPower, HIGH); // активувати датчик
    delay(10); // затримка 10 мілісекунд
    int val = analogRead(sensorPin); // зчитати аналоговий
показник з датчика
    digitalWrite(sensorPower, LOW); // виключити датчик
    return val; // повернути поточне значення
}

```

A.1.3 Текст програми модуля автоматичного вмикання/вимикання водопостачання

```

int x; // привласнюємо ім'я для значень з аналогового
входу A0
int RelayH = 3; // Реле сигналізації підключено до
цифрового виходу 3
int aPin=A3;

void setup() // процедура setup
{
    pinMode(RelayH, OUTPUT);

```

```

    pinMode(A3, INPUT); // до аналогового входу A3
    підключимо датчик
    Serial.begin(9600); // підключаємо монітор порта
}

void loop()

{
    x = analogRead(aPin); // змінна "x" знаходиться в
    інтервалі від 0 до 1023
    if (x > 100) { digitalWrite(RelayH, LOW );
        Serial.println("Trouble"); }
    //x більше 100 - вмикаємо світлодіод
    if (x < 100) { digitalWrite(RelayH, HIGH); } // x
    менше 500 - вимикаємо світлодіод
    Serial.println(x); // виводимо значення датчика на
    монітор
    delay(1000); // затримка в одну секунду

```

A.1.4 Текст програми модуля звукового сповіщення підсистеми моніторингу

```

const int  SENSOR_ANALOG_OUTPUT  =  A0;  // контакт
підключення аналогового виходу датчика
const int  SOUND_Pin  =  11;  // контакт підключення
виведення реле
const int  SOUND_FREQ  [3]  =  {587,466,293};  //частоти
звукового сигналу
const int  WATER_LEVELS  [3]  =  {600,500,400};  // значення
рівнів води

```

```

int sensor_value = 0; // змінна для збереження значення
датчика

void setup()

{
    Serial.begin(9600); // ініціалізація послідовного
порту
    pinMode(SOUND_PIN, OUTPUT); //налаштування пінів
індикації світлодіодів у режимі виводу
}

void loop()

{
    sensor_value = analogRead(SENSOR_ANALOG_OUTPUT); //
отримання значення з аналогового виводу датчика
    Serial.print("Analog value = ");
    Serial.println(sensor_value); // виведення значення в
монітор послідовного порту Arduino
    // Виведення звуку різної частоти для різних рівнів
занурення
    if(sensor_value>WATER_LEVELS [0])
        tone(SOUND_PIN, SOUND_FREQ [0],2000);
    else if(sensor_value>WATER_LEVELS [1])
        tone(SOUND_PIN, SOUND_FREQ [1],2000);
    else if(sensor_value>WATER_LEVELS [2])
        tone(SOUND_PIN, SOUND_FREQ [2],2000);
    else
        noTone(SOUND_PIN);
}

```

```
    delay(1000); // пауза перед наступним отриманням  
    значення 1000 мс  
}
```

A.1.5 Текст програми модуля світлового сповіщення підсистеми моніторингу

```
#include <LiquidCrystal.h>

int State = 0;
int f = 0;

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

void setup()

{
    lcd.begin(16, 2);
    pinMode(9, INPUT);
    pinMode(10, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);
}

void loop()

{
    State = digitalRead(9);
    if(State==LOW) {
```

```
digitalWrite(12, LOW);
digitalWrite(13, LOW);
if(f==0){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Hello world!");
    f=1;
}
if ((millis()/1000)%2==0){
    digitalWrite(11, HIGH);
    digitalWrite(10, LOW);
}
else{
    digitalWrite(10, HIGH);
    digitalWrite(11, LOW);
}
}
if(State==HIGH){
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
    if(f==1){
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Goodbye world!");
        f=0;
    }
    if ((millis()/1000)%2==0){
        digitalWrite(13, HIGH);
        digitalWrite(12, LOW);
    }
}
```

```

    else{
        digitalWrite(12, HIGH);
        digitalWrite(13, LOW);
    }
}
delay (20);
}

```

A.2 Тексти програм для OpenHAB

A.2.1 Файл demo.items

```

Group Room          (All)

String Sensors "$s" { serial="COM3@9600" }
/* Room */
Switch Air_Conditioning_FF_SW      "Air conditioning"
    <heating>                      (Room)
Number Temperature_FF              "Temperature [%d
°C]"    <temperature>            (Room)
Number Humidity_FF                 "Humidity [%d %%]"
    <water>                        (Room)
Number Water_Sensor_FF             "Water      sensor
[%d]"                              (Room)
Switch Light_Sensor_FF             "Light      sensor
[%s]"                              (Room)

```

A 2.2 Файл demo.rules

```

import org.openhab.core.library.types.*
import org.openhab.core.persistence.*

```

```

import org.openhab.model.script.actions.*
//import java.lang.Math

var Number hum = null
var Number tem = null
var Number light = 0
var Number water = null
rule "Records last weather update time"
when
    Item Weather_Temperature received update
then
    postUpdate(Weather_LastUpdate, new DateTimeType())
end

rule "Send Telegram message"
when
    Item Air_Conditioning_FF_SW changed
then
    if(Air_Conditioning_FF_SW.state == ON)
    {
        sendCommand(Sensors, "O")
    }
    else if(Air_Conditioning_FF_SW.state == OFF)
    {
        sendCommand(Sensors, "F")
    }
    sendTelegram("ZNTU_OpenHab_bot", "Air conditioning
is %s", Air_Conditioning_FF_SW.state.toString)
    sendSms("+380508107549", "Text message")
end

```

```

rule "Water sensor"
when
    Item Water_Sensor_FF received update
then
    if(Water_Sensor_FF.state > 400)
    {
        sendTelegram("ZNTU_OpenHab_bot", "ALERT! Water
in the room")
    }
end

```

```

rule "Send OpenHab notification"
when
    Item Temperature_FF changed
then
    if(Temperature_FF.state <= 24) {

        sendNotification("dmitry.kravchenko95@gmail.com",
"test rule!!")
    }
end

```

```

rule "Turn on air conditioning if temp in room < 15
degrees"
when
    Item Weather_Temperature received update or
    Item Temperature_FF received update
then

```

```

        if((Temperature_FF.state < 15) &&
(Weather_Temperature.state < 10))
sendCommand(Air_Conditioning_FF_SW, ON)
end

```

```

rule "Turn off air conditioning if outside temp less than
1 C"

```

```

when

```

```

    Item Air_Conditioning_FF_SW changed from OFF to ON
then

```

```

    if( Weather_Temperature.state < 1) {
        postUpdate(Air_Conditioning_FF_SW, OFF)
    }

```

```

end

```

```

rule "Serial binding on COM port"

```

```

when

```

```

    Item Sensors received update
then

```

```

    var          String[]          buffer          =
Sensors.state.toString.split("\\\\|")
    if (buffer.get(0) == "A")
    {
        hum = new Float(buffer.get(2))
        tem = new Float(buffer.get(1))
        water = new Float(buffer.get(3))
        light = new Float(buffer.get(4))
        postUpdate(Temperature_FF,tem)
        postUpdate(Humidity_FF, hum)
        postUpdate(Water_Sensor_FF, water)
    }

```

```

        postUpdate(Light_Sensor_FF, light)
    }
end

```

```

rule "Timer Sensor"
when
    Time cron "*/5 * * * * ?" or
    System started
then
    sendCommand(Sensors, "U")
end

```

A 2.3 Файл demo.sitemap

```

sitemap demo label="Smart House"
{
    Frame label="The 4th Floor" {
        Text label="Room" icon="firstfloor" {
            Switch item=Air_Conditioning_FF_SW
            Text item=Temperature_FF
            Text item=Humidity_FF
            Text item=Light_Sensor_FF icon="light-on"
        }
    }

    Frame label="Weather" {
        Item=Weather_Temperature value
        color=[Weather_LastUpdate=="Uninitialized"="gray",Weath
er_LastUpdate>90="gray",>25="green",>15="orange",>5="re
d",<=5="blue"] {

```

```

Frame {
    //Item=Weather_Temp_Max
    valuecolor=[>25="green",>15="orange",>5="red",<=5="blue"
    "]

    //Item=Weather_Temp_Min
    valuecolor=[>25="orange",>15="green",>5="orange",<=5="blue"]

    Text          item=Weather_Temp_Feel
    icon="temperature"
    valuecolor=[Weather_Temp_Feel>25="red",Weather_Temp_Feel>12="orange",Weather_Temp_Feel>6="orange",Weather_Temp_Feel<=5="aqua"]

    Text          item=Weather_Humidity
    icon="humidity"

    Text item=Weather_Pressure
    Text          item=Weather_Wind_Speed
    icon="wind"

    Text item=Weather_Rain icon="water"
    Text item=Weather_Snow icon="snow_1"
    Text          item=Weather_Temp_MinMax
    icon="temperature-max"

    Item=Weather_LastUpdate icon="clock-on"
    visibility=[Weather_LastUpdate>30]
    valuecolor=[Weather_LastUpdate>120="red",
    Weather_LastUpdate>300="orange"]
    }
}

Group item=Astronomical label="Astronomical
Data" icon="sun"
}

```

```
Frame label="Date" {  
    Text item=Date  
}  
}
```

ДОДАТОК Б
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дипломна робота бакалавра
Програмне забезпечення для
моніторингу водопостачання в
системі «Розумний будинок»

Виконала
ст. гр. КНТ-129

Ксенія НЕЧАЙ

Керівник
к.т.н., доцент

Анжеліка ПАРХОМЕНКО

Рисунок Б.1 – Титульний слайд

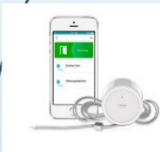
Актуальність роботи

2

Порівняльна характеристика існуючих платформ домашньої автоматизації

Платформа домашньої автоматизації	API	Мова програмування	Кількість протоколів	MQTT	Вебінтерфейс
OpenHAB	REST	Java	+	+	+
Domoticz	JSON	C++	-	+	+
Jeedom	HTTP, JSON	PHP	+	-	+
Home Assistant	Python, REST	Python	+	+	+

Огляд існуючих пристроїв моніторингу водопостачання



D-Link

Джерело: <https://meetflo.com/>;
<https://eu.dlink.com/uk/en/products/dch-s160-mydlink-home-water-sensor>;



Driblet



Water Hero

<https://elemental.green/driblet-smart-water-meter/>;



Flo

Рисунок Б.2 – Слайд 2

Актуальність роботи

3

Порівняльна характеристик існуючих систем моніторингу водопостачання

Система	Dripless	D-Link	Driblet	Water Hero	Flo
Простота встановлення	Помірна – встановити прямо в трубу, де вода входить до дому	Легко – датчик на підлозі	Складно – підключити до всіх водопровідних труб	Помірна – встановити прямо в трубу, де вода входить до дому	Помірна – встановити прямо в трубу, де вода входить до дому
Водоспоживання та статистика	+	-	+	+	+
Цілі споживання та поради	+	-	+	-	+
Мобільний застосунок	+	-	+	-	+
Вебінтерфейс	+	-	-	+	-
Підключення	Wi-Fi & GSM	Wi-Fi	Wi-Fi	Ethernet	Wi-Fi
Виявлення витоку	+	+	-	+	+
Метод виявлення	ШИ	Визначення води на підлозі	-	Виявлення перевищення витрати води	Виявлення перевищення витрати води
Дія на витік	Відключення води та сирена	Сирена	-	Відключення води	Відключення води
Моніторинг насосу	+	-	-	-	-
Джерело живлення	Самокерований	Розетка AC	Самокерований	Розетка змінного струму	Розетка змінного струму

Рисунок Б.3 – Слайд 3

Мета та завдання дипломної роботи

4

Мета роботи – дослідження та розробка програмно-апаратного забезпечення підсистеми моніторингу водопостачання для запобігання витоку води та надлишковому її споживанню в системі Розумний будинок.

Завдання роботи:

- ❖ дослідження технологій розробки сучасних IoT систем для моніторингу водопостачання;
- ❖ аналіз вимог та розробка структури підсистеми моніторингу водопостачання СРБ;
- ❖ вибір та обґрунтування апаратних та програмних засобів розробки;
- ❖ конструювання програмного забезпечення підсистеми моніторингу водопостачання;
- ❖ інтеграція підсистеми моніторингу водопостачання в СРБ.

Рисунок Б.4 – Слайд 4

Аналіз вимог та розробка архітектури системи

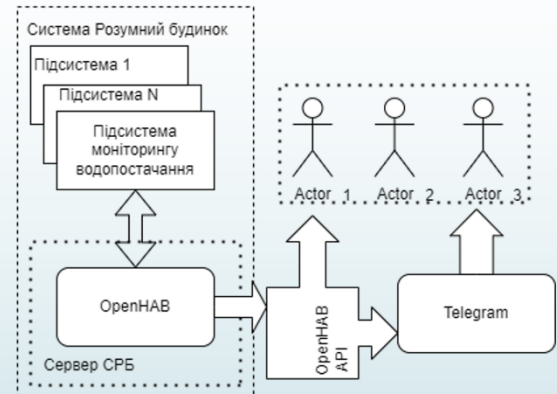
5

Функціональні вимоги:

- контроль рівня води;
- автоматичне вмикання/вимикання подачі води;
- звукова та світлова індикація інформації про рівень води.

Нефункціональні вимоги:

- доступна елементна база;
- інтеграція в СРБ на платформі OpenHAB;
- нотифікація через Telegram про надзвичайні ситуації.



Архітектура СРБ з підсистемою моніторингу водопостачання

Рисунок Б.5 – Слайд 5

Аналіз та вибір апаратних засобів розробки

6

	RaspberryPI	Arduino	BeagleBoneBlack	Udoo	Порівняння платформ розробки						
CPU	ARM, BCM2835	ATMEGA8, ATMEGA168, ATMEGA328, ATMEGA1280	ARM, Cortex-A8	MX6Quad, 4xARM Cortex, A9 core, Atmel SAM3X8E, ARM CortexM3 CPU							
RAM	256-512 МБ	16-32 КБ	512МБ	1ГБ							
LAN(Мбіт)	10/100	-	10/100	10/100/1000							
Аналогові входи	0	6									
ОС	Raspbian;OSMC OpenELEC;Windows IoT Core; RISC OS; 5V/USB	Linux, WindowsMAC	Плата/ характеристика	Arduino UNO R3	Arduino Nano	Arduino PRO Mini	Arduino Leonardo	Arduino Micro	Arduino NanoEvery	Arduino Mega2560 Rev 3	
			Мікроконтролер	ATmega328p	ATmega328p	ATmega328p	ATmega32u4	ATmega32u4	ATmega4809	ATmega2560	
Живлення			Тип USB	USB-B	MiniB USB	MiniB USB	Micro USB (USB B)	MicroUSB	MicroUSB	USB-B	
Засоби програмування	C, C++, Java, Python	C, C++, Java, Python	Порти вводу/виводу	Цифрові In/Out: 14 Аналогові In: 6	14	14	14	20	20	14	54
			Протоколи зв'язку	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	
			Тактова частота	16МГц	16МГц	16МГц	16МГц	16МГц	20МГц	16МГц	
			Flash пам'ять	32КБ	32КБ	32КБ	32КБ	32КБ	32КБ	32КБ	
			Живлення	Напруга: 7-12 В Струм: 20mA	7-12 В 20mA	7-12 В 20mA	7-12 В 20mA	7-12 В 10mA	7-12 В 10mA	7-12 В 20mA	7-12 В 20mA
Порівняння плат Arduino											

Порівняння плат Arduino

Рисунок Б.6 – Слайд 6

Аналіз та вибір програмних засобів розробки

7

Критерій	Python	Java	C++
Ш видкодів	+	++	++
Час розробки	+	++	+++
Довжина коду	+	++	+++
Методи відлагодження	+	+	+

Порівняння мов програмування для Arduino

Порівняння середовищ програмування для Arduino

Характеристика	Arduino IDE	PlatformIO
Простота	+	+
Підтримка	+	+
Платформа	Java	Python
Операційна система	Windows, OS X, Linux	Windows, macOS, Linux, FreeBSD
Менеджер бібліотек	-	+
Підтримка плагінів	-	+
Наявність вебверсії	+	-
Доступність	Open-source	Open-source

Рисунок Б.7 – Слайд 7

Структура програмного забезпечення

8



Рисунок Б.8 – Слайд 8

Схеми підключення компонентів підсистеми

9

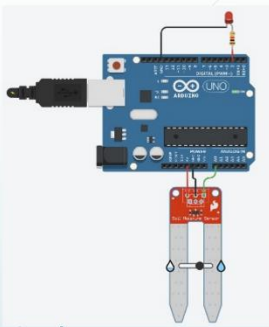


Схема для дослідження роботи датчика вологості в Tinkercad

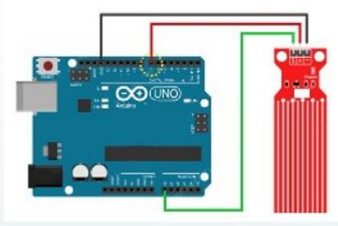


Схема для дослідження роботи базового модуля

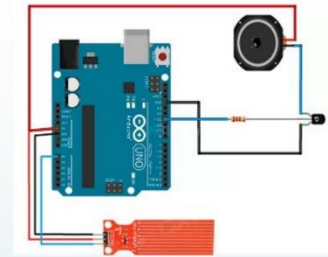


Схема для дослідження роботи модуля звукового сповіщення

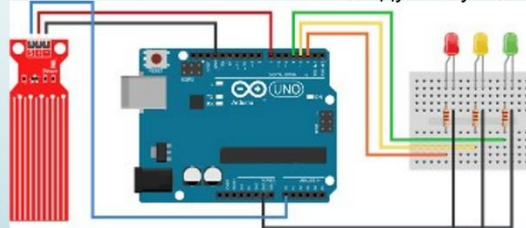


Схема для дослідження роботи модуля світлового сповіщення

Рисунок Б.9 – Слайд 9

Схема алгоритму функціонування підсистеми та результати моніторингу

10

Water level: 0	Датчик сухий
Water level: 0	
Water level: 0	
Water level: 0	
Water level: 80	Датчик занурений частково
Water level: 130	
Water level: 260	
Water level: 390	
Water level: 411	Датчик занурений повністю
Water level: 420	
Water level: 436	
Water level: 448	
Water level: 485	
Water level: 511	
Water level: 521	
Water level: 524	
Water level: 533	

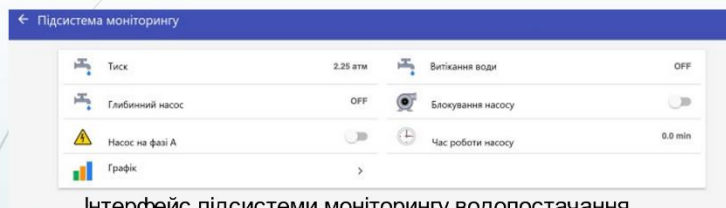
Виведення результатів роботи програми у монітор послідовного порту



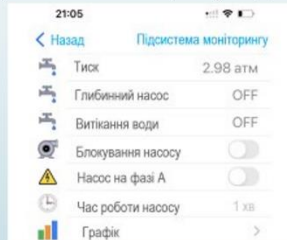
Рисунок Б.10 – Слайд 10

11

Схема алгоритму та результати розробки конфігураційних файлів для OpenHAB



Інтерфейс підсистеми моніторингу водопостачання



Відображення інформації в мобільному застосунку tuopenhab



Рисунок Б.11 – Слайд 11

12

Результати тестування розробленого ПЗ в Telegram та OpenHAB

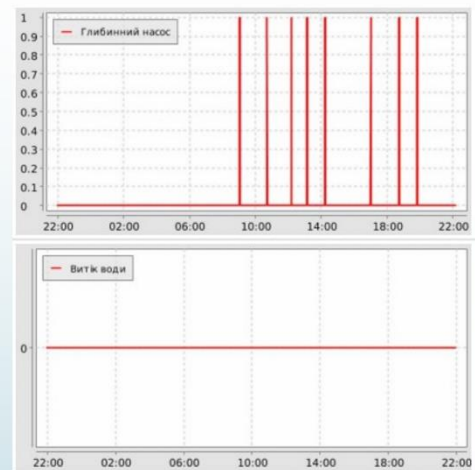
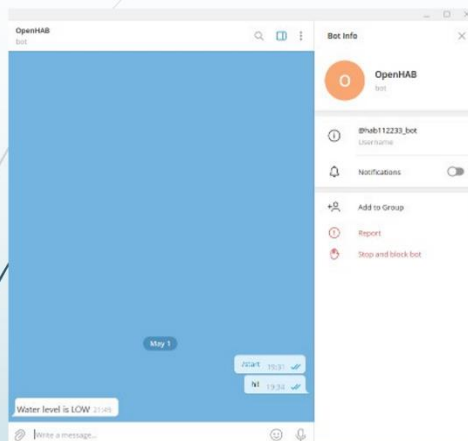


Рисунок Б.12 – Слайд 12

Висновки

13

В результаті виконання дипломної роботи виконано проектування та конструювання програмного забезпечення підсистеми моніторингу водопостачання та її інтеграцію в систему Розумний будинок на платформі OpenHAB.

Практична цінність розробки полягає в тому, що впровадження підсистеми моніторингу водопостачання дозволить уникнути витоків води та надлишкового її споживання, за умови забезпечення комфортного користування водними ресурсами.

Проект надає можливості для майбутнього розширення системи, тобто додавання інших датчиків та пристроїв управління, а також зміну інтерфейсу та підключення додаткових сервісів.

Рисунок Б.13 – Слайд 13