

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
Кафедра автоматичного управління в технічних системах**

«До захисту допущено»

Завідувач кафедри

_____ О.І. Ролік

«__» _____ 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки 6.050103 «Програмна інженерія»

**на тему: «Програмно-апаратний комплекс управління системою рухомих
об'єктів»**

Виконав (-ла):

студент (-ка) IV курсу, групи ІТ-51

Сопов Олексій Олександрович _____

Керівник:

Професор кафедри АУТС, д.т.н., доцент Корнієнко Б.Я. _____

Рецензент:

Доцент кафедри ОТ, к.т.н., доцент Павлов В.Г. _____

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент (-ка) _____

Київ – 2019 рік

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки – 6.050103 «Програмна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.І. Ролік

« ___ » _____ 2019 р.

ЗАВДАННЯ

на дипломний проект студенту

Сопову Олексію Олександровичу

1. Тема проекту «Програмно-апаратний комплекс управління системою рухомих об'єктів», керівник проекту Професор кафедри АУТС, д.т.н., доцент Корнієнко Б.Я., затверджені наказом по університету від « ___ » _____ 2019 р. № _____

2. Термін подання студентом проекту _____

3. Вихідні дані до проекту

а) технології (у тому числі мови програмування) – Android, Arduino, Java, C, Bluetooth, б) середовища розробки: Arduino Studio, Android Studio, в) мікроконтролер Arduino

4. Зміст пояснювальної записки

1. Вступ 2. Опис об'єкта проектування та розробка вимог до характеристик 3. Проектування комплексу та вибір засобів реалізації. 4. Реалізація компонентів 5. Можливості подальшого розвитку та розширення 6. Висновки.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

1. Діаграма компонентів 2. Схема алгоритму перевірки статусу 3. Діаграма послідовності 4. Діаграма станів.

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Отримання завдання	10.04.2019	
2	Планування дипломного проекту	15.04.2019	
3	Аналіз існуючих рішень	20.04.2019	
4	Розробка структури проекту	24.04.2019	
5	Розробка мобільного застосунку	01.05.2019	
6	Розробка програмної частини хабу	10.05.2019	
7	Аналіз результатів роботи	19.05.2019	
8	Оформлення проекту	30.05.2019	
9	Захист дипломного проекту	19.06.2019	

Студент

Сопов О.О.

Керівник проекту

Корнієнко Б.Я.

АНОТАЦІЯ

Сопов О.О. Програмно-апаратний комплекс управління системою рухомих об'єктів. КПІ ім. Ігоря Сікорського, Київ, 2019.

Ключові слова: програмно-апаратний комплекс, хаб, клієнтський застосунок.

Основна частина документу викладена у пояснювальній записці, виконаній на 64 сторінках, містить 19 рисунків, 1 таблицю. та 1 додаток.

Дипломний проект присвячений проектуванню та розробці програмно-апаратного комплексу управління системою рухомих об'єктів, який надає можливість користувачу управляти декількома рухомими об'єктами одночасно через єдиний інтерфейс.

Структура розробленого комплексу складається з двох компонентів: хаб та клієнтський застосунок. Хаб реалізовано на мікроконтролері Arduino з використанням мови програмування Arduino, клієнтський застосунок – на Android з використанням мови програмування Java.

SUMMARY

Sopov O.O. Software-hardware complex of management of the system of moving objects. KPI them Igor Sikorsky, Kyiv, 2019.

Key words: software-hardware system, hub, client application.

The bulk of the document is presented in the explanatory note, executed on 64 pages, contains 19 figures, 1 table. and 1 supplement.

The diploma project is devoted to the design and development of software and hardware management system for moving objects, which allows the user to control several moving objects simultaneously through a single interface.

The structure of the developed complex consists of two components: hub and client application. The hub is implemented on the Arduino microcontroller using the Arduino programming language, the client application - on Android using the Java programming language.

Номер рядка	Формат	Позначення	Найменування	Кіл. листів	№ екз.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IT51.240БАК.002 ПЗ	Пояснювальна записка			
6						
7	A3	IT51.240БАК.003 Д1	Діаграма компонентів	1		
8						
9	A3	IT51.240БАК.004 Д2	Схема алгоритму перевірки	1		
10			статусу			
11						
12	A3	IT51.240БАК.005 Д3	Діаграма послідовності	1		
13						
14	A3	IT51.240БАК.006 Д4	Діаграма станів	1		
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						

					IT51.240БАК.002 ТП			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Сопов О.О.			Програмно-апаратний комплекс управління системою рухомих об'єктів Відомість технічного проекту	Лім.	Арк.	Акрушіє
Перевір.		Корнієнко Б.Я.					1	1
Реценз.		Павлов В.Г.				КПІ ім. Ігоря Сікорського ФІОТ гр. IT-51		
Н. Контр.		Шинкевич Н.К.						
Затверд.								

Пояснювальна записка
до дипломного проекту
на тему: «Програмно-апаратний комплекс
управління системою рухомих об'єктів»

Київ – 2019 рік

ЗМІСТ

ВСТУП.....	5
1 ОПИС ОБ’ЄКТА ПРОЕКТУВАННЯ ТА РОЗРОБКА ВИМОГ ДО ХАРАКТЕРИСТИК.....	7
1.1 Опис об’єкта проектування.....	7
1.2 Функціональні вимоги.....	8
1.3 Нефункціональні вимоги.....	10
1.4 Аналіз аналогів розроблюваного комплексу.....	12
1.4.1 Класичний варіант управління рухомими об’єктами.....	12
1.4.2 Централізовані системи.....	14
1.5 Висновки до розділу.....	16
2 ПРОЕКТУВАННЯ КОМПЛЕКСУ ТА ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ....	17
2.1 Проектування структури комплексу.....	17
2.1.1 Структура з використанням декількох пультів управління.....	17
2.1.2 Структура з використанням кільцевої топології.....	19
2.1.3 Структура з використанням хабу та клієнтського додатку.....	21
2.1.4 Вибір структури серед проаналізованих.....	23
2.2 Проектування хабу.....	25
2.2.1 Вибір методу комунікації.....	25
2.2.2 Протокол зв’язку.....	29
2.2.3 Вибір компонентів.....	30
2.3 Проектування клієнтського додатку.....	35
2.3.1 Опис принципу роботи.....	35
2.3.2 Цільова платформа.....	36
2.3.3 Вибір мови програмування.....	38
2.3.4 Проектування користувацького інтерфейсу.....	39
2.4 Розробка то опис діаграми компонентів.....	41

					<i>IT51.240БАК.002 ПЗ</i>		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Сопов О.О.			Лім.	Арк.	Акрушів
Перевір.		Корнієнко Б.Я.				2	63
Реценз.		Павлов В.Г.			КПІ ім. Ігоря Сікорського ФІОТ гр. ІТ-51		
Н. Контр.		Шинкевич Н.К.					
Затверд.							
					Програмно-апаратний комплекс управління системою рухомих об’єктів Пояснювальна записка		

2.5	Опис алгоритмів	43
2.5.1	Алгоритм перевірки статусу компонентів системи.....	43
2.5.2	Алгоритм передачі команд до рухомих об'єктів	44
2.6	Висновки до розділу	46
3	РЕАЛІЗАЦІЯ КОМПОНЕНТІВ.....	47
3.1	Реалізація клієнтського застосунку.....	47
3.2	Реалізація хабу.....	54
3.3	Висновки до розділу	56
4	МОЖЛИВОСТІ ПОДАЛЬШОГО РОЗВИТКУ ТА РОЗШИРЕННЯ	57
4.1	Збільшення ширини покриття хабу.....	57
4.2	Реалізація протоколу зв'язку для вирішення прикладних задач.....	59
4.3	Автоматизація прийнять рішень хабом	59
4.4	Висновки до розділу	61
	ВИСНОВКИ.....	62
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63
	ДОДАТОК А.....	2

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ

Програмно-апаратний комплекс – набір технічних і програмних засобів, що працюють спільно для виконання однієї задачі.

Користувач – будь-яка людина, яка користується програмним застосунком.

Хаб програмно-апаратного комплексу – вузол, у якому сходяться інформаційні потоки із різних джерел.

Керований рухомий об'єкт – будь-який об'єкт, який може змінювати своє положення в просторі під впливом надісланих до нього сигналів користувачем

UML – Unified Modeling Language. Стандартна мова для визначення, візуалізації, побудови та документування артефактів програмних та паратних систем.

Tx, Rx – Transmit, Receive. Передача та зчитування відповідно.

JSON – JavaScript Object Notation. Текстовий формат обміну даними, на основі об'єкту JavaScript.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Керовані рухомі об'єкти стрімко розвиваються в останній час та мають великий потенціал для подальшого розвитку та вдосконалення. До керованих рухомих об'єктів можна віднести безліч роботоподібних пристроїв з різними методами управління, від іграшкової машини до багатofункціонального дрона.

Такі об'єкти мають все більш широке застосування у багатьох областях діяльності людини. Наприклад, у агропромисловості використовуються для точного, ефективного і своєчасного внесення добрив. Також, використання керованих рухомих об'єктів може стати у нагоді для нагляду за безпекою на великих територіях з важливими інфраструктурними об'єктами.

Не менш важливий внесок керовані рухомі об'єкти надають метеорології, кліматології для екологічного моніторингу, при картографуванні місцевості, фотовідеозйомці тощо.

Дуже перспективним напрямком для керованих рухомих об'єктів є їх використання у місцях з обмеженою для оператора видимістю, у місцях техногенних катастроф або місцях інших надзвичайних ситуацій для швидкого та точного обстеження необхідної зони.

Хоча керовані рухомі об'єкти і є доволі потужними та мають безліч функціональності для комфортної роботи з ними, керування ними є досить складним процесом та вимагає від оператора значних навичок та умінь. Більше того, більшість сучасних методів управління рухомими об'єктами дає змогу керувати лише одним об'єктом в один проміжок часу.

Для досягнення більш високої продуктивності використання керованих рухомих об'єктів у промисловості, різних областях дослідницької та наукової діяльності тощо, необхідно керувати не одним рухомим об'єктом, а цілою системою подібних об'єктів, які сумісно мають виконувати одну поставлену ціль.

					IT51.240БАК.002 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

Таке їх використання підвищить функціональні можливості, швидкість виконання поставлених задач у декілька раз, дозволить виконувати необхідні задачі паралельно, використовуючи декілька рухомих об'єктів. На жаль, сьогодні такий підхід не використовується у широких масах.

Метою даної роботи є проектування та реалізація програмно-апаратного комплексу управління системою рухомих об'єктів. Такий комплекс з вирішенням проблеми керування декількома об'єктами одночасно, та проблеми складності керування ними.

Для успішного та легкого керування системою рухомих об'єктів необхідно вирішити ряд питань, серед яких: методи та структура керування системою об'єктів, можливість легкого керування та моніторингу для оператора, можливість зробити такий метод керування універсальним.

У даній роботі описано концепцію розроблюваного програмно-апаратного комплексу управління системою рухомих об'єктів, поставлено функціональні та нефункціональні вимоги, проаналізовано вже існуючі подібні рішення. На основі цього, обрано найбільш актуальний та правильний метод реалізації. У даній роботі спроектовано усі компоненти системи, обрано відповідний метод комунікації і показані основні етапи реалізації такого комплексу.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						6
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

1 ОПИС ОБ'ЄКТА ПРОЕКТУВАННЯ ТА РОЗРОБКА ВИМОГ ДО ХАРАКТЕРИСТИК

1.1 Опис об'єкта проектування

На етапі опису об'єкта проектування опишемо проблему, яку необхідно вирішити та висунемо ідею вирішення цієї проблеми. Опис об'єкта є фундаментом для реалізації усіх наступних етапів.

Керовані рухомі об'єкти є в наш час доволі популярними, проте системи з їх використанням мають багато недоліків, які заважають використовувати їх повною мірою:

- основним методом керування рухомим об'єктом є пряме керування одним об'єктом в один проміжок часу;
- керування такими об'єктами є доволі складним і вимагає від оператора багатьох навичок та умінь;
- керування рухомими об'єктами не є універсальним; тобто для кожного типу об'єкту є власні реалізації керування.

Такі недоліки не дають можливості більш широко використовувати керовані рухомі об'єкти у промисловості для підвищення продуктивності праці, автоматизації складних процесів. Більше того, загальновідомі методи управління не включають в себе можливість паралельного використання декількох рухомих об'єктів, що змушує розвивати їх вглиб, тобто покращуючи функціональні складові, що є доволі складним шляхом, а не вшир, збільшуючи кількість об'єктів.

Для вирішення цієї проблеми було висунуто програмно-апаратний комплекс управління системою рухомих об'єктів.

Під терміном «система рухомих об'єктів» будемо вважати сукупність рухомих об'єктів, які використовуються для виконання поставленої задачі.

Такий програмний комплекс складається із наступних механізмів:

					IT51.240БАК.002 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

- низькорівневий механізм управління рухомими об'єктами. За допомогою даного механізму буде проходити управління рухомими об'єктами на апаратному рівні;
- високорівневий механізм управління рухомими об'єктами оператором. Даний механізм – деяка абстракція над керуванням рухомими об'єктами на низькому рівні;
- механізм зв'язку компонент системи.

Така сукупність механізмів надає можливість управляти системою рухомих об'єктів. Більш детальний опис таких механізмів, вибір фактичної їх реалізації, обґрунтування вибору тієї чи іншої реалізації буде наведено у наступних розділах.

1.2 Функціональні вимоги

Функціональні вимоги при розробці складних систем визначають функції системи або її компонента, де функція описується як специфікація поведінки між виходами і входами.

Функціональні вимоги можуть включати обчислення, обробку даних та маніпулювання ними, технічні деталі, а також інші специфічні функціональні можливості, які визначають, як система повинна функціонувати. Функціональні вимоги підтримуються нефункціональними вимогами. Як правило, функціональні вимоги виражаються в формі «система повинна виконувати вимогу». План реалізації функціональних вимог докладно описаний в проекті системи [1].

Як визначено, функціональні вимоги визначають конкретні результати системи. Функціональні вимоги визначають архітектуру компонентів системи.

Опишемо функціональні вимоги до програмно-апаратного комплексу управління системою рухомих об'єктів.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

Функціональні вимоги щодо апаратної частини програмно-апаратного комплексу управління системою рухомих об'єктів:

- має надаватися можливість підключитися до нього рухомих об'єктам;
- має надаватися можливість надсилати команди до будь-якого підключеного до нього об'єкта;
- керування підключеними до комплексу об'єктами універсальним чином, тобто керування усіма об'єктами виконується одним (або більш широким) методом;
- має проводитися безперервна комунікація із об'єктами з мінімальною частотою 1 запит на секунду;
- апаратна частина має надавати вхідний інтерфейс для перегляду інформації про підключені об'єкти та бути мостом для доступу до таких об'єктів;
- апаратна частина має вести деталізований запис усіх важливих подій, що відбуваються із комплексом.

Функціональні вимоги щодо програмної частини програмно-апаратного комплексу управління системою рухомих об'єктів:

- програмна частина надає повну інформацію про комплекс, включаючи інформацію про усі підключені об'єкти;
- програмна частина має надавати можливість надіслати сигнали до апаратної частини для надання сигналу рухомих об'єктам;
- має надаватися можливість надіслати сигнал до конкретного об'єкта, або до цілої системи;
- має надаватися можливість керування станом комплексу: включення, виключення або переведення у інший стан.

Виходячи із даних функціональних вимог, далі проводиться проектування та безпосередня реалізація програмно-апаратного комплексу управління системою рухомих об'єктів.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3 Нефункціональні вимоги

Нефункціональні вимоги при розробці складних систем визначають критерії, які можуть використовуватися для оцінки роботи системи, а не для конкретних дій. Вони контрастують з функціональними вимогами, які визначають конкретну поведінку або функції.

Нефункціональні вимоги мають вигляд «система повинна бути», загальна властивість системи в цілому або конкретного аспекту, а не конкретної функції. Загальні властивості системи зазвичай відзначають різницю між успішним чи невдалим проектом розробки [2].

Нефункціональні вимоги часто називають «атрибутами якості» системи. Неофіційно їх іноді називають «здібностями» з таких атрибутів, як стабільність і переносимість. Серед нефункціональних вимог можна виділити:

- вимоги до надійності та безпеки;
- вимоги до експлуатації;
- вимоги до атрибутів якості;
- вимоги до вживаного обладнання та комплектуючі, ПЗ;
- вимоги до дизайну і юзабіліті;
- вимоги до масштабованості.

Опишемо нефункціональні вимоги до програмно-апаратного комплексу управління системою рухомих об'єктів:

Нефункціональні вимоги до апаратної частини:

- мінімальна кількість рухомих об'єктів, яка підтримується – 8. Більша кількість об'єктів є допустимою;
- уся інформація надається та надсилається у форматі, який легко читати людині та легко генерувати програмними засобами;
- уся інформація між компонентами передається бездротовими каналами;
- апаратна частина має бути надійною та доступною – максимальний час відгуку 1 секунда

									Арк.
									10
Змн.	Арк.	№ докум.	Підпис	Дата	IT51.240БАК.002 ПЗ				

- апаратна частина має складатися як мінімум із таких компонент: флеш-пам'ять 3 КВ, оперативна пам'ять 2 КВ, частота тактового генератора 16 МГц;
- надання команд до рухомих об'єктів має виконуватися із максимальною затримкою в 1 секунду;
- відстань зв'язку має бути не менш, ніж 100 метрів;
- апаратна частина має бути портативною;
- апаратна частина має бути масштабованою вглиб та вшир;
- апаратна частина має враховувати те, що об'єкти рухомі та періодично перевіряти їх статус.

Опишемо нефункціональні вимоги до програмної частини програмно-апаратного комплексу управління системою рухомих об'єктів:

- будь-яке управління, зчитування інформації тощо має проходити через користувацький інтерфейс;
- інтерфейс має бути зрозумілим та ергономічним;
- інтерфейс має відображати усю наявну інформацію про підключені об'єкти, апаратну частину та статус комплексу;
- продуктивність – програмна частина повинна виконувати всі передбачені операції без довгих очікувань і зависань. Виконання кожної операції має проходити в найкоротший термін;
- експлуатаційна придатність – програмна частина повинна виконувати всі необхідні вимоги і бути працездатним незалежно від часу його використання.

Надалі, нефункціональні вимоги будуть взяті до уваги під час проектування та реалізації комплексу, який розробляється.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

1.4 Аналіз аналогів розроблюваного комплексу

Розглянемо та проаналізуємо вже існуючі рішення, для того, щоб дізнатися які методи вирішення задачі управління декількома об'єктами одночасно вже є, чи є недоліки у вже існуючих рішень, чи можна їх вдосконалити. Також, варто зосередитися на існуючих архітектурних рішеннях та, можливо, покращити їх або зробити нове, яке буде більш пасувати для поставленої задачі.

При аналізі вже існуючих рішень особливу увагу звернемо на компоненти систем, методи їх комунікації. Проаналізуємо чи відповідають такі рішення вже поставленим вимогам, виділимо влучні рішення та проаналізуємо ті рішення, яких варто уникати.

Під час дослідження існуючих рішень та реалізацій було знайдено системи, які частково відповідають поставленим функціональним та нефункціональним вимогам управління системою рухомих об'єктів; але системи, яка б у повну міру виконувала необхідне, знайдено не було. Основною відмінністю була статичність усіх компонент системи або неможливість керування декількома пристроями одночасно.

1.4.1 Класичний варіант управління рухомими об'єктами

До класичного варіанту можна віднести найбільш простий та найбільш широковживаний варіант – керування рухомим об'єктом за допомогою пульта управління. Схематично такий метод можна побачити на рисунку 1.1.

Такий метод використовується дуже давно та є найбільш очевидним та дуже простим. Маємо у даному методі: користувач, пульт керування та, безпосередньо, об'єкт яким керують. Користувач подає сигнали до об'єкту управління за допомогою відповідного пульта, у якому інкапсульована логіка взаємодії та комунікації із рухомим об'єктом. Надалі, інформація передається каналом зв'язку та оброблюється рухомим об'єктом.

					IT51.240БАК.002 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

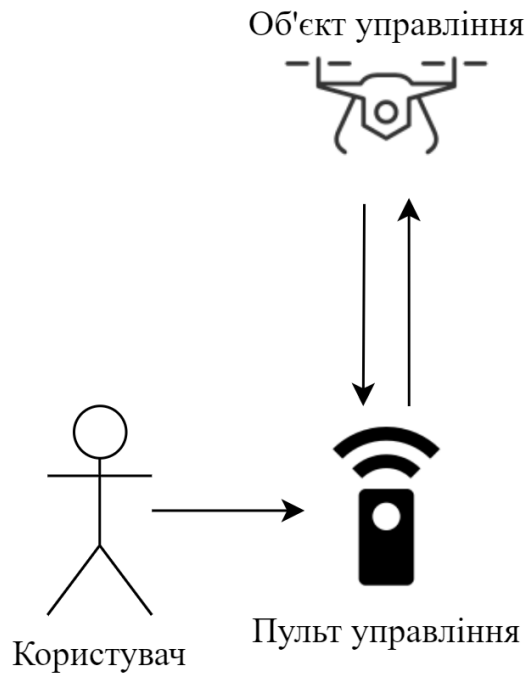


Рисунок 1.1 – Схема класичного методу управління рухомими об'єктами

До очевидних переваг відноситься:

- простота у реалізації;
- наявність зв'язку один до одного: один об'єкт управління – один пульт;
- можливість змінювати пульт управління безпосередньо для пристрою, яким він керує.

До недоліків такого підходу, в контексті розроблюваного комплексу, можна віднести:

- неможливість керувати декількома об'єктами одночасно;
- відсутність візуальної інформації у вигляді інтерфейсу користувача;
- зміни у рухомому об'єкті можуть привести до необхідності повністю модифікувати, або, навіть, змінити пульт керування.

Хоча даний класичний варіант не відповідає поставленим вимогам, адже надає можливість керувати лише одним об'єктом, він описує дуже надійний та широкодоступний метод управління.

Зазвичай, такі системи використовують радіозв'язок для управління рухомими об'єктами.

Наприклад, пульт управління DJI Mavic Pro [3], що є одним з лідерів на ринку, використовує радіозв'язок на частоті 2.4 GHz, що дає можливість надсилати сигнали на відстань до 7 кілометрів без перешкод.

Програмно-апаратний комплекс проектується для роботи із управління системою рухомих об'єктів, та серед вимог є вимога того, що зв'язок має бути бездротовим. Тому такий тип зв'язку максимально відповідає умовам та добре зарекомендував себе на ринку керування рухомими об'єктами, тому він буде розглянутий детально при виборі методів зв'язку.

1.4.2 Централізовані системи

Централізовані системи в останній час набули великої популярності, та відповідають багатьом функціональним та нефункціональним вимогам щодо розроблюваного комплексу. Принцип їх функціонування базується на комунікації хаба із датчиками. Схематично такий метод управління можна побачити на рисунку 1.2.

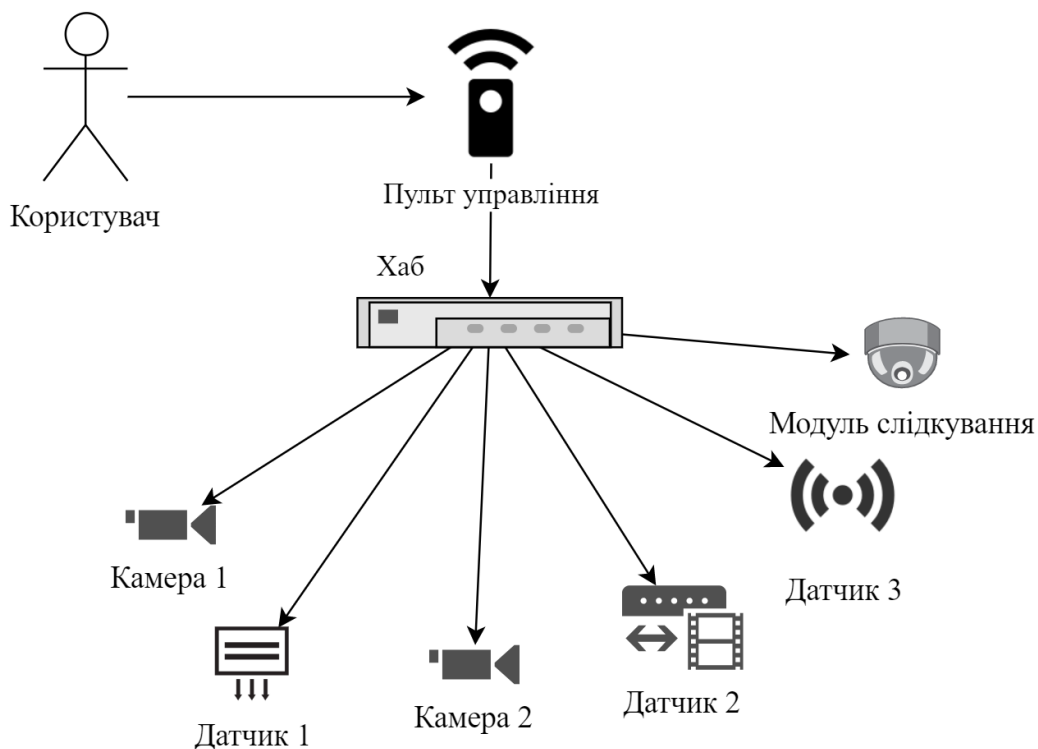


Рисунок 1.2 – Схема централізованого методу управління

Як бачимо, система розумного дому складається з основного модуля – хаба, до якого підключається інші пристрої. Хаб підтримує підключення до нього різних камер, датчиків, модулів тощо.

Користувач у свою чергу взаємодіє з цією системою або безпосередньо через хаб, або через спеціальний пульт управління.

Тобто, принцип функціонування систем розумного дому такий: до хабу (головного модуля системи) підключаються різні датчики, камери, тощо, з яких він зчитує дані та аналізує їх, надає інформацію користувачу. Таки принцип використовує більшість систем розумного дому. Наприклад, Ajax Systems [4] використовує Ajax Hub для підключення таких пристроїв як Ajax MotionProtect, Ajax doorprotect тощо.

Серед переваг такого підходу можна виділити:

- можливість легко підключити додатковий пристрій завдяки наявності основного модуля – хаба;
- централізація будь-якої взаємодії із системою;
- делегація низькорівневої взаємодії із компонентами до хаба та надання користувачу інтерфейсу керування.

Даний принцип майже в повній мірі відповідає поставленим вимогам, але є деякі недоліки таких систем:

- зазвичай, датчики виконують єдину функцію – збирають дані конкретного показника, тому компоненти системи не є багатофункціональними і незалежними; Також, такі датчики зазвичай є статичними;
- відсутність можливості додати власний пристрій до такої системи. На жаль, протоколи зв'язку компонент системи є закритими і не підлягають зміні стороннім особам;
- деякі моделі розумного дому використовують дротовий зв'язок.

Хоча така система не розрахована на роботу із рухомими об'єктами через статичність датчиків та низьку частоту запитів до підключених об'єктів (для

					<i>IT51.240BAK.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

Ajax Systems – 12 секунд), тут показано варіант архітектури, яка дозволяє підключити декілька об'єктів одночасно завдяки наявності хаба.

Також, така система використовує комфортний для кінцевого користувача метод комунікації із системою – надаються програмний додаток, який надає можливість переглядати стан підключених об'єктів та, в деяких реалізаціях, змінювати стан системи.

Хоча така система повною мірою не відповідає поставленим вимогам, її архітектура варта уваги для побудови власної реалізації програмно-апаратного комплексу управління системою рухомих об'єктів.

1.5 Висновки до розділу

У даному розділі була описана концепція програмно-апаратного комплексу управління системою рухомих об'єктів. Також, було висунуто функціональні та нефункціональні вимоги до даного комплексу, які необхідно врахувати та виконати під час детального проектування та реалізації – саме виконання цих вимог показує відповідність комплексу тому, що було заплановано. Були розглянуті та проаналізовані рішення, які тою чи іншою мірою виконують поставлені вимоги. Також було знайдено актуальний варіант комунікації: через радіоканал та актуальний варіант архітектури системи: із використанням хаба та програмного додатку для керування комплексом.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

2 ПРОЕКТУВАННЯ КОМПЛЕКСУ ТА ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1 Проектування структури комплексу

Проектування структури зумовлює подальший вибір методів комунікації, можливість подальшого масштабування, відмовостійкість, можливість фактичної реалізації тощо. Вибір структури є базою для подальшого дослідження та проектування.

Розглянемо можливі структури програмно-апаратного комплексу управління системою рухомих об'єктів, виберемо серед них найкращу, яку і будемо розглядати надалі більш детально та на базі якої будемо реалізовувати комплекс та усі його компоненти.

При виборі структури комплексу особливу увагу звернемо на можливість підключення декількох рухомих об'єктів до системи одночасно, можливість масштабувати та можливість бездротового зв'язку компонентів. Також, основним критерієм вибору буде виконання усіх поставлених функціональних та нефункціональних вимог.

На даному етапі будуть розглянуті базові варіанти, які у подальшому можуть бути розширені або модифіковані.

2.1.1 Структура з використанням декількох пультів управління

Така структура є найбільш простою та найбільш очевидною для управління декількома рухомими одночасно. Основна ідея такого методу управління є безпосереднє управління одним рухомих об'єктом за допомогою окремого пульта. Усі рухомі об'єкти укупі утворюють систему рухомих об'єктів.

Схематично таку структуру у контексті програмно-апаратного комплексу управління системою рухомих об'єктів можна побачити на рисунку 2.1.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Така структура складається із декількох пультів управління та, безпосередньо, об'єктів управління. Кожний пульт управління відповідає тільки за один відповідний об'єкт.

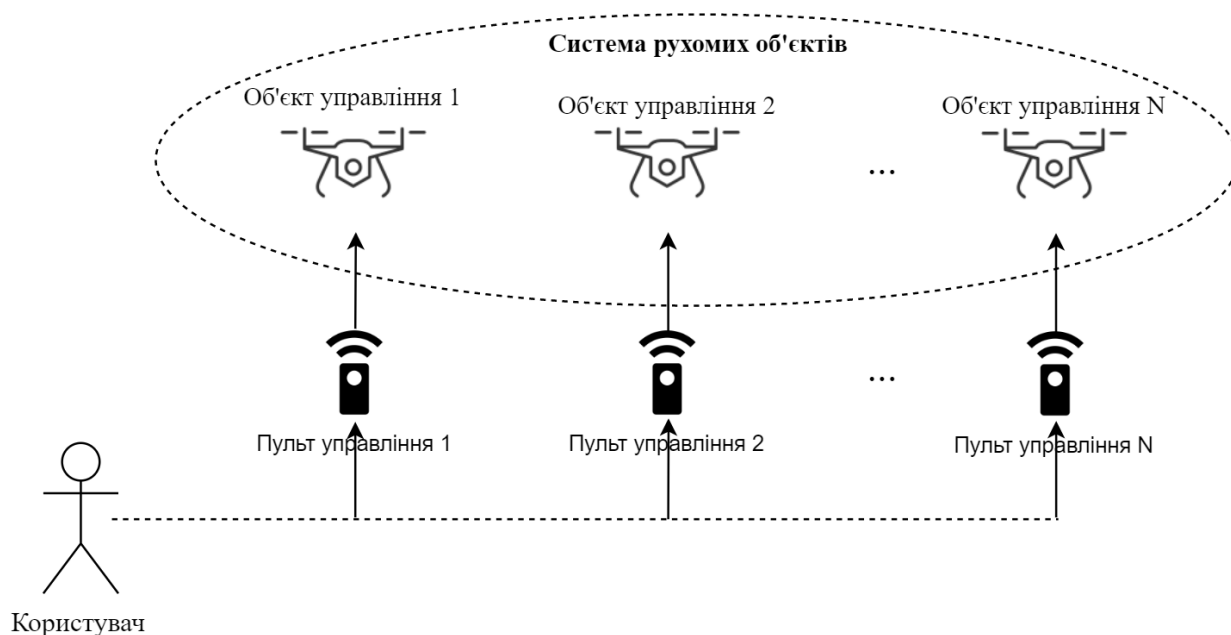


Рисунок 2.1 – Структура з використанням декількох пультів управління

Принцип використання є наступним: оператор вирішує яким з об'єктів зараз необхідно управляти, бере відповідний пульт управління та надсилає команди до обраного об'єкта.

Перевагами такої структури є:

- легкість в реалізації;
- простота інфраструктури;
- поломка одного з компонентів не веде до поломки усього комплексу;
- можна легко підключити новий рухомий об'єкт до даного комплексу.

Хоча, така структура і є простою, вона має багато недоліків.

Опишемо основні недоліки такої структури в контексті виконання поставлених функціональних та нефункціональних вимог:

- відсутня можливість одночасного управління системою рухомих об'єктів одним оператором в один момент часу;
- є необхідність підтримувати велику кількість фізичних пристроїв;

- управління такою системою є складною, та після додавання нових компонентів управління всіма ними унеможлиблюється;
- така структура є дорогою у своїй реалізації, адже для кожного об'єкту необхідно використовувати свій пульт управління;
- відсутня зручна візуальна інформація про підключені об'єкти до системи.

Отже, така структура є доволі простою та очевидною, виключає можливість поламки комплексу через один компонент, дозволяє відділити управління конкретним об'єктом.

Проте, така структура використовує у собі велику кількість компонентів, які є дорогими у розробці та складними у використанні. Також, збільшуючи кількість рухомих об'єктів, кількість компонентів системи збільшується на два (пульт та, безпосередньо, об'єкт керування). Тому, така структура не є вигідною для подальшого проектування та реалізації.

2.1.2 Структура з використанням кільцевої топології

Структура з використанням кільцевої топології дуже схожа до кільцевої топології комп'ютерних мереж [5]. Така структура вимагає від рухомих об'єктів взаємного обміну інформацією.

Основною ідеєю даної структури є те, що оператор посилає сигнал до одного, заздалегідь відомого, компонента системи рухомих об'єктів. Надалі він передає сигнал наступному компоненту системи. Таку структуру зображено на рисунку 2.2. Компонентами такої структури є пульт управління, та, безпосередньо, система рухомих об'єктів.

Опишемо алгоритм дій більш детально: оператор, за допомогою пульта управління, посилає сигнали до рухомого об'єкта у якому зазначено якому компоненту системи його адресовано. Надалі цей рухомий об'єкт передає, за необхідності, інформацію подальшому рухомому об'єкту (якщо пакет адресовано не йому). Так інформація може передаватися до останнього.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

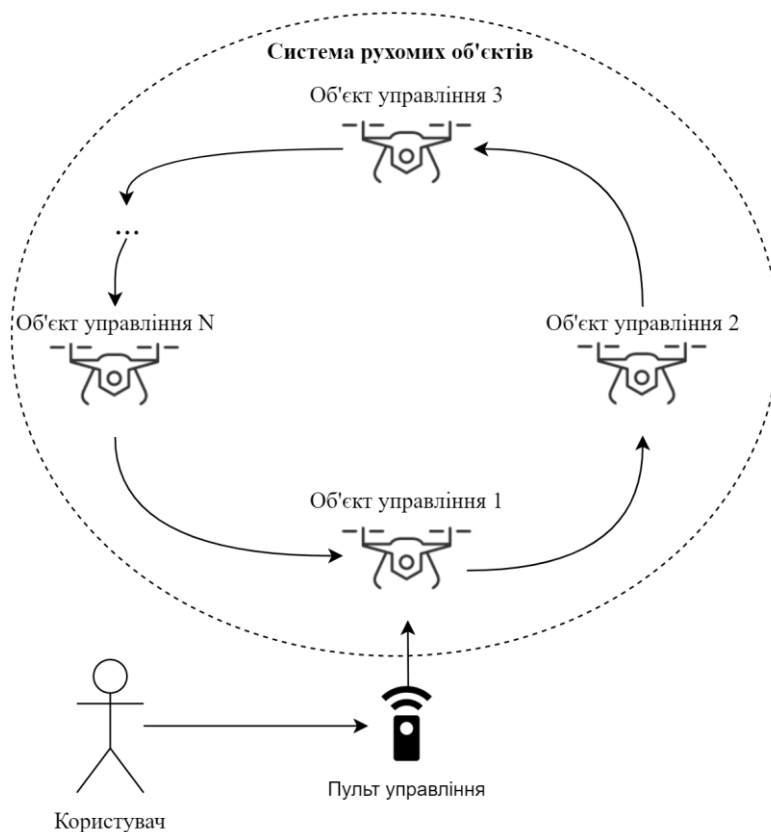


Рисунок 2.2 – Структура з використанням кільцевої топології

До переваг такої структури можна віднести:

- централізований доступ до контролю над усіма рухомими об'єктами;
- можливість додати до системи додатковий компонент;
- простота реалізації.

Але, не зважаючи, на такі переваги, така структура має досить багато недоліків:

- вихід з ладу однієї з ланки системи унеможливорює передачу даних до наступних ланок;
- конфігурація послідовності не може бути виконана під час безпосередньої експлуатації, тобто, вимагає попередньої конфігурації перед використанням;
- передача пакету до останнього у послідовності компонента займає більше часу, ніж до найпершого;
- кожний компонент системи має витратити зайві ресурси на переадресацію пакетів.

Отже, хоча така структура і задовольняє більшість поставлених функціональних вимог, підтримка такої структури є доволі складною через можливість виходу з ладу всієї системи через вихід з ладу однієї з ланок. Також, керування системою рухомих об'єктів, використовуючи таку структуру може бути доволі складним.

2.1.3 Структура з використанням хабу та клієнтського додатку

Структура з використанням хабу та клієнтського додатку була частково описана при огляді вже існуючих рішень та зарекомендована себе як оптимальне рішення для поставлених задач. Опишемо такий метод більш детально та проаналізуємо чи виконуються усі поставлені функціональні та нефункціональні вимоги.

Таку структуру у контексті програмно-апаратного комплексу управління системою рухомих об'єктів можна побачити на рисунку 2.3.

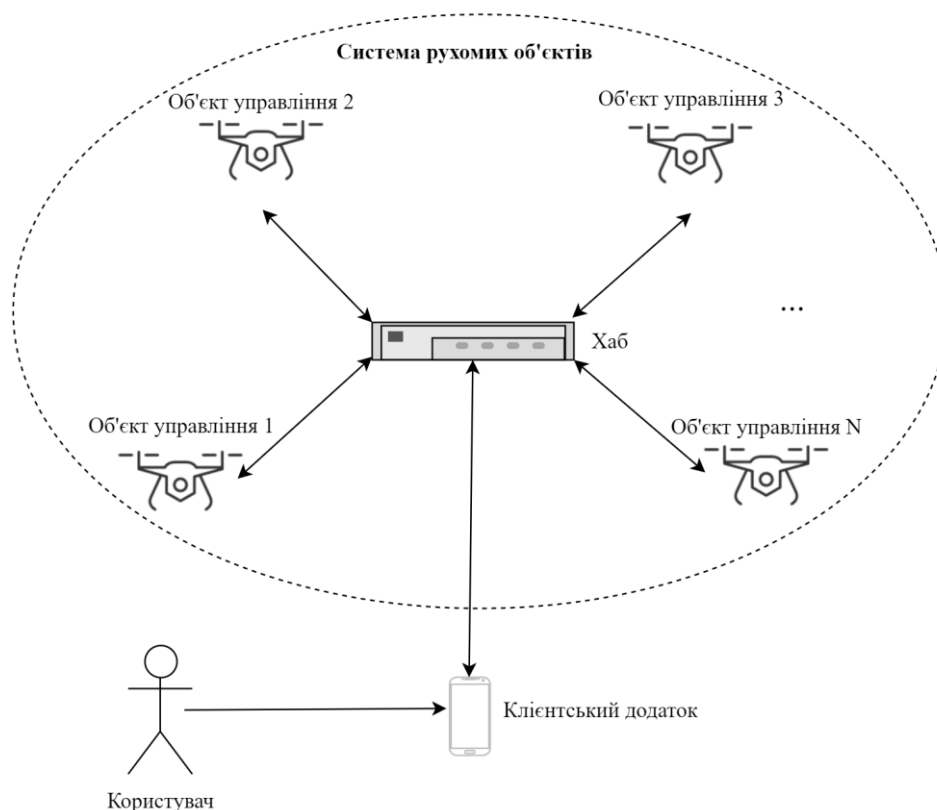


Рисунок 2.3 – Структура з використанням хабу та клієнтського додатку

Опишемо таку структуру більш детально. Основними компонентами є: хаб, клієнтський додаток, та, безумовно, система рухомих об'єктів.

Хаб реалізує логіку управління рухомими об'єктами через спеціальний канал зв'язку; також хаб підтримує безперервний зв'язок із рухомими об'єктами та надає всю інформацію до публічного інтерфейсу.

Клієнтський додаток виконує функцію зв'язку із хабом, зчитування необхідних даних та виведення їх у зручному вигляді для користувача.

Основними перевагами такої структури є:

- можливість підключити до комплексу декілька рухомих об'єктів;
- можливість централізованого управління всіма рухомими об'єктами одночасно;
- можливість виконувати усе управління через клієнтський додаток, у якому інкапсульовано усю логіку взаємодії із хабом;
- така структура може бути реалізована на бездротовому зв'язку, наприклад, радіозв'язок, Bluetooth тощо.

Серед недоліків такої структури можна виділити ймовірність поломки хабу, що унеможливить взаємодію із рухомими об'єктами. Але, така проблема є вирішуваною за допомогою реплікації хаба.

Така структура відповідає більшості поставлених функціональних та нефункціональних вимог щодо архітектурних рішень. Єдине, у чому така структура не є очевидною – забезпечення універсального методу комунікації.

Така структура також дозволяє легко модифікувати та масштабувати її за допомогою реплікації, масштабування вглиб під впливом покращення фізичних комплектуючих, дозволяє додати додаткові ланки, такі як ланка безпеки, ланка збереження даних, ланка обробки великих масивів даних тощо.

Отже, дана структура включає в себе такі компоненти:

- хаб, який централізує доступ до рухомих об'єктів;
- клієнтський додаток, який реалізує комунікацію із хабом та надає можливість користувачеві надсилати необхідну інформацію.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Також, така структура відповідає більшості поставлених вимог, дозволяє підключити одночасно декілька рухомих об'єктів та дозволяє масштабувати її.

2.1.4 Вибір структури серед проаналізованих

Попередньо було розглянуто декілька варіантів структури для програмно-апаратного комплексу управління системою рухомих об'єктів. Серед них:

- структура з використанням декількох пультів управління;
- структура з використанням кільцевої топології;
- структура з використанням хабу та клієнтського додатку.

Надалі, для зручності, будемо позначати їх варіант А, Б та В відповідно.

Кожна з цих структур мала свої переваги та недоліки, хоча кожна з них відповідає до вимоги керувати системою рухомих об'єктів. Тому, серед запропонованих структур виберемо ту, яка найкраще відповідає вимогам відмовостійкості, масштабованості, в контексті додавання нових ланок до системи, та простоти у використанні.

Для кожної структури поставимо оцінки від 1 до 5 та опишемо причини тієї чи іншої оцінки. Той варіант, який має найбільше балів буде взято для подальшого проектування та реалізації.

Оцінимо відмовостійкість для кожного з варіантів. Варіант А продовжить повністю функціонувати, якщо одна з ланок вийде з ладу адже усі ланки незалежні між собою – тому варіант А має 5 за пунктом відмовостійкість. Варіант Б припинить повноцінну передачу даних до наступних ланок при виході з ладу однієї, проте, усі попередні будуть функціонувати – тому оцінка 2. Варіант В повністю вийде з ладу якщо хаб пошкоджено, але, такий варіант дозволяє ввести репліки хаба та при виході з ладу одного підключити наступний, тому за відмовостійкість – 4.

Оцінимо можливість масштабування для кожного з варіантів. Варіант А не виключає можливості масштабування, дозволяє легко додати ланки до

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

комплексу, проте, такі дії вимагають використання додаткової апаратури та ускладнюють керування, тому за масштабованість – 3. Варіант Б дозволяє легко додавати нові ланки до системи, проте це зумовлює сповільнення передачі даних та підвищують ризик поламки системи, тому за масштабованість – 3. Варіант В дозволяє підключати велику кількість рухомих об'єктів до хабу, проте ця кількість обмежена; але таку кількість можна збільшити вводячи мережу хабів. Тому, за масштабованість варіант В отримує 5.

Оцінимо простоту використання для кожного з варіантів. Варіант А не є зручним у використанні. Це зумовлено тим, що додавання нової ланки включає додавання нового пульта управління, а чим більша їх кількість, тим управління становиться складнішим. Найкращим випадком є той випадок, коли у системі лише один компонент. За простоту використання варіант А отримує 2. Варіант Б потребує постійного вибору адресата при управлінні. Хоча, за умови зручного інтерфейсу це може не бути складністю. Також, варіант Б вимагає додаткової конфігурації системи перед її експлуатацією. Тому, за простоту використання варіант Б отримує 4. Варіант В надає зручний користувацький інтерфейс для управління всіма рухомими об'єктами, дозволяє читати необхідну інформацію про них. Проте, паралельне управління декількома об'єктами може ускладнитися при великій їх кількості. Тому за простоту використання варіант В отримує 4.

Оцінивши усі 3 варіанти за критеріями відмовостійкості, масштабованості, в контексті додавання нових ланок до системи, та простоти у використанні, можна підбити підсумки та обрати найкращий.

Для простоти, зведені результати відображено у таблиці 2.1.

Таблиця 2.1 – Зведені результати порівняння варіантів структури комплексу

	А	Б	В
Відмовостійкість	5	2	4
Масштабованість	3	3	5
Простота використання	2	4	4

Як видно з таблиці 2.1, найбільшу кількість балів набрав варіант В – структура з використанням хабу та клієнтського додатку. Так, даний варіант дійсно на перший погляд і є найкращим та найперспективнішим, що й було доведено детальним його аналізом.

Тому, для подальшого проектування та реалізації програмно-апаратного комплексу управління системою рухомих об'єктів буде обрано структуру з використанням хабу та клієнтського додатку.

2.2 Проектування хабу

2.2.1 Вибір методу комунікації

Вибір методу комунікації зумовлює вибір методу, за допомогою якого інформація буде передаватися від хаба до системи рухомих об'єктів, або до конкретної її ланки, технології, за допомогою якого інформація передається кінцевому користувачу.

Метод комунікації має відповідати функціональним та нефункціональним вимогам, поставленим до комплексу, а саме:

- мінімальна кількість рухомих об'єктів, яка підтримується – 6. Більша кількість об'єктів є допустимою;
- уся інформація між компонентами передається бездротовими каналами;
- відстань зв'язку має бути не менш, ніж 100 метрів.

Необхідно обрати метод комунікації із системою рухомих об'єктів та метод комунікації з кінцевим користувачем.

Найперший критерій, який необхідно розглянути – це наявність бездротового каналу. Це дозволить відкинути дуже велику кількість методів передачі інформації дротами.

Серед найкращих методів бездротового зв'язку є метод радіопередачі. Від даного методу є деякі розширення, такі як Bluetooth, Wi-Fi тощо. Розглянемо такі

					IT51.240БАК.002 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

технології та оберемо ту, яка найкраще відповідає поставленим вимогам та є найпростішою у експлуатації.

Для початку, розглянемо Bluetooth, який в останній час зарекомендував себе як надійний та широко використовуваний метод комунікації для технічних приладів.

Bluetooth - це стандарт бездротової технології для обміну даними між пристроями на коротких відстанях з використанням короткохвильових ультрависокочастотних-радіохвиль в промислових, наукових і медичних радіодіапазоні.

Він працює на частотах від 2402 до 2480 МГц або від 2400 до 2483,5 МГц, включаючи захисні смуги шириною 2 МГц в нижній частині і шириною 3,5 МГц у верхній частині. Bluetooth ділить передані дані на пакети і передає кожен пакет по одному з 79 призначених каналів Bluetooth. Кожен канал має смугу пропускання 1 МГц. Зазвичай він виконує 1600 тактів в секунду, з включеним адаптивним стрибкоподібним зміною частоти.

Офіційно радіостанції класу 3 мають радіус дії до 1 метра, клас 2, найбільш часто зустрічається в мобільних пристроях, 10 метрів і клас 1, в основному для промислових випадків, 100 метрів [6].

Так, технологія Bluetooth є широкою у спектрі можливостей та варіацій використання, проте має дуже малу відстань дії (до 100 метрів, при робочій відстані у 20 метрів).

Більш детально розглянемо технологію Wi-fi.

Wi-Fi – це сімейство радіотехнологій, яке зазвичай використовується для бездротових локальних мереж (WLAN) пристроїв і засноване на стандартах IEEE 802.11. Wi-Fi використовує кілька частин сімейства протоколів IEEE 802 і призначений для безшовного взаємодії зі своїм провідним родинним протоколом Ethernet.

Пристрої, які можуть використовувати технології Wi-Fi, включають в себе комп'ютери, мікроконтролери, планшети, телефони тощо. Сумісні пристрої

										Арк.
										26
Змн.	Арк.	№ докум.	Підпис	Дата	IT51.240БАК.002 ПЗ					

можуть підключатися один до одного через Wi-Fi через точку бездротового доступу, а також до під'єднаних пристроїв Ethernet і можуть використовувати їх для доступу в Інтернет. Така точка доступу має радіус дії близько 20 метрів та дуже чутлива до перешкод.

Різні версії Wi-Fi визначаються різними стандартами протоколу IEEE 802.11, причому різні технології радіозв'язку визначають діапазони, діапазони радіозв'язку і швидкості, які можуть бути досягнуті. Wi-Fi найчастіше використовує СВЧ-діапазон ISM 2,4 гігагерца. На близькій відстані деякі версії Wi-Fi, що працюють на відповідному обладнанні, можуть розвивати швидкість понад 1 Гбіт / с [7].

Технологія Wi-Fi має той самий недолік, що й технологія Bluetooth, а саме малий діапазон дії. Також, Wi-Fi є доволі складним в підтримці для невеликих апаратних систем.

Оберемо метод комунікації із системою рухомих об'єктів.

Для зв'язку із системою рухомих об'єктів необхідно мати незалежний модуль, який буде відповідати тільки за зв'язок із рухомими об'єктами. Такий модуль має знаходитися безпосередньо у хабі, адже усі обчислювальні операції та координація виконуються саме там.

Ключовими параметрами для даного зв'язку є відстань дії та можлива швидкість передачі даних.

Серед розглянутих методів найкращим є радіозв'язок на частоті 2.4 ГГц, адже він має більшу відстань дії, ніж Bluetooth та Wi-Fi, та має доволі швидку передачу даних.

Тому, методом передачі даних від хабу до системи рухомих об'єктів є використання радіо-модуля, який використовує радіозв'язок.

Такий метод передачі даних дозволить їх легко передавати на великі відстані. Також для такого методу зв'язку наявно багато методів перевірки правильності передачі даних, що дозволить захистити контроль рухомими об'єктами.

					<i>IT51.240BAK.002 ПЗ</i>	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

Оберемо метод комунікації з кінцевим користувачем.

Як було виділено у структурі комплексу, користувач для взаємодії з хабом використовує клієнтський додаток, який інкапсулює логіку взаємодії з хабом. Для цього хаб має включати публічний інтерфейс, через який відбувається передача даних користувачу.

Варіантів для такого публічного інтерфейсу може бути декілька:

- хаб є інтернет сервером, з точки зору користувача, та надає доступ до комунікації через веб-інтерфейс. У такому випадку веб-інтерфейс і буде користувацьким додатком;
- хаб передає сиру інформацію через один з каналів зв'язку, та користувацький додаток сам реалізує логіку відображення таких даних.

Для комунікації з кінцевим користувачем виберемо другий метод, а саму передачу сирової інформації хабом через один з каналів зв'язку – публічний інтерфейс. Такий вибір зумовлено тим, що це не змушує хаб виконувати додаткові операції з формування веб-сторінок, збереження користувацького інтерфейсу тощо.

Такий метод дозволяє розділити зони відповідальності: хаб відповідає за цілісність, правильність даних та їх передачу, а користувацький додаток – за відображення інформації користувачу та відображення користувацького інтерфейсу.

В якості каналу зв'язку між хабом та користувацьким додатком оберемо Bluetooth, адже інші методи тут будуть занадто складними та потужними для простої передачі сирих даних. Bluetooth дуже добре використовувати на невеликих відстанях, що є влучним для комплексу, адже, зазвичай, користувач знаходитиметься недалеко від хабу. Також, Bluetooth є легким у налаштуванні та достатнім за швидкістю передачі даних.

					IT51.240БАК.002 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

2.2.2 Протокол зв'язку

Протокол зв'язку між хабом, підключеними об'єктами та користувацьким додатком – є основним механізмом комунікації. Протокол - це обумовлені наперед правила передачі даних між пристроями. Такий протокол має складатися мінімальної та необхідної інформації для комунікації в обох напрямках: хаб – пристрій та пристрій – хаб. Протокол зв'язку дозволяє уніфікувати взаємодію між компонентами.

Спроекуємо базу для протоколу. База протоколу є необхідним мінімумом для ідентифікації компонентів у системі та подальшого її розширення для вирішення більш конкретних задач та підключення специфічних рухомих об'єктів.

Схема протоколу буде подана у вигляді списку необхідних полів та їх найменувань. Базовий протокол запиту від хабу має вигляд:

- версія протоколу;
- токен пристрою до якого йде звернення;
- ідентифікатор запиту;
- масив команд;
- масив додаткових параметрів;
- основна інформація про хаб.

Базовий протокол відповіді від пристрою має вигляд:

- версія протоколу;
- токен поточного пристрою;
- ідентифікатор відповіді на запит, та ідентифікатор запиту;
- статус коректності команди;
- пакет телеметрії.

Даний протокол включає в себе безперервну комунікацію між компонентами. Обов'язковою умовою є перевірка статусу усіх об'єктів системи на справність шляхом посилання до них запиту хабом кожні 3 секунди; об'єкти

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

системи маю відповідати на дані запити з максимально можливою швидкістю, але не більше, ніж 3 секунди. У разі виникнення ситуації, коли об'єкт не отримував запитів протягом 15 секунд – його задача повернутися на місце, де було отримано першу команду від хабу (за GPS координатою).

Як бачимо із специфікації протоколу, він дуже схожий на загальновідомий гіпертекстовий протокол передачі даних. Така форма протоколу дозволяє легко його модифікувати для підтримки більш конкретних задач.

Публічний протокол хабу передає інформацію у вільній формі тексту, або форматі JSON [8], в залежності від запиту та типу даних, які необхідно передати.

Обов'язковим пакетом телеметрії є: показники GPS, акселерометра, гіроскопа та час збору цих даних.

Також, команди, їх чисельні значення та очікувана реакція має бути додатково прописана в розширенні протоколу для конкретних задач.

Основною вимогою, яку необхідно накласти на хаб та усі рухомі об'єкти – повна підтримка даного протоколу. Компоненти не можуть взаємодіяти без повної його підтримки.

2.2.3 Вибір компонентів

Важливим етапом під час проектування будь-якої системи або комплексу є вибір компонентів. Правильно обрані компоненти зумовлюють легку подальшу реалізацію, можливість подальшої підтримки, можливість масштабування та можливість модифікування.

На даному етапі оберемо компоненти для центрального блоку програмно-апаратного комплексу управління системою рухомих об'єктів – хаба.

Основними задачами хабу є комунікація із системою рухомих об'єктів, надання інформації для публічного інтерфейсу та виконання обчислювальних операцій. Також, деякі характеристики було зумовлено нефункціональними вимогами до комплексу.

					<i>IT51.240BAK.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Для хабу необхідно обрати компоненти для:

- комунікації із рухомими об'єктами;
- комунікації із додатком користувача;
- виконання обчислювальних операцій, керування під'єднаними пристроями – тобто, основний модуль.

Оберемо основний модуль, до якого будемо надалі підключати модуль комунікації із рухомими об'єктами та модуль комунікації із додатком користувача.

При виборі такого модулю будемо звертати увагу на швидкодію такого модулю, можливість підключити додаткові компоненти та на наявність готових бібліотек для швидкої роботи, зосереджуючись на вирішенні проблеми, а не на низькорівневій архітектурі.

Розглянемо найпопулярніші в наш час рішення для подібних задач: плати на базі Raspberry Pi та на базі Arduino. Опишемо переваги та недоліки кожної з таких плат. Оберемо конкретну модель, яка буде базою для хаба.

Raspberry Pi - серія одноплатних комп'ютерів, створена Фондом Raspberry Pi. Ці плати є висококласними комп'ютерами, які дозволяють використовувати їх в широкому діапазоні: від навчальних цілей до великих, повноцінних систем і комплексів.

Частота процесора становить від 700 МГц до 1,4 ГГц. Об'єм вбудованої пам'яті становить від 256 МБ до 1 ГБ. Захищені цифрові карти в форматі MicroSDHC використовуються для зберігання операційної системи і пам'яті програм. Плати мають від одного до чотирьох портів USB. Для виведення відео підтримуються HDMI і композитне відео зі стандартним 3,5-мм роз'ємом для виведення звуку. Більш низький вихідний рівень забезпечується кількома контактами GPIO, які підтримують загальні протоколи, такі як I2C. Моделі B мають порт Ethernet, а деякі плати мають вбудований Wi-Fi і Bluetooth [9]. Плату Raspberry Pi 3 Model B+ можна побачити на рисунку 2.4.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31



Рисунок 2.4 – Плата Raspberry Pi 3 Model B+

До основних переваг Raspberry Pi можна віднести:

- потужна обчислювальна можливість;
- порівняно велика кількість оперативної пам'яті;
- вбудований Wi-Fi та Bluetooth.

Проте, така плата є занадто дорогою. Також, її обчислювальні можливості є набагато більшими ніж ті, які необхідно для побудови комплексу управління системою рухомих об'єктів. Для Raspberry Pi на ринку є невелика кількість радіомодулів та бібліотек для роботи з ними. Необхідно виділити багато часу для конфігурації плати, та тільки потім вирішувати поставлені задачі, що робить Raspberry Pi не самим якісним варіантом для використання.

Розглянемо плати Arduino.

Плати Arduino – це плати для побудови програмно-апаратних систем автоматизації та робототехніки.

Arduino - компанія з програмного забезпечення та програмного забезпечення з відкритим кодом, яка розробляє та виробляє одноплатні мікроконтролери та комплекти мікроконтролерів для створення цифрових

Arduino має великий вибір периферії, яку можна підключити до Arduino Uno. Тому, модуль радіопередачі для комунікації із рухомими об'єктами та Bluetooth модуль для комунікації із додатком користувача оберемо серед готових рішень Arduino. Це гарантує взаємну сумісність таких компонентів та дозволить використати готові бібліотеки із схожими інтерфейсами під час розробки.

Радіомодуль: Arduino NRF24L01 [11] який працює на частоті 2.4 ГГц, підтримує 128 каналів зв'язку та має швидкість передачі до 2 Мбіт на секунду. Зображення даного радіомодуля можна побачити на рисунку 2.6.

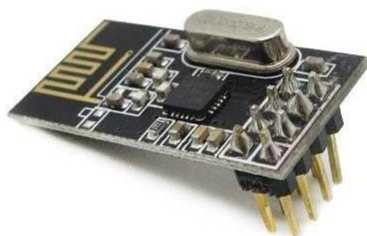


Рисунок 2.6 – Радіомодуль Arduino NRF24L01

Bluetooth модуль: Arduino HC-05 [12] модуль, який має швидкість передачі даних до 1 Мбіт на секунду. Зображення даного радіомодуля можна побачити на рисунку 2.7.



Рисунок 2.7 – Arduino HC-05 Bluetooth модуль

					IT51.240БАК.002 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Дані модулі забезпечать повну бездротову комунікацію хабу із зовнішнім світом. Перевагою саме цих модулів над конкурентними є те, що вони прості у налаштуванні, дешеві та мають відкритий вихідний код бібліотек для роботи з ними.

Програмування плат Arduino виконується на C-подібній мові – Arduino, що забезпечує швидкість виконання усіх команд та відсутність додаткових витрат на підтримку високих абстракцій мови програмування.

2.3 Проектування клієнтського додатку

2.3.1 Опис принципу роботи

Клієнтський додаток є основним механізмом взаємодії користувача із хабом, а, відповідно, із системою рухомих об'єктів.

Головна мета клієнтського додатку – інкапсулювати складну логіку взаємодії із хабом у зручний для користувача додаток, який буде відповідати на основні запитання щодо надсилання команд об'єктам, перегляду статусів об'єктів системи та хабу.

Схематично місце клієнтського додатку у програмно-апаратному комплексі управління системою рухомих об'єктів можна побачити на рисунку 2.8.

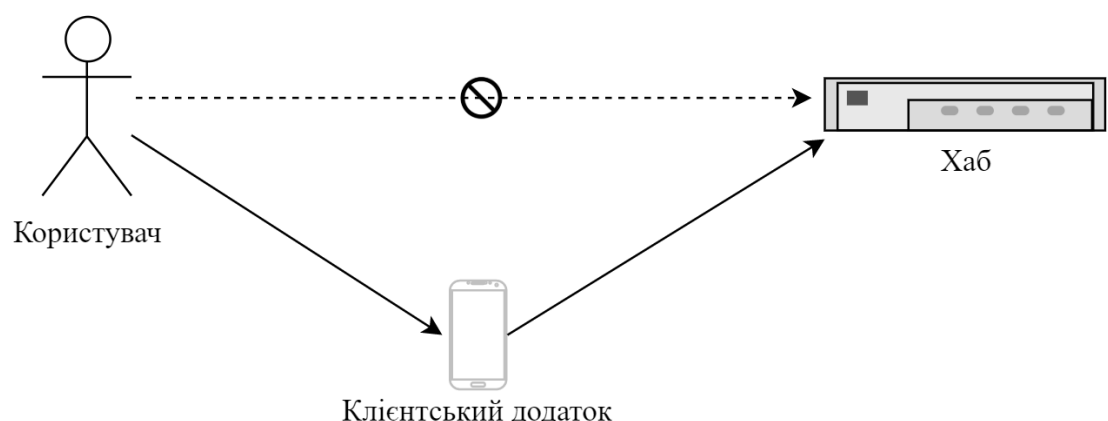


Рисунок 2.8 – Схематичне зображення місця клієнтського додатку у комплексі

Як видно із такого схематичного зображення, користувач не може взаємодіяти із хабом напряду. Причиною цьому є відсутність зрозумілого для людини публічного інтерфейсу хабу. Єдиним методом доступу до хабу ззовні є Bluetooth канал. Даним каналом хаб передає усю необхідну інформацію користувачу.

Тобто принципом роботи клієнтського додатку є адресація команд до хабу. У додатку виводиться уся необхідна інформація через інтерфейс користувача, надається можливість використання кнопок дій, полів введення даних. Дана інформація збирається і надалі передається до хабу. Хаб, у свою чергу, віддає відповідь, яка також відображається користувачу.

2.3.2 Цільова платформа

Як було визначено під час проектування центрального модуля комплексу – хабу, для доступу для публічного інтерфейсу необхідно використовувати технологію Bluetooth. Тому, вибір цільової платформи клієнтського додатку буде зумовлено вибором платформи, яка добре сумісна із такою технологією.

Серед основних кандидатів на роль цільової платформи можуть бути персональні комп'ютери, мобільні платформи.

Клієнтський додаток не має виконувати важких обчислювань, а повинен бути лише мостом для керування рухомими об'єктами через хаб. Тому, будемо орієнтуватися на легку інтеграцію з Bluetooth та простоту використання кінцевим користувачем.

Мобільні пристрої є набагато зручнішими ніж персональні комп'ютери, при виконанні поставлених задач через:

- портативність – надається можливість переносити пульт управління у вигляді клієнтського додатку на мобільній платформі до місця використання системи рухомих об'єктів та до хабу безпосередньо;

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

- простота використання – мобільні пристрої за допомогою зручного та ергономічного інтерфейсу дозволяють виконувати навіть складні задачі.
- зазвичай мобільні пристрої мають вбудовану підтримку Bluetooth, що дозволяє програмувати додаток більш швидко.

Тому, як платформу для клієнтського додатку оберемо мобільні пристрої через їх портативність, простоту використання та підтримку Bluetooth. Оберемо операційну систему.

Серед головних кандидатів можна виділити Android і iOS. Розглянемо дані операційні системи окремо.

Android – операційна система, розроблена для мобільних пристроїв, електронних книг, смарт-годин, телевізорів та інших пристроїв. Операційна система Android має ядро Linux і має власну реалізацію Java Virtual Machine – Dalvik.

Компанія Google надає можливість писати власні програми на мові програмування Java, використовуючи бібліотеки Android Native Development Kit. Також, надається можливість імпортувати бібліотеки і компоненти додатків мови C та інших [13].

Серед переваг Android можна виділити:

- Android має найкращі показники з вбудованих функцій, як інтернет-серфінг, взаємодія з додатками Google;
- Android надає встановлювати програми без інтернету, а також підтримує налагодження додатків по USB, що дозволяє писати і тестувати власні програми на своєму Android-пристрої;
- широка підтримка вбудованої периферії, такої як Bluetooth, GPS тощо.

Далі, розглянемо операційну систему IOS.

IOS – Операційна система, що випускається компанією Apple. Призначена для смартфонів, планшетів і програвачів

Для створення програмного забезпечення під IOS використовується мова програмування Objective-C або Swift і середовище розробки Xcode. Також,

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

можна розробляти програмні додатки з використанням мов програмування C та C ++ [14].

Серед переваг IOS можна виділити

- інтерфейс, зрозумілий на інтуїтивному рівні;
- відсутність вразливостей безпеки;
- багатозадачність (мале споживання ресурсів).

Серед недоліків iOS можна виділити:

- відсутня можливість прямого доступу до файлової системи;
- низька підтримка програмного доступу до периферії, такої, як Bluetooth.

На даний час близько 60% всіх мобільних пристроїв знаходяться під управлінням операційної системи Android, коли під керуванням IOS – менше 35%.

Проаналізувавши операційні системи Android і iOS, прийнято рішення, що цільова платформа для розробляється буде Android.

Дане рішення було прийнято за далі описаним причин. По-перше, Android – відкрита платформа, в зв'язку з чим, на ній можна реалізувати більше функцій. По-друге, Android надає широкий, у порівнянні з iOS, вибір API і різних бібліотек. По-третє, Android має більш широку аудиторію, ніж інші операційні системи.

2.3.3 Вибір мови програмування

У попередньому пункті, цільової операційною системою був обраний Android. Для Android необхідно написати програму на мові програмування, компілятори яких здатні генерувати байт-код для Dalvik.

На даний час найпопулярнішим і раціональним способом є написання Android-додатки на строго типізований об'єктно-орієнтованій мові програмування Java. Дана мова є об'єктно-орієнтованою мовою програмування і має строгу типізацію.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

Перелічимо основні особливості мови:

- програми, написані цією мовою, транслюються в спеціальний байт-код, який може бути розпізнаний на будь-якому пристрої, на якому встановлена віртуальна машина Java;
- програми на Java, під час виконання, повністю контролюються віртуальною машиною, яка підтримує систему безпеки. Будь-які дії, які перевищують повноваження програми можуть викликати переривання;
- використання віртуальної машини призводить до деякого зниження продуктивності.

Задля підвищення продуктивності до Java прийнято низку удосконалень:

- байт-код транслюється в машинний код безпосередньо під час роботи програми, дозволяючи зберігати версії класів в машинному кодї;
- у стандартних бібліотеках використовується код, орієнтований на платформу.

Серед основних можливостей Java можна виділити: автоматизована робота з пам'яттю, система обробки винятків, велика стандартна бібліотека класів, можливість створення мережевих додатків, можливість створення багатопотокових додатків, система узагальнень, можливість виразів лямбда виразів та замикань.

З усього вищезгаданого стає зрозумілим, що Java є ідеальним варіантом для програмування користувацького додатку програмно-апаратного комплексу управління системою рухомих об'єктів через наявність великої кількості бібліотек для роботи з Android, швидкості дії та можливості абстрагуватися від задач низького рівня та сконцентруватися на вирішенні поставлених задач.

2.3.4 Проектування користувацького інтерфейсу

Користувацький інтерфейс – це та частина продукту, яку користувач бачить при використанні, саме інтерфейс зумовлює простоту використання та

									Арк.
									39
Змн.	Арк.	№ докум.	Підпис	Дата					

можливості програмного продукту. Правильно спроектований користувацький інтерфейс, його переходи, дизайн, розташування компонентів керування є одними з ключових показників якості програмного продукту.

Основними задачами, які мають бути реалізовані у користувацькому інтерфейсі застосунку:

- показ усіх підключених об'єктів до комплексу;
- перегляд інформації про хаб;
- надсилання команд до конкретних об'єктів, або до усієї системи одночасно.

Вирішенням таких задач є багатосторінковий користувацький інтерфейс, на кожній сторінці якого відображено необхідні компоненти.

Сторінками користувацького інтерфейсу для вирішення усіх вказаних вище задач є:

- сторінка підключення до хабу;
- сторінка перегляду інформації про хаб;
- сторінка зі списком усіх підключених компонентів системи;
- сторінка опису обраного підключеного компоненту з елементами керування.

Сторінка підключення до хабу надає можливість користувачеві обрати необхідне Bluetooth з'єднання та, натиснувши, відповідну кнопку підключення – підключитися до хабу, що дозволяє виконувати усі необхідні задачі.

Основна задача сторінки перегляду про хаб – надати користувачеві можливість подивитися усі метрики хабу, такі як: його статус, чи наявні та справні усі необхідні модулі, положення у просторі, ідентифікатор та усі інші можливі дані.

Сторінка зі списком усіх підключених компонентів системи дозволяє користувачеві переглянути всі підключені компоненти системи у єдиному списку. Кожний елемент списку має ідентифікатор об'єкта, його статус та мінімально необхідну інформацію для подальшої ідентифікації об'єкта.

									Арк.
									40
Змн.	Арк.	№ докум.	Підпис	Дата					

Сторінка опису обраного підключеного компоненту з елементами керування з'являється тоді, коли користувач обрав конкретний компонент із списку усіх підключених компонентів системи. На даній сторінці відображається детальна інформація щодо підключеного компоненту: його положення у просторі, історія всіх надісланих йому команд; також відображається інтерфейс для керування ним. Інтерфейс керування складається із елементів (кнопки, списки команд тощо), які допомагають користувачеві обрати та надіслати команду рухомому об'єкту.

Основний принцип роботи користувача із застосунком:

- встановлення Bluetooth з'єднання із хабом;
- перегляд інформації хабу;
- перегляд підключених компонентів та вибір конкретного компоненту для керування ним;
- перегляд інформації про підключений компонент; керування ним; надалі – перехід до списку підключених компонентів.

Тобто, користувач, підключившись до хабу, обирає необхідний компонент, яким необхідно керувати, надалі, безпосередньо, керує ним та повертається до списку компонентів для подальшого вибору компоненту, яким необхідно керувати.

2.4 Розробка то опис діаграми компонентів

У UML діаграма компонентів зображує, як компоненти з'єднані між собою для формування великих компонентів або програмних та програмно-апаратних систем. Вони використовуються для ілюстрації структури складних систем. [15]

Діаграма компонентів програмно-апаратного комплексу управління системою рухомих об'єктів складається з двох основних вузлів – клієнтський застосунок та хаб. Діаграму компонентів зображено на кресленику IT51.240БАК.003 Д1.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

Основним інтерфейсом взаємодії між цими вузлами є Bluetooth канал, через який хаб надає основну інформацію, а, клієнтський застосунок робить підключення до цього каналу та зчитує дані.

Програмно-апаратний комплекс управління системою рухомих об'єктів має два публічні інтерфейси:

- інтерфейс для надання команд через графічний користувацький інтерфейс;
- інтерфейс для передачі даних радіоканалом до рухомих об'єктів.

Клієнтський застосунок має два основних компоненти та одну підсистему, а саме: компонент графічного користувацького інтерфейсу, Bluetooth модуль та систему обробки даних відповідно. Користувацький інтерфейс та використовує систему обробки даних для того, щоб отримати оброблені дані від хабу та для того, щоб надати команди до рухомих об'єктів. Система обробки даних є своєрідним мостом між інтерфейсом користувача та Bluetooth модулем, який надсилає команди до хабу. Така система використовує бібліотеки для роботи із Bluetooth модулем, та бібліотеку обробки та генерації запитів.

Хаб складається з двох основних підсистем та двох компонентів: підсистема синхронізації компонентів, модуль роботи із системою рухомих об'єктів, Bluetooth модуль та радіомодуль відповідно.

Через радіомодуль у хабі виконується комунікація із рухомими об'єктами, тому саме через цей компонент реалізовано публічний інтерфейс. Система синхронізації компонентів виконує основну задачу хабу з точки зору взаємодії із застосунком користувача – зчитує дані через відповідну бібліотеку роботи із Bluetooth, та оброблює таку інформацію. Модуль роботи із системою рухомих об'єктів використовує дану систему для зчитування запитів користувача у зручному форматі, та для відправки результатів команд назад користувачу. Тобто, система синхронізації є посередником між командами, які надіслав користувач через застосунок користувача та радіомодулем, який виконує відправку команд до системи рухомих об'єктів.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

2.5 Опис алгоритмів

2.5.1 Алгоритм перевірки статусу компонентів системи

Як описано у базовому протоколі зв'язку, обов'язковою умовою комунікації хабу із системою рухомих об'єктів є перевірка статусу усіх об'єктів на справність шляхом посилання до них періодичного запиту.

Такий алгоритм є методом контролю рухомих об'єктів, перевірки їх справності та реакцій на надіслані команди. Алгоритм перевірки статусу компонентів системи дозволяє своєчасно побачити несправності та усунути їх.

Кроки алгоритму перевірки статусу компонентів системи зі сторони хабу:

- надіслати пакет із запитом статусу компонента;
- очікувати на відповідь протягом трьох секунд;
- якщо відповідь не була надана протягом трьох секунд – повторити той самий запит; якщо відповідь надано – повернутися очікувати три секунди періоду відправки;
- повторювати кроки з повторною відправкою запиту та очікуванням трьох секунд на відповідь до того моменту, поки відповідь не буде надано;
- якщо відповідь не надано після п'яти ітерацій відправки запитів – позначити компонент як несправний;
- продовжувати надсилати запити статусу кожні п'ять секунд; якщо відповідь буде надано – повернутися до стандартного режиму відправки з періодом 3 секунди.

Схему даного алгоритму подано на кресленику IT51.240БАК.004 Д2.

Поданий алгоритм необхідно виконувати для всіх компонентів системи паралельно. Тобто перевіряти статуси потрібно для усіх компонентів системи без виключення у паралельному режимі, або, послідовно, із мінімально можливою затримкою. Періодичність та порівняно висока частота запитів статусу (три секунди) такого алгоритму є необхідністю, адже хаб підтримує

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

зв'язок із рухомими об'єктами, які часто міняють своє положення та мають багато внутрішніх компонентів, які можуть вийти з ладу у будь-який момент.

Кроки алгоритму перевірки статусу компонентів системи зі сторони рухомого об'єкта:

- при ініціалізації зберегти початкове положення у просторі рухомого об'єкта;
- розпочати відлік секунд від минулого запиту хабу (при ініціалізації – від моменту встановлення радіозв'язку із хабом);
- очікувати на запит статусу від хабу;
- якщо запит отримано – зібрати дані щодо статусу компонентів системи та надіслати їх хабу; встановити лічильник часу у початкове положення;
- якщо запит не було отримано протягом п'ятнадцяти секунд – повернутися на місце, з якого було розпочато роботу.

Дані алгоритми є обов'язковими для реалізації як на хабі, так і на рухомому об'єкті, адже перевірка статусу компонентів системою є обов'язковою частиною протоколу зв'язку, який, у свою чергу є обов'язковим для взаємодії хабу і рухомих об'єктів.

2.5.2 Алгоритм передачі команд до рухомих об'єктів

З точки зору кінцевого користувача, керування рухомих об'єктом є лінійною задачею передачі команд із клієнтського застосунку безпосередньо до рухомого об'єкта, проте команди проходять багато кроків перед тим, як бути обробленими рухомих об'єктом та відповідь на них буде відображена.

Алгоритм передачі команд від користувача до рухомих об'єктів зумовлює послідовність кроків по яких проходить команда рухомому об'єкту та послідовність кроків як результат цієї команди відображається на користувацькому застосунку.

Основними кроками даного алгоритму є:

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

- вибір команди на користувацькому застосунку та натискання клавіши для відправки;
- формування запиту до хабу;
- відправка запиту по Bluetooth каналу та відображення того, що команду було відправлено;
- зчитування команди хабом та відправка відгуку про те, що команду прочитано; відображення даного статусу на клієнтському застосунку;
- обробка запиту хабом;
- формування запиту до рухомого об'єкта та відправка запиту радіоканалом до рухомого об'єкта;
- обробка запиту рухомих об'єктом; формування результату та відправка результату по радіоканалу до хабу;
- обробка результату хабом та передача результату по Bluetooth каналу;
- обробка та відображення результатів клієнтським застосунком.

Діаграму послідовності передачі даних від користувача до рухомого об'єкта показано на кресленику IT51.240БАК.005 ДЗ.

Тобто, такий алгоритм складається з послідовної передачі та обробки даних від клієнтського застосунка до хабу по Bluetooth каналу, від хабу до рухомого об'єкта та у зворотному порядку для надання відповіді.

Передача даних радіоканалом у даному алгоритмі відбувається у три кроки:

- передача даних від хаба до рухомого об'єкта;
- зчитування даних рухомих об'єктом;
- передача відповіді до хабу.

Хоча задача є доволі тривіальною, тут можливі випадки, коли той чи інший пакет не було передано. Тому можливо три варіанти розвитку подій: пакет запиту успішно доставлено, відповідь успішно отримано; пакет запиту не було доставлено; пакет запиту доставлено, відповідь не отримано.

Стандартний алгоритм передачі даних радіоканалом полягає у наступному: хаб починає зв'язок, передаючи пакет даних до рухомого об'єкта. Як тільки весь

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

пакет передано, він очікує отримання пакету відповіді. Коли рухомий об'єкт отримує пакет, він відправляє пакет відповіді хабу.

Якщо пакет при передачі від хабу було втрачено, необхідно повторно відправити той самий пакет. Сценарій такого випадку: після того, як хаб вперше передав пакет, він чекає отримання пакету відповіді. Якщо хаб не отримує пакет протягом часу очікування, пакет передається повторно. Коли повторно переданий пакет прийнятий рухомим об'єктом, він відсилає пакет відповіді.

При умові втрати пакету відповіді від рухомого об'єкта, необхідно повторно відправити пакет. Тут, навіть якщо рухомий об'єкт приймає пакет у першій спробі, через те, що було втрачено пакет відповіді, хаб вважає, що об'єкт взагалі не отримував пакет. Тому, після сплину часу очікування – хаб повторно передає пакет. Надалі, коли об'єкт приймає пакет, він міститиме той самий ідентифікатор, тому він відкидає його і знову відсилає ту саму відповідь.

2.6 Висновки до розділу

У даному розділі було спроектовано центральний компонент програмно-апаратного комплексу управління системою рухомих об'єктів – хаб. Було обрано найкращий варіант комунікації хабу із системою рухомих об'єктів – радіозв'язок, та варіант комунікації із клієнтським додатком – Bluetooth. Такі методи покривають усі потреби для бездротової комунікації. Також, було розроблено протоко завдяки якому хаб буде взаємодіяти із системою рухомих об'єктів. Підтримка даного протоколу є обов'язковою умовою для правильної роботи та поведінки комплексу і рухомого об'єкта. На даному етапі було обрано конкретні фізичні компоненти системи та побудовано схему їх підключення.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ

3.1 Реалізація клієнтського застосунку

В межах даного розділу розглянемо програмну реалізацію клієнтського застосунку на мобільній платформі Android, використовуючи Android Development Kit, мови програмування Java та технології Bluetooth.

Основною складовою клієнтського застосунку є користувацький інтерфейс, компоненти якого було виділено під час проектування, а саме:

- сторінка підключення до хабу;
- сторінка перегляду інформації про хаб;
- сторінка зі списком усіх підключених компонентів системи;
- сторінка опису обраного підключеного компоненту з елементами керування.

Першою сторінкою є сторінка підключення до хабу. Дану сторінку під час роботи застосунку зображено на рисунку 3.1. Детальну діаграму станів користувацького інтерфейсу наведено на кресленнику IT51.240БАК.006 Д4.

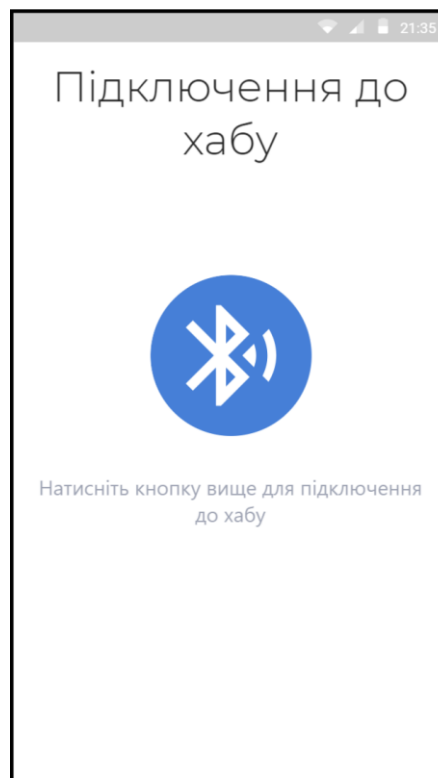


Рисунок 3.1 – Сторінка підключення до хабу

									Арк.
									47
Змн.	Арк.	№ докум.	Підпис	Дата					

Дана сторінка складається із одного основного компонента – кнопки заклику до дії підключитися до хабу та підказки що потрібно зробити. Після натискання даної кнопки користувача буде переведено на сторінку хабу, якщо він уже має Bluetooth підключення із хабом, інакше – відкриється діалог вибору Bluetooth підключень, серед яких необхідно обрати хаб.

Для використання Bluetooth, запрошено основні дозволи `android.permission.BLUETOOTH` та `android.permission.BLUETOOTH_ADMIN`. Завдяки даним дозволам, можна доступитися до Bluetooth із розроблюваного застосунку. Bluetooth підключення реалізовано для двосторонньої взаємодії із хабом: реалізовано підключення, у якому можливе як читання даних, так і їх надсилання. Для того, щоб користувацький інтерфейс залишався завжди доступний, читання та надсилання реалізовано у окремому потоці, який наслідувано від класу `Thread`.

Після успішного підключення до хабу, користувач потрапляє на сторінку, де показана детальна інформація про хаб. Дану сторінку зображено на рисунку 3.2.

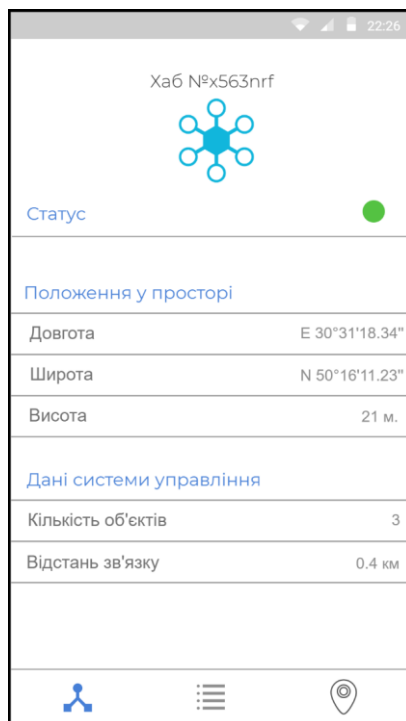


Рисунок 3.2 – Сторінка перегляду інформації про хаб

На даній сторінці відображено усю інформацію, яку було отриману від хабу після підключення до нього. Зверху показано унікальний ідентифікатор хабу, та статус хабу. Далі показано усю іншу доступну інформацію: положення хабу у просторі (довгота, широта та висота над поверхнею) та дані системи управління. Ці дані характеризують хаб у контексті комунікації із системою рухомих об'єктів, а саме: кількість підключених до хабу об'єктів та максимальна відстань, на якій хаб може тримати зв'язок із рухомими об'єктами.

У поточній реалізації показано лише базову інформацію щодо хабу, проте, є можливість багато інших властивостей хабу, які будуть характеризувати його більш детально. На даній сторінці основними програмними компонентами є TextView та ImageView для відображення статичної інформації.

У нижній частині екрану наявне меню, яке дозволяє швидко переходити між сторінками. Таке меню дозволяє користувачу орієнтуватися де саме у програмі він зараз знаходиться, та куди може перейти. Програмно меню реалізовано за допомогою TabLayout класу у Android. Натиснувши на другий компонент меню, користувач потрапляє на сторінку зі списком об'єктів, зображену на рисунку 3.3.

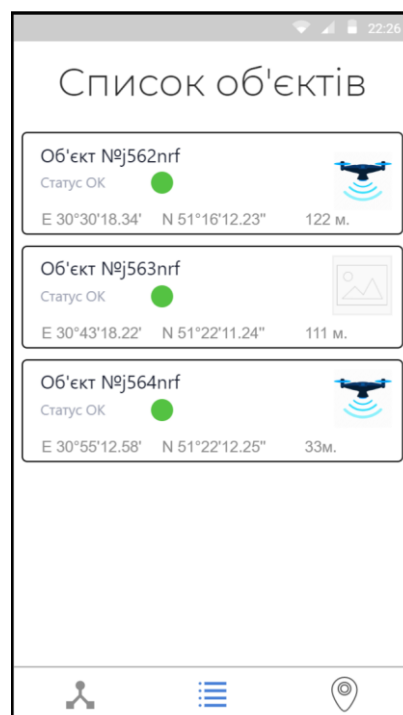


Рисунок 3.3 – Сторінка зі списком усіх підключених об'єктів системи

					<i>IT51.240BAK.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

Після переходу на дану сторінку, другий елемент меню виділено синім кольором, що є індикатором, що користувач знаходиться на другій сторінці. На даній сторінці відображено список підключених до комплексу рухомих об'єктів. Список виведено за допомогою компоненту ListView.

На кожному елементі списку зазначено унікальний ідентифікатор об'єкта, його статус за допомогою тексту та візуального кольорового індикатора (зелений позначає нормальний статус, червоний – компонент не справний, або зв'язок з ним втрачено). Також, показано координати об'єкта у просторі та його іконку. Реалізовано відображення базового списку параметрів об'єкта на списку, проте, можливе додання більш важливої інформації для конкретного випадку.

Даний список дозволяє перейти до сторінки, на якій відображено детальну інформацію про об'єкт. Для цього достатньо натиснути на необхідний елемент у списку. Такий перехід реалізовано за допомогою реалізації слухача подій ItemClick, який створює об'єкт Intent із використанням двох Activity.

Після натискання на один з елементів списку, користувач потрапляє на екран, подібний до того, що зображено на рисунку 3.4.

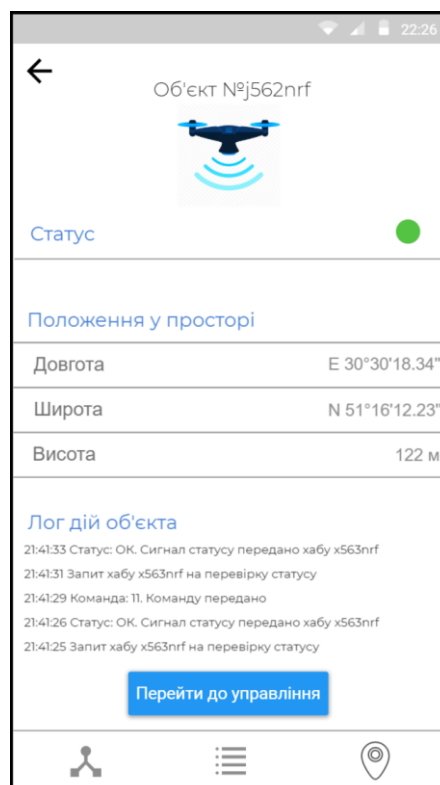


Рисунок 3.4 – Сторінка інформації про підключений об'єкт

					IT51.240БАК.002 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

На даній сторінці відображено детальну інформацію про рухомий об'єкт. Зверху сторінки зображено іконку об'єкта, його унікальний ідентифікатор та статус. Далі, виведено інформацію про положення об'єкта у просторі та список усіх команд та відповідей від об'єкта. Також, надається можливість перейти до управління даним об'єктом за допомогою синьої кнопки з написом «Перейти до управління», або повернутися до списку об'єктів, натиснувши кнопку у вигляді стрілки у верхній частині екрану.

У поточній реалізації виведено лише базову інформацію про рухомий об'єкт, якої може бути не достатньо для повного аналізу його стану та прийняття рішення щодо наступної команди для даного об'єкта. Тому, на даний екран слід додати інформацію, яка є необхідною для вирішення конкретної задачі, наприклад, відеопотік від камери об'єкта, показники різних датчиків тощо.

Натиснувши кнопку «Перейти до управління» користувач потрапляє на сторінку, де надано можливість надіслати команди обраному об'єкту. Дану сторінку зображено на рисунку 3.5 Тут відображається інформація про те, саме яким об'єктом зараз йде управління (ідентифікатор об'єкта та його зображення) та інформація про статус об'єкта.

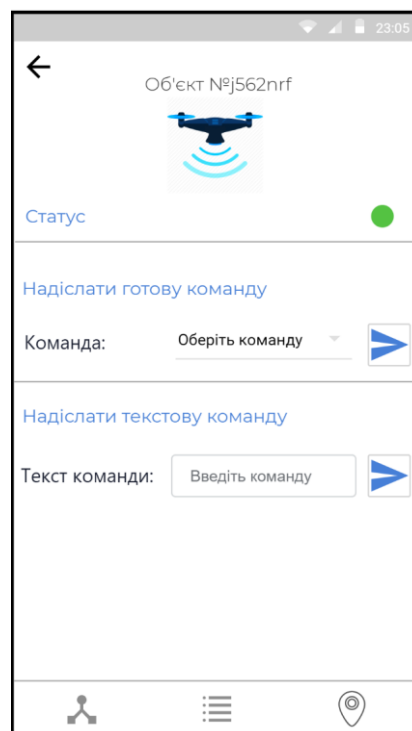


Рисунок 3.5 – Сторінка управління об'єктом

									Арк.
									51
Змн.	Арк.	№ докум.	Підпис	Дата					

У поточній реалізації надано можливість обрати команди серед запропонованих, а саме: перейти до початкового положення, надати пакет інформації, перевірити статус та інші. Також, можливо надіслати команди у вигляді тексту команди: це має бути команда у форматі, який може бути розпізнано даним рухомим об'єктом. Для того, щоб надіслати команду, необхідно або вибрати її зі списку, або написати її у відповідне текстове поле та натиснути синю кнопку відправки.

Такий формат передачі даних дозволяє виконати базове управління об'єктом, спробувати його основні можливості. Для досягнення більшої інтерактивності, на дану сторінку може бути додано такі компоненти як джойстики, кнопки управління тощо.

У випадку, якщо така команда не підтримується застосунком на даний момент, користувач отримує попередження, зображене на рисунку 3.6.

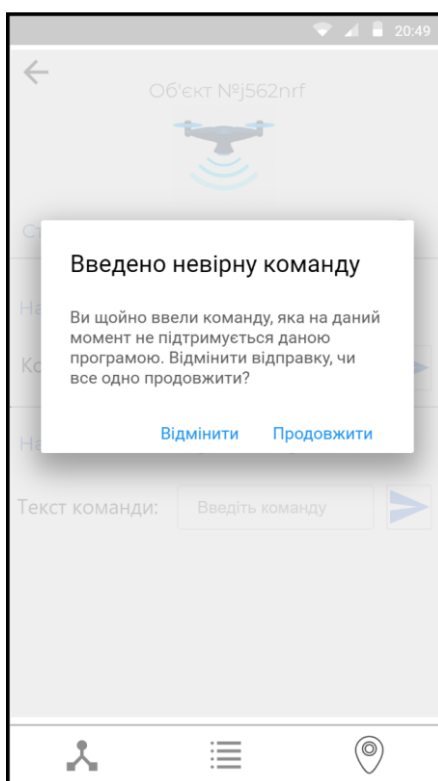


Рисунок 3.6 – Попередження про введену неправильну команду

Користувачеві надається можливість або скасувати дану команду, або все одно відправити її до хабу. У разі відправки, застосунок не буде приводити команду до загальноприйнятого виду, а буде передавати команду у тому вигляді,

у якому її надав користувач. У разі відміни відправки команди, користувача буде повернуто до тої сторінки, з якої він відправляв команду та червоним кольором буде виділено поле, у якому присутня дана команда.

При натисканні на третій елемент у меню, користувач потрапляє на сторінку, де у інтерактивній формі показано фактичне положення об'єктів на мапі. Тут користувач може передивитися де знаходиться кожний об'єкт у режимі реального часу. Дану сторінку зображено на рисунку 3.7

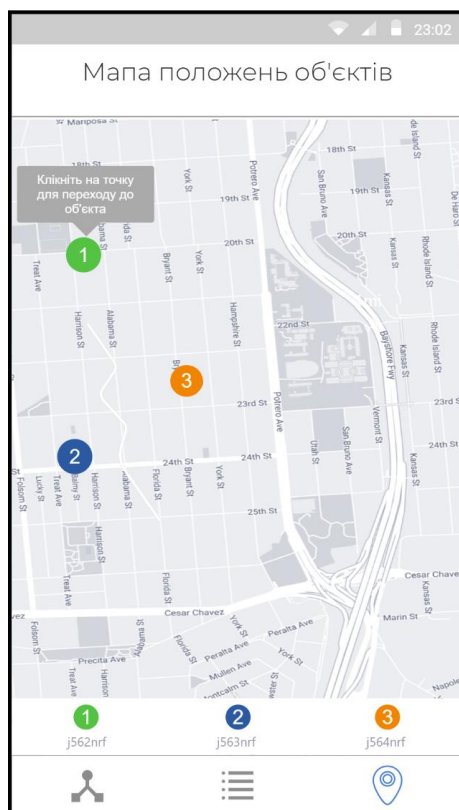


Рисунок 3.7 – Сторінка мапи положень об'єктів

Поточна реалізація клієнтського застосунку включає в себе такий функціонал лише як прототип для візуалізації можливого подальшого рішення та розвитку застосунку.

Користувачеві надається можливість переглянути усі підключені об'єкти у нижній частині сторінки з можливістю горизонтального прокручування списку та на інтерактивній карті місцевості. Натискаючи на елемент на карті, або на елемент у списку знизу сторінки, користувач потрапляє на сторінку перегляду інформації про рухомий об'єкт.

					IT51.240БАК.002 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Реалізація хабу

Компонентами, за допомогою яких має бути реалізовано хаб, було обрано плату Arduino Uno, радіомодуль Arduino NRF24L01 та Bluetooth модуль Arduino HC-05.

Першим етапом для реалізації хабу є підключення компонентів у єдиний механізм, підключення відповідних портів вводу-виводу, подача живлення правильної напруги.

Для функціонування Arduino Uno, необхідною напругою є 7-12В. Живлення Arduino може бути підключено або за допомогою USB у відповідний порт, або за допомогою іншого джерела у порт VIN. Робочою напругою Arduino є 5В, проте, є можливість виводу напруги 5В та 3.3В. Макет підключення компонентів зображено у додатку А.

Arduino HC-05 для правильної роботи вимагає 3.3В, тому живлення цього модуля проведено від плати Arduino. Також, для передачі даних необхідно підключити Tx та Rx ніжки для передачі та зчитування даних відповідно – їх підключено до ніжок 1 та 0 від Arduino Uno.

Модуль NRF24L01 вимагає напругу 3.3В, тому його теж підключено до живлення від плати Arduino. Також, для коректної роботи даного модуля необхідно підключити CE, CSN, SCK, MOSI, MISO (протокол SPI [16]), що підключені до ніжок 9, 10, 13, 11, 12 плати Arduino Uno.

Реалізацію логіки буде виконано за допомогою мови Arduino. Конфігурація компонентів хабу відбувається у методі void setup(), який викликається найпершим при старті роботи.

Для роботи із Bluetooth, виконується метод Serial.begin(96000), що дозволяє писати та читати дані, та виконується radio.begin() для роботи із радіомодулем, де radio це об'єкт типу RF24.

					IT51.240БАК.002 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

Основний код логіки знаходиться у методі void loop() який виконується ітеративно під час роботи програми. Читання та надсилання даних по радіоканалу відбувається за допомогою виклику методів radio.write та radio.read відповідно.

Під час реалізації було використано сторонні файли вихідного коду:

- SPI.h;
- nRF24L01.h;
- RF24.h

та відповідні їх реалізації у .cpp файлах.

Проект складається з п'яти файлів:

- main.ino;
- liviness_probe.ino;
- bluetooth_communication.ino;
- objects_communication.ino;
- command_dispatching.ino.

У файлі main.ino базову логіку роботи хабу. Тут проводиться підключення та налаштування усіх компонентів системи, проводиться відповідна конфігурація ніжок на платі Arduino Uno. Саме у даному файлі наведено основні методи логіки роботи: void setup() та void loop().

У файлі liviness_probe.ino реалізовано алгоритм перевірки статусу компонентів системи. У ньому наявні методи, які, безпосередньо, надсилають сигнал на перевірку статусу компонента, методи, які контролюють частоту та необхідність надсилання запитів.

Методи у файлі bluetooth_communication.ino надають можливість оброблювати команди, надіслані користувачем через застосунок Bluetooth каналом, та надають можливість надіслати відповідь користувачу. Робота з Bluetooth модулем виконується через Tx та Rx ніжки плати Arduino Uno.

У файлі `objects_communication.ino` реалізовано логіку передачі та зчитування даних рухомих об'єктів. Тут реалізована робота із списком підключених об'єктів, реалізовано роботу із NRF24L01 модулем радіопередачі.

Методи, надані у файлі `command_dispatching.ino` виконують допоміжну функцію правильної обробки запитів від користувача, правильного надсилання відповідей, формування даних у правильному форматі для комунікації із системою рухомих об'єктів.

3.3 Висновки до розділу

У даному розділі розглянуто основні етапи реалізації компонентів програмно-апаратного комплексу управління системою рухомих об'єктів, а саме клієнтського застосунку та хабу. Описані ключові елементи реалізації клієнтського застосунку мовою програмування Java із використанням технології Android, показано приклади користувацького інтерфейсу. Для опису реалізації хабу надано схему підключення компонентів, описані основні бібліотеки та підходи для реалізації комунікації хабу через радіоканал та Bluetooth.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

4 МОЖЛИВОСТІ ПОДАЛЬШОГО РОЗВИТКУ ТА РОЗШИРЕННЯ

4.1 Збільшення ширини покриття хабу

У проєкті подано структуру, у якій використано один хаб із радіомодулем для комунікації із рухомими об'єктами, що зумовлює те, що ширина покриття, тобто, максимальна відстань дії, програмно-апаратного комплексу управління системою рухомих об'єктів є обмеженою. Шириною покриття комплексу є максимальна відстань дії радіомодуля хабу.

Для використання розробленого комплексу у невеликих масштабах такої ширини покриття може бути достатньо, проте, можуть виникнути задачі, які потребують більшої у декілька разів відстані дії.

Дану проблему можна вирішити декількома способами:

- заміна радіомодуля на той, що має більшу відстань дії;
- побудувати мережу хабів.

Перший варіант, а саме заміна радіомодуля на той, що має більшу відстань дії, накладає деякі обмеження:

- радіомодуль великого радіусу дії може коштувати у десятки разів дорожче, ніж, наприклад, радіомодуль з відстанню дії до кілометра.
- такий варіант вимагає встановлення однакових радіомодулів, як на хабі, так і на рухомому об'єкті;
- відстань дії будь-якого радіомодуля є обмеженою.

Виходячи з даних обмежень очевидно, що встановлення радіомодуля з більшим радіусом дії вирішує проблему лише частково, наприклад, може збільшити відстань дії у декілька разів. Проте, такий варіант не є гнучким та може бути занадто дорогим та не рентабельним.

Другий варіант, а саме, побудувати мережу хабів, полягає у тому щоб, створити декілька однотипних хабів, які будуть комунікувати між собою, передаючи інформацію щодо підключених об'єктів, надавати інтерфейс для передачі команд тощо.

					IT51.240БАК.002 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Такий варіант має багато переваг над варіантом із заміною радіомодуля на той, що має більшу відстань дії, а саме:

- не вимагає зміни рухомого об'єкта;
- мережа хабів не обмежена їх кількістю і може покривати дуже велику територію;
- фінансові витрати необхідні лише для побудови однотипних хабів;

Схематично програмно-апаратний комплекс управління системою рухомих об'єктів можна побачити на рисунку 4.1

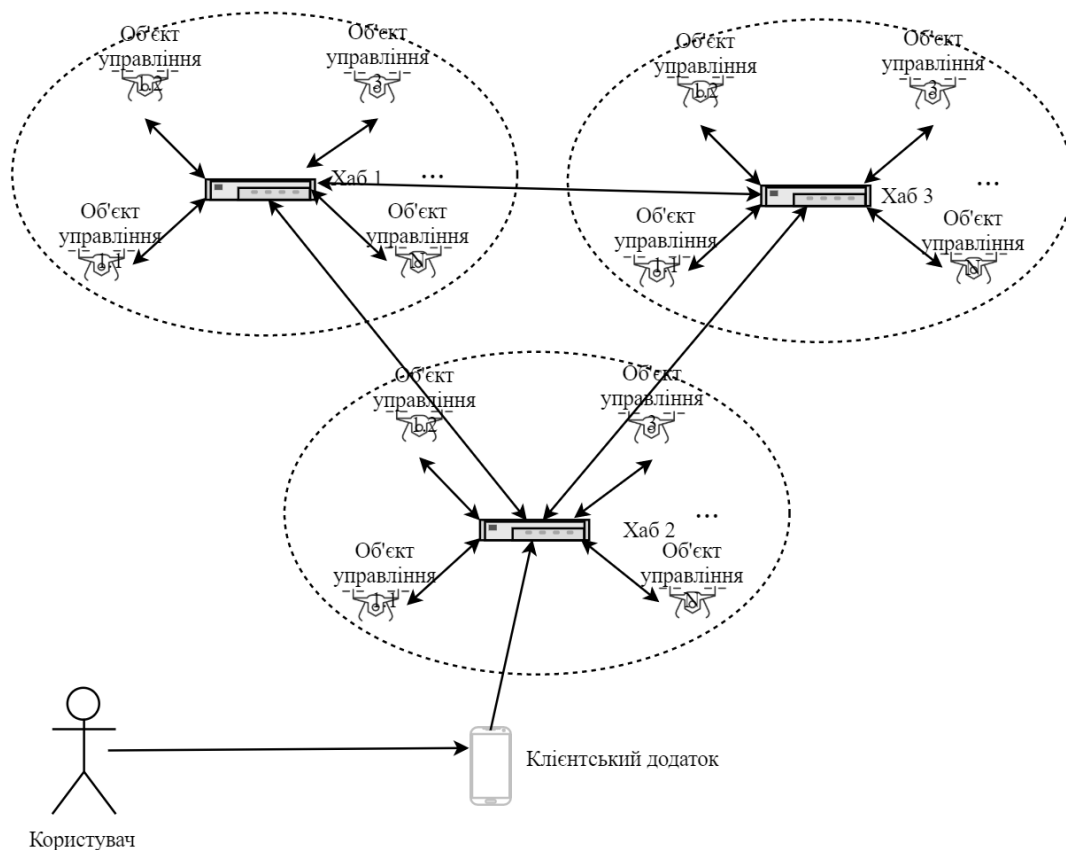


Рисунок 4.1 – Схема комплексу із використанням мережі хабів

Такі мережі хабів можна будувати за багатьма топологіями: шина, кільце, дерево та їх будь-які комбінації, що робить такий варіант дуже гнучким.

З точки зору кінцевого користувача, використання такого варіанту це – збільшена кількість видимих підключених об'єктів, збільшена кількість хабів, проте метод управління залишається незмінним. Зі сторони рухомих об'єктів

немає ніяких змін – лише хаб реалізовує додаткову логіку підтримку зв'язку із мережею та передачі відповідної інформації.

4.2 Реалізація протоколу зв'язку для вирішення прикладних задач

У проекті розглянуто базовий протокол зв'язку, який включає в себе лише необхідну та обов'язкову інформацію для взаємодії компонентів між собою. Такий протокол дає можливість лише базового управління об'єктами, та не дозволяє виконувати інші прикладні задачі.

Протокол спроектовано так, що його можна доповнити та модифікувати беручи за основу базовий. Таку модифікацію необхідно проводити як на хабі, так і на рухомих об'єктах.

Перш за все, треба змінити версію протоколу передачі даних. Надалі, необхідно додати додаткові команди, описати реакції на ці команди рухомими об'єктами, описати очікувані відповіді хабу.

Наприклад, для виконання задач по дослідженню території, хаб має передавати чотири точки, які описують площу необхідної для вивчення території. Надалі, рухомий об'єкт має прийняти ці дані відповідно до протоколу та проаналізувати усю територію описаної чотирма точками. Також, рухомий об'єкт має передавати відповідь хабу у описаному форматі – для дослідження місцевості це масив даних, який описує зображення такої місцевості.

Модифікацій базового протоколу може бути безліч, для вирішення багатьох прикладних задач, передачі даних різного формату та змісту.

4.3 Автоматизація прийнять рішень хабом

Розроблений програмно-апаратний комплекс управління системою рухомих об'єктів дозволяє користувачеві повністю керувати рухомими об'єктами через єдиний інтерфейс: дозволяє передавати об'єктам, переглядати

					IT51.240БАК.002 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

необхідну інформацію тощо. Проте, виключає автоматизацію прийняття рішень хабом та змушує оператора слідкувати за усіма компонентами одночасно.

Хоча така проблема не є критичною, вона сповільнює управління рухомими об'єктами при збільшенні їх кількості, сповільнює прийняття рішень та підвищує шанс на помилку.

Якщо задача до системи рухомих об'єктів є чітко поставленою та математично описаною, процес управління можна автоматизувати.

Хаб володіє усією необхідною інформацією щодо підключених до нього об'єктів, їх положення, статусів тощо та може бути запрограмований на вирішення поставлених задач на базі наявних у нього даних. Задля того, щоб прискорити обчислення та керування рухомими об'єктами, буде виділено окремий модуль прийняття рішень, так як показано на рисунку 5.2.

Блок прийняття рішень абстрагує алгоритми, за допомогою яких створюються нові команди рухомим об'єктам при початку роботи, та, безпосередньо, під час їх роботи. Такий модуль може мати в собі багато різних алгоритмів, методів та механізмів прийняття рішень, серед яких:

- математично описані алгоритми, наприклад, за допомогою формул, систем рівнянь тощо;
- алгоритми, описані за допомогою кінцевого автомату;
- алгоритми машинного навчання для прогнозування тих чи інших метрик рухомих об'єктів [17];
- штучна нейронна мережа для вирішення задач розпізнання образів, кластеризації тощо [18].

Підхід із використанням модуля прийняття рішень дозволить абстрагувати оператора від керування кожним об'єктом окремо, та дозволить поставити чітко описану задачу на хаб, яку він буде виконувати за допомогою відповідних алгоритмів прийняття рішення.

					<i>IT51.240BAK.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

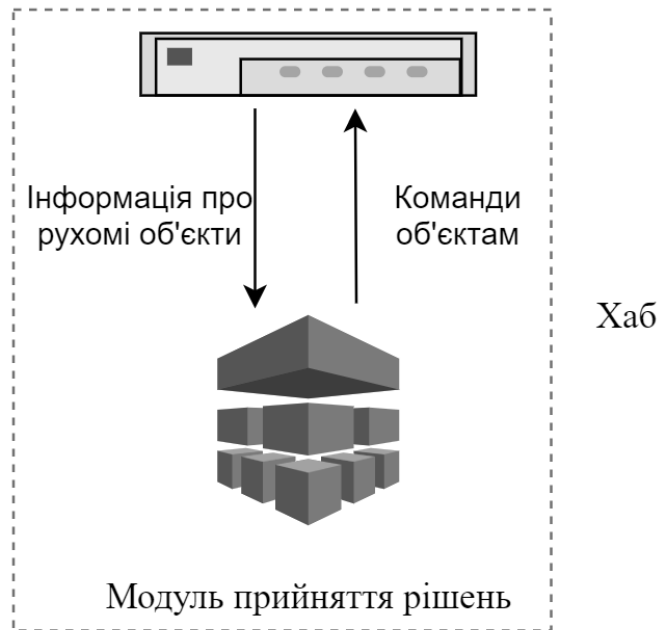


Рисунок 5.2 – Структура хабу із використанням модуля прийняття рішень

4.4 Висновки до розділу

У розділі описано основні можливості подальшого розвитку та розширення програмно-апаратного комплексу, такі як: збільшення ширини покриття хабу, що дозволяє збільшувати відстань керування рухомими об'єктами у декілька раз; розширення протоколу зв'язку, що дозволяє виконувати прикладні задачі; та автоматизація прийняття рішень хабом, що зменшую навантаження на оператора під час керування складними системами, підвищує точність виконання та суттєво зменшує шанс виникнення помилки.

					IT51.240БАК.002 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У даній роботі спроектовано та реалізовано програмно-апаратний комплекс управління системою рухомих об'єктів. Такий комплекс є вирішенням проблеми керування декількома об'єктами одночасно, та проблеми складності керування ними. Розроблений комплекс, є актуальним для підприємств, для виконання наукової та дослідницької діяльності.

Показано основні етапи вибору структури комплексу, розробки протоколу зв'язку, етапи проектування структурних компонентів.

Як оптимальну рішення обрано структуру програмно-апаратного комплексу управління системою рухомих об'єктів з двома основними компонентами: хаб, який інкапсулює логіку управління та координації рухомими об'єктами, та користувацький застосунок, головною метою якого є полегшити взаємодію користувача із вищезгаданим хабом. Головним механізмом зв'язку між компонентами обрано радіоканал.

Хаб, який є центральним модулем комплексу реалізовано на базі плати Arduino Uno, радіомодуля NRF24L01 та Bluetooth модуля. Основна програмна частина виконана з використанням мови програмування C. Клієнтський додаток, реалізовано на платформі Android за допомогою використання мови програмування Java.

					IT51.240БАК.002 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Stellman, A. Applied Software Project Management / A. Stellman, J.Greene. – М.: O'Reilly Media, 2008. – 336 с.
2. Petter L. Quantification and Traceability of Requirements / L. Petter. – М.: O'Reilly Media, 2005. – 115 с.
3. Сайт компанії dji [Електронний ресурс]. – Режим доступу: <https://www.dji.com/mavic>
4. Сайт Ajax Systems [Електронний ресурс]. – Режим доступу: <https://ajax.systems>
5. Олифер В. Компьютерные сети. Принципы, технологии, протоколы. / В. Олифер, Н. Олифер. – М.: Питер, 2012. – 944 с.
6. Сайт технології Bluetooth [Електронний ресурс]. – Режим доступу: <http://www.bluetooth.com>
7. Сайт технології Wi-Fi [Електронний ресурс]. – Режим доступу: <https://www.wi-fi.org/>
8. Сайт формату даних JSON [Електронний ресурс]. – Режим доступу: <https://www.json.org/>
9. Сайт Raspberry Pi [Електронний ресурс]. – Режим доступу: <https://www.raspberrypi.org/>
10. Сайт Arduino [Електронний ресурс]. – Режим доступу: <https://www.arduino.cc/>
11. Сайт, де зазначено радіомодуль NRF24L01 [Електронний ресурс]. – Режим доступу: <https://arduino.ua/prod231-radiomodyl-nrf24l01-2-4-ggc>
12. Сайт, де зазначено Bluetooth HC-05 [Електронний ресурс]. – Режим доступу: <https://arduino.ua/prod1039-bluetooth-hc-05>
13. Матеріали офіційного сайту мови програмування Java [Електронний ресурс]. – Режим доступу: <https://www.java.com>.

					IT51.240БАК.002 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

14. Neuburg M. Programming iOS 12: Dive Deep into Views, View Controllers, and Frameworks / M. Neuburg. – М.: O'Reilly Media, 2018. – 1176 с.
15. Буч Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон. – М.: ДМК Пресс, 2006. – 496 с.
16. Лапин А. Интерфейсы. Выбор и реализация / А. Лапин. – М.: Техносфера, 2005. – 168 с.
17. Alpaydin E. Introduction to Machine Learning / E. Alpaydin. – М.: London: The MIT Press, 2017. – 400 с.
18. Brian D. Pattern Recognition and Neural Networks / D. Brian. – М.: Cambridge University Press, 2007. – 403 с.

					<i>IT51.240БАК.002 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

ДОДАТОК А

Макет підключення компонентів хабу

