

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Радіотехнічний факультет
Кафедра радіотехнічних систем**

До захисту допущено:

Завідувач кафедри

_____ Сергій ЖУК

«__» червня 2023 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньою програмою «Радіотехнічні інформаційні технології»

спеціальності 172 «Телекомунікації та радіотехніка»

**на тему: « Програмно апаратна система відображення інформації радіо-
локаційної системи »**

Виконав (-ла):

студент (-ка) IV курсу, групи РТ-91

Мосеев Сергій Віталійович _____

Керівник:

Доцент, к.т.н. Катін Павло Юрійович _____

Рецензент:

Доцент, к.т.н. Мартинюк Сергій Євстафійович _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент (-ка) _____

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Радіотехнічний факультет
Кафедра радіотехнічних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітня програма «Радіотехнічні інформаційні технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій ЖУК

«__» квітня 2023 р.

ЗАВДАННЯ

на дипломну роботу студенту

Прізвище, ім'я, по батькові

1. Тема роботи « Програмно апаратна система відображення інформації радіолокаційної системи », керівник роботи доцент, к.т.н. Катін Павло Юрійович затверджені наказом по університету від «30» травня 2023 р. № 2061-с

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи Максимальне значення дальності цифрової радіолокаційної системи, роздільна здатність цифрової радіолокаційної системи по дальності, роздільна здатність цифрової радіолокаційної системи по азимуту, ширина діаграми спрямованості.

4. Зміст роботи Провести аналіз завдання, порівняти характеристики міні комп'ютерів, розрахувати характеристики радіолокаційної системи, описати імітатор для програмно апаратної системи відображення ,розробити схеми

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) Схема структурна програми, схема електрична принципова, схема структурна сублока відображення, інші матеріали(слайди), які необхідні для презентації результатів роботи.

7. Дата видачі завдання 17 квітня 2023р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Узгодження та ознайомлення ТЗ	17.04.23-20.04.23	Виконано
2	Ознайомлення з літературними джерелам	21.04.23-30.04.23	Виконано
3	Аналіз ТЗ	01.05.23-11.05.23	Виконано
4	Проведення аналізу міні комп'ютерів	12.05.23-17.05.23	Виконано
5	Розрахунок параметрів РЛС	17.05.23-24.05.23	Виконано
6	Ознайомлення з програмною частиною	24.05.23-29.05.23	Виконано
7	Розробка схем	30.05.23-06.06.23	Виконано
8	Підготовка до презентації	07.06.23-19.06.23	Виконано

Студент

Сергій МОСЄЄВ

Керівник

Павло КАТІН

АНОТАЦІЯ

Дипломна робота складається з пояснювальної записки обсягом 55 сторінок, містить 8 рисунків, 1 таблиці та 15 посилань

Дана робота присвячена дослідженню систем відображення інформації для радіолокаційних систем (РЛС). У роботі проведений загальний аналіз трьох популярних міні-комп'ютерів: Orange Pi, Raspberry Pi та Odroid-N2L. Проведено порівняльний аналіз їх характеристик і можливостей.

Далі розглянуті основні аспекти розрахунку параметрів РЛС. Подано опис РЛС і проведено розрахунок загальних параметрів. Аналізується сигнал-шум і розраховується мінімальна потужність приймача. Також розглядаються параметри антени і проводиться їх розрахунок.

Наступний розділ присвячений програмній частині системи відображення інформації РЛС. Розглядається загальна структура бібліотек QT для графічних інтерфейсів і склад бібліотек для реалізації графічного інтерфейсу РЛС. Представлена структурна схема програмного рішення, а також переваги та недоліки використання QT для його реалізації.

В наступному розділі розглядається апаратна частина системи відображення інформації РЛС. Описується розробка електричної схеми приймального пристрою та розробка структурної та принципової схеми

Ключові слова: Радіолокаційна система, QT, міні комп'ютер, програмно апаратна система, система відображення.

ANNOTATION

The thesis consists of an explanatory note of 55 pages, contains 8 figures, 1 table and 15 references

This work is devoted to the study of information display systems for radar systems. The work includes a general analysis of three popular mini-computers: Orange Pi, Raspberry Pi and Odroid-N2L. A comparative analysis of their characteristics and capabilities was carried out.

Next, the main aspects of the calculation of radar parameters are considered. A description of the radar is given and the calculation of the general parameters is carried out. The signal-noise is analyzed and the minimum power of the receiver is calculated. Antenna parameters are also considered and their calculation is carried out.

The next section is devoted to the software part of the radar information display system. The general structure of QT libraries for graphical interfaces and the composition of libraries for implementing the graphical interface of the radar are considered. The structural scheme of the software solution is presented, as well as the advantages and disadvantages of using QT for its implementation.

In the next section, the hardware part of the radar information display system is considered. The development of the electrical circuit of the receiving device and the development of the structural and principle diagram are described

Key words: Radar system, QT, mini computer, hardware and software system, display system.

ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломної роботи

на тему: Програмно апаратна система відображення інформації радіолокаційної системи

Київ — 2023 року

ЗМІСТ

Перелік скорочень.....	9
Вступ.....	10
1 Загальний аналіз міні комп'ютерів	11
1.1 Orange pi.....	11
1.2 Raspberry pi	14
1.3 Odroid-n2l	16
1.4 Порівняльний аналіз міні комп'ютерів	18
2 Розрахунок параметрів рлс	21
2.1 Опис РЛС	22
2.2 Розрахунок загальних параметрів	24
2.3 Аналіз сигнал шум	25
2.4 Розрахунок мінімальної потужності приймача.....	27
2.5 Розрахунок параметрів антени	28
3 ПРОГРАМНА ЧАСТИНА СИСТЕМИ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ рлс	32
3.1 Загальна структура бібліотек QT для графічних інтерфейсів.....	32
3.2 Склад бібліотек для реалізації графічного інтерфейсу РЛС	37
3.3 Структурна схема програмного рішення.....	40
3.4 Переваги QT для реалізації програмного рішення	44
3.5 Недоліки QT для реалізації програмного рішення	45
4 АПАРАТНА ЧАСТИНА СИСТЕМИ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ рлс	47
4.1 Розробка електричної схеми приймального пристрою радіолокаційної системи.....	47

4.2 Розробка структурної схеми сублока відображення	49
4.3 Розробка принципової схеми	51
Висновки	53
Перелік джерел посилань	54

ПЕРЕЛІК СКОРОЧЕНЬ

РЛС – Радіолокаційна станція

GUI – Графічний інтерфейс користувача

API – Прикладний програмний інтерфейс

ОС – Операційна система

ВСТУП

Радіолокаційній станції відіграють надзвичайну роль у сфері спостереження та безпеки. РЛС виявляють і відстежують переміщення об'єктів у повітрі, морі. Для ефективного функціонування треба мати необхідний потужний програмно-апаратний комплекс відображення. У процесі обробки та відображення даних ключову роль відіграє комплекс виведення інформації. Основними завданнями є: збір, аналіз та візуалізація інформації про виявленні об'єкти. Він допомагає зрозуміти оператору отриману інформацію радарним датчиком, що надає йому швидко реагувати на ситуації.

Система відображення складається з програмних і апаратних компонентів. Апаратна частина включає в себе екрани та інші пристрої що забезпечують якісне відображення інформації. Програмна частина містить алгоритми обробки сигналів, місць розташування об'єктів, програми візуалізацій необхідні для ефективного аналізу та сприйняття даних.

Важливість програмного та апаратного забезпечення полягає в тому щоб допомогти оператору швидко аналізувати данні отримані від РЛС датчиків. Завдяки цій системі оператори можуть визначати напрямок та швидкість цілі та дізнатись що це за ціль. Оскільки радіолокаційні системи використовують від систем управління повітряним рухом до управління морським, його можна адаптувати під потреби і вимоги кожного із спеціалістів, такі як: інтеграція з іншими системами, обмін даними в реальному часі. Однією з переваг програмно апаратної частини системи формувань зображення є здатність адаптуватися до мінливих умов. З плином часу вдосконалення програмно-апаратної частини дозволило вдосконалити точність швидкість та надійність обробки даних. Розвиток і вдосконалення тривають і надалі, що означає що можна очікувати що надалі інформацію можна буде отримати ще швидше і надійніше.

1 ЗАГАЛЬНИЙ АНАЛІЗ МІНІ КОМП'ЮТЕРІВ

Міні комп'ютер - це компактні обчислювальні пристрої, які мають функціональність схожі зі звичайними персональними комп'ютерами, але з меншими розмірами та потужністю. Вони ідеально забезпечують компактність та портативність, що робить їх ідеальним рішенням де вимагається не великий розмір та вага комп'ютерів. Хоча міні комп'ютери мають маленькі габарити це не означає що вони мають обмежені можливості. Деякі міні ПК мають значну обчислювальну потужність якої достатньо для вирішення складних рішень. Більш прості моделі можуть впоратись із повсякденними для нас справами, такими як веб-серфінг , робота з офісними програмами та навіть відтворення мультимедійного контенту. Міні комп'ютери підтримують різні операційні системи, такі як Windows та Linux. Це дає розробникам велику гнучкість у виборі програмного забезпечення та адаптації до своїх потреб. Сховище є обмеженим аспектом у міні комп'ютерів, однак їх можливо розширити за допомогою зовнішніх пристроїв або внутрішніх роз'ємів. Загалом міні комп'ютери стають все більш популярними для вибору користувачів, які для своїх задач потребують компактність і портативність. Вони застосовують у різних галузях від освіти та різних галузях промисловості до домашнього використання. Міні комп'ютери постійно вдосконалюються завдяки стрімкому розвитку технологій , та пропонують користувачам більш потужні опції. Гарним прикладом слугує Orange pi, Raspberry pi та Ondroid-n2l.

1.1 Orange pi

Orange pi – це серія одноядерних комп'ютерів виробництва Shenzhen Xunlong Software Co. Міні ПК їх виробництва пропонують зручний і доступний спосіб досягти повної обчислювальної потужності в компактному дизайні.[4] Завдяки своїм функціям та продуктивності Orange Pi можуть задовольнити потреби широкого кола користувачів, від початківців до досвідчених розробників.

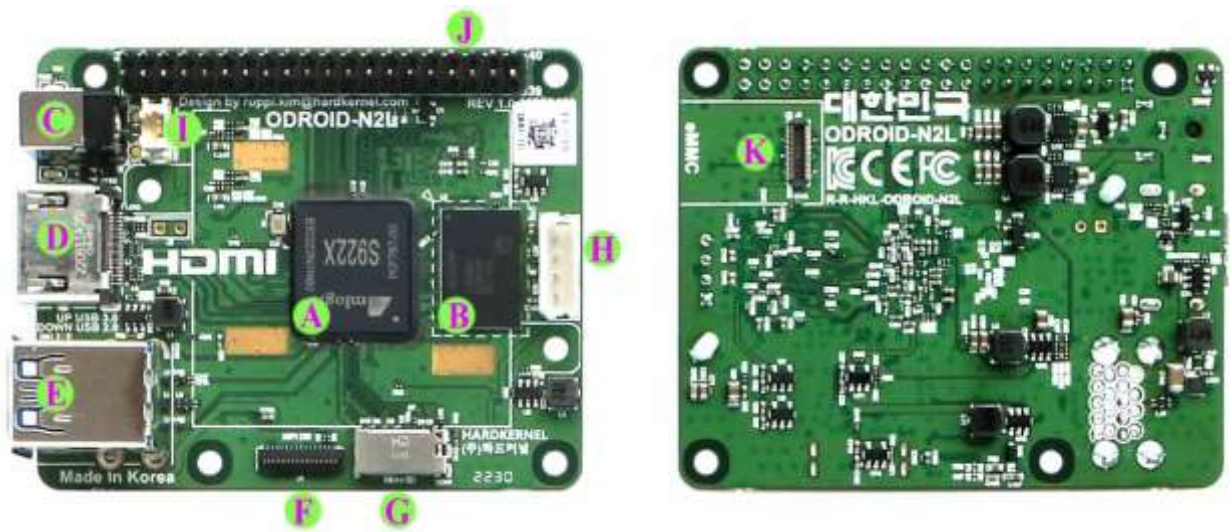


Рис. 1.1 – міні комп'ютер Orange pi

Orange Pi доступний в моделях з різними технічними характеристиками. Кожна модель має різний процесор, кількість ядер, обсяг оперативної пам'яті та графічні можливості. Це дозволяє користувачам вибирати платформу яка найкраще відповідає їхнім потребам. Для прикладу модель із процесором Allwinner H3 можна використовувати для побутових цілей, тобто відтворення мультимедіа або ж веб-серфінг. Моделі з процесором Rockchip RK3399 можна використовувати для розробки відеоігор.[2]

Orange pi має ряд портів і роз'ємів, які дозволяють підключати периферійні пристрої для розширення його функціональності. Порт USB слугує для підключення периферійних пристроїв, порт HDMI для підключення до монітора або телевізора, порт Ethernet для підключення до мережі, аудіо роз'єми для виводу та вводу звуку, підключення карти пам'яті для розширення функціональності. Це робить Orange pi універсальною платформою для широкого спектру проектів і додатків.

Однією з найважливіших особливостей Orange pi є його процесор. Залежно від конкретної моделі, Orange pi може використовувати різні процесори. Вони забезпечують достатню продуктивність для різноманітних завдань, від повсякденного використання до більш складних завдань розробки. Моделі Orange pi можуть бути оснащені чотирьох або восьми ядерними процесорами з

різною частотою. Це дозволяє їм забезпечувати швидкість обробки, необхідну для багатьох завдань, включаючи відеоігри, обробки відео, веб сервери та інші вимогливі програми. Деякі моделі підтримують апаратне прискорення відео та 3D графіки, що робить їх ідеальними для розробки мультимедійних проектів.

Окрім процесора Orange pi підтримує і оперативну пам'ять. Залежно від моделі можна встановити різний об'єм оперативної пам'яті від 512 мегабайт до 4 гігабайт. Більший обсяг оперативної пам'яті дозволяє ефективніше виконувати ресурсномісткі завдання і працювати з більшими та складними додатками.

Orange pi підтримує різні способи зберігання даних, таких як внутрішню флеш пам'ять або ж порти для sd карт. Це дозволяє користувачам обирати найкращий спосіб зберігання та доступу до своїх даних.

Orange pi пропонує різні способи підключення та розширення функціональності. Однією з багатьох функцій є порт USB. За допомогою цього порту можна підключити різні зовнішні пристрої, такі як клавіатура, миша або ж флеш накопичувач. Для підключення монітору або телевізору для перегляду відео та графіки слугує порт HDMI. Ця функція буде особливо корисною для проектів як потребують відображення відеосигналу. Для збору даних або віддаленого керування в Orange pi є порт Ethernet. Деякі моделі підтримують бездротове підключення до мереж за допомогою WI-FI або Bluetooth. Також цей міні комп'ютер має роз'єм для підключення зовнішніх аудіо пристроїв, таких як навушники і мікрофони.

Однією з найважливіших особливостей Orange pi є операційна система. Orange pi підтримує широкий спектр операційних систем таких як Linux, Debian, а також Android. Цей список дозволяє вибрати ОС яка найбільше підходить для поставлених завдань. Дистрибутиви Linux, Debian пропонують стабільну і надійну роботу, а також широкий спектр програмного забезпечення і інструментів для розробки. Окрім вибору операційної системи Orange pi підтримує різні середовища розробки та мови програмування. Використовуючи поширені мови програмування такі як Python, c/c++ і java. До того ж

Orange pi надає доступ до різних бібліотек та інструментів які полегшують розробку та реалізацію проектів. Підключення до хмарних сервісів та платформ інтернету дозволяє легко інтегрувати Orange pi в розподілені системи.[3]

Orange Pi-це потужна та гнучка платформа, яка пропонує безліч можливостей для розробки та реалізації найрізноманітніших проектів. Компактний розмір, низька вартість і широкі можливості підключення роблять її привабливим варіантом для ентузіастів, розробників і любителів.

1.2 Raspberry pi

Raspberry pi - один найпопулярніших одно платних комп'ютерів на ринку, випущений в 2012 році компанією Raspberry pi Foundation. Швидко набирає популярність серед розробників та навчальних закладів.[7]



Рис. 1.2 - Міні комп'ютер Raspberry pi

Raspberry pi пропонує гнучкість і потужність в компактному форматі і відкриває широкий спектр можливостей для реалізації проектів. Головною перевагою Raspberry pi є доступність у порівнянні зі звичайними комп'ютерами із економічної точки зору. Це робить його привабливим рішенням для проектів із малим бюджетом або ж для студентів які почали цікавитись вбудованими системами.[6] Залежно від моделі Raspberry pi може виконувати різні задачі. Як і інші міні комп'ютери Raspberry pi підтримує різні операційні системи такі

як Linux і Windows. Вибір операційної системи дозволить користувачам обрати ОС під їхні потреби.[5]

Raspberry Pi знаходить широке застосування в сфері робо техніки. Його можна використовувати для створення різних типів роботів, від простих мобільних роботів до складних систем автоматизації та штучного інтелекту. Завдяки потужним обчислювальним можливостям, вбудованому підключенню датчиків та управлінню пристроями, Raspberry Pi є чудовим вибором для розробки робото технічних проектів.[6]

У сфері освіти Raspberry Pi також має широке застосування. Raspberry Pi допомагає інтегрувати інформатику та програмування у шкільну програму, створюючи доступну та просту у використанні навчальну платформу. Вона дозволяє учням та студентам експериментувати, розробляти проекти та вивчати основи програмування.

Raspberry Pi дуже популярний у сфері інтернет серфінгу. Його можна використовувати для збору та аналізу даних, моніторингу та управління підключеними пристроями. Raspberry Pi широко використовується для домашніх розваг. На ньому можна створювати ігрові приставки, системи домашньої автоматизації, або ж медіа центр. Завдяки доступності та гнучким можливостям Raspberry Pi дає змогу займатися творчістю та експериментами просто вдома. Компактний розмір та низька споживча потужність та гнучкість робочої системи роблять його ідеальним вибором для вбудованих проектів, де потрібно забезпечити комп'ютерну функціональність в обмеженому просторі.

Вбудовані системи на базі Raspberry Pi використовуються у таких галузях як автоматизація промислових процесів та керування системами безпеки. Застосування Raspberry Pi дозволяє розширювати розробку прототипів та виробництва власних пристроїв. Багато компаній вже використовують як основу для створення своїх власних промислових пристроїв цей міні комп'ютер. Це дозволяє прискорити процес розробки, знизити витрати та створити ефективні рішення для конкретних промислових потреб. Raspberry Pi має значний вплив у сфері досліджень та інновацій.[7] Його доступність та широкі можливості

роблять його привабливим для винахідників та дослідників. Багато дослідників використовують Raspberry pi для проведення наукових експериментів та досліджень у різних галузях. Його використовують для аналізу результатів та виконання обчислень. Відкрите програмне забезпечення які супроводжують Raspberry pi дозволяє розробникам експериментувати та створювати нові проекти та ділитися своїми досягненнями. Це сприяє розвитку самої системи та поширенню знань.

Raspberry pi використовують для запуску стартапів.[7] Завдяки доступності він дозволяє швидко перевірити ідею та створити прототип ідеї. Це допомагає розвиватися маленьким компаніям та індивідуальним розробникам які хочуть вивести свій продукт на ринок. Також одним із ключових факторів успіху Raspberry pi є широка та активна спільнота користувачів цього міні комп'ютера.

1.3 Odroid-n2l

Одним із найсучасніших міні комп'ютерів є Ondroid-n2l який розробляється компанією Hardkernel. Він поєднує в собі потужну продуктивність , розширені можливості та гнучкість використання ,що робить його привабливим для вибором для широкого використання.[10]



Рис. 1.3 – Міні комп'ютер Odroid-n2l

Однією з його головних переваг є його процесор. Android-n2l працює на базі потужного процесора Amlogic S922x з восьми ядерною архітектурою, що працює на частоті до 1,8 ГГц. Це дозволяє забезпечувати високу продуктивність та швидкість обробки даних, що дозволяє справитися з вимогливими завданнями.[8]

Android-n2l оснащений двома гігабайтами оперативної пам'яті для швидкої передачі даних та ефективності, що дозволяє запускати декілька додатків одночасно без шкоди для продуктивності.

Як і будь який інший міні комп'ютер він має розширені можливості підключення. Він має порт HDMI для підключення до монітору або телевізора, порт Ethernet слугує для високошвидкісного підключення до інтернету. Крім того в Android-n2l підтримує бездротове з'єднання Wi-Fi. Для підключення зовнішніх пристроїв в Android-n2l є порти USB 2.0 і більш швидкий USB 3.0. Також він підтримує карти пам'яті micro SD і eMMC для зберігання даних.

Однією із сильних сторін Android-n2l є його сумісність із різними апаратними розширеннями і модулями. Android-n2l можна використовувати для широкого спектру застосувань. Завдяки своїй потужності та гнучкості Android-n2l може ефективно виконувати завдання із графічним моделюванням, машинним навчанням.[9] Його застосовують для домашніх серверів, медіацентрів. Він також може бути використаний для розробки послуг де надійна і потужна апаратна платформа має важливе значення.

Компанія Hardkernel регулярно надає оновлення програмного забезпечення, нові версії операційної системи та патчі безпеки, що дозволяє користувачам вдосконалювати свої пристрої та отримувати оновлення. Android-n2l також підтримує розробку програмного забезпечення та надає документацію, щоб допомогти розробникам зрозуміти апаратну архітектуру та використовувати її у своїх проєктах.[8] Це означає що кожен користувач може знайти відповіді на свої питання та розробляти проєкти разом з іншими учасниками та отримувати підтримку для свої проблем і недоліків. Це дозволило Android-n2l зібрати широку спільноту шанувальників.

1.4 Порівняльний аналіз міні комп'ютерів

Для проведення аналізу міні комп'ютерів ми спочатку будемо дивитися на їхню потужність. Orange pi і Raspberry pi пропонують різні моделі з різними процесорами та об'ємами пам'яті. Загалом вони різняться за рівнем продуктивності проте загалом вони пропонують досить прийнятну продуктивність для багатьох завдань. Залежно від конкретної моделі Orange pi може мати відносно низько потужні, одно або багатоядерні процесори з частотою від одного до двох ГГц. Об'єм оперативної пам'яті також може варіюватися від п'ятсот двадцяти мегабайт до чотирьох гігабайт.

Raspberry pi відомий своєю продуктивністю та хорошою сумісністю із різними операційними системами та програмним забезпеченням. Як і Orange pi, Raspberry pi може мати одно або багатоядерний процесор з частотою роботи від одного до півтора ГГц. Об'єм оперативної пам'яті може бути від одного до восьми гігабайт.

Серед цих міні комп'ютерів Odroid-n2l має потужний шести ядерний процесор з архітектурою ARM Cortex-a73. Він особливо привабливий для завдань які потребують більшої обчислювальної потужності. Odroid-n2 може бути використаний як і для важких завдань так і для відео обробки і відео гри. Процесор працює на частоті до трьох ГГц що забезпечує швидке виконання завдань. Об'єм оперативної пам'яті становить від двох до чотирьох гігабайт, що дозволяє працювати з великими за обсягами даними або ж вимогливими програмами.

Для зовнішнього підключення ці міні комп'ютери мають достатньо з'єднань та портів для підключення зовнішніх пристроїв. Ми можемо розширювати їх можливості підключивши USB пристрої для підключення монітора, клавіатури, датчики. Бездротові можливості такі як WI-FI та Bluetooth дозволяють зв'язувати міні комп'ютери без дротів.

Загалом всі три міні комп'ютери мають велику спільноту користувачів яка надає підтримку та допомогу. Вони постійно оновлюють свої програмне

забезпечення, що забезпечує виправлення помилок покращення та нові функції. Також завдяки активній спільноті можливо знайти багато розроблених проєктів, документацію та поради що допоможуть реалізувати нам ідеї та проєкти на цих платформах.

Важливо зазначити що ці міні комп'ютери досить різні по ціні. Якщо Orange pi відомий своєю доступною ціновою політикою. Це означає що ми зможемо знайти потужний міні комп'ютер за помірну ціну. Компанія пропонує різні моделі з різними характеристиками та функціоналом що допомагає обрати оптимальний варіант для наших потреб. Raspberry pi також відомий своєю доступністю, проте зазвичай він має більшу ціну ніж Orange pi. Ondroid-n2i зазвичай дорожчий ніж Raspberry pi. Загалом це пояснюється його високою продуктивністю та потужними характеристиками.

Для порівняння використаємо таблицю.

Табл. 1.1

Характеристика	Orange Pi	Raspberry Pi	Odroid-N2
Процесор	ARM Cortex-A7, A53, A55	Broadcom ARM Cortex-A72	Amlogic S922X (ARM Cortex-A73, A53)
Частота процесора	Залежить від моделі	Залежить від моделі	До 2.4 ГГц
Графічний процесор	Mali	Broadcom VideoCore IV	ARM Mali-G52
Оперативна пам'ять	Залежить від моделі	Залежить від моделі	До 4 ГБ
Вбудована пам'ять	Залежить від моделі	Залежить від моделі	До 128 ГБ

Операційна система	Linux, Android, інші	Linux, Windows 10 IoT	Linux, Android
Роз'єми	USB, HDMI, Ethernet, GPIO	USB, HDMI, Ethernet, GPIO	USB, HDMI, Ethernet, GPIO
Підтримка Wi-Fi	Залежить від моделі	Залежить від моделі	Так
Підтримка Bluetooth	Залежить від моделі	Залежить від моделі	Так
Ціна	Зазвичай нижча	Середня	Зазвичай вища

Якщо брати до уваги текст після детального порівняння міні комп'ютерів, наш вибір пав на Orange pi. Основною причиною цього вибору стала цінова політика, яка вважається однією з найбільш доступних серед аналогічних. При порівнянні технічних складових Orange pi з іншими міні комп'ютерами ми виявили, що характеристики їх основних компонентів були досить схожими. Він виявився економічно вигідним рішенням для нас оскільки відповідає фінансовим можливостям без втрати продуктивності та можливостей.

2 РОЗРАХУНОК ПАРАМЕТРІВ РЛС

Розраховуємо дальність дії радіолокаційної системи. Для цього скористаємося посібником [1]. Для початку введемо максимальну дальність роботи РЛС. Для зручності позначимо його R_{\max} . Він буде дорівнювати 30000 кілометрів. Кут огляду буде дорівнювати 360 градусів. Позначимо його кутом β . Взято цей параметр, щоб РЛС міг відстежувати цілі навколо себе без обмежень в області азимуту. Роздільну здатність по азимуту позначимо $\Delta\beta$. Цей кут буде рівним 1.5 градусів. Цей вибір обумовлений тим щоб РЛС міг розрізнити два об'єкти на відстані 1.5 градуса один від одного в горизонтальній площині. Кут місця буде рівним 14.5. Цей параметр взято із технічного завдання, позначати будемо літерою e . У технічному завданні вказано максимальну швидкість яку, РЛС повинна виявляти і відстежувати. Максимальна швидкість буде рівною шестисот метрів за секунду. Зазначено значення роздільної здатності по швидкості. Він рівний тридцяти метрів за секунду. Цей параметр означає що РЛС може розрізнити два об'єкти з різницею в швидкості. В нашому випадку роздільна здатність по швидкості буде розрізнити об'єкти з різницею в тридцять метрів за секунду. Швидкість будемо позначати V_{\max} , а роздільну здатність ΔV . Коефіцієнт відображення рівня шумів в системі РЛС, буде показувати 3. При скороченні будемо писати Кш. Коефіцієнт використання поверхні антени буде позначатися літерою h . Цей коефіцієнт відображає ефективність використання енергії, яку отримує антена, у виконанні своїх функцій. Він рівний 0.85. Правильне виявлення сигналу буде рівним 0.89. При письмі будемо позначати $P_{\text{вв}}$. Помилкове виявлення сигналу буде рівним $6.5 * 10^{-6}$, позначатимемо $P_{\text{хт}}$. Для використання деяких формул нам будуть потрібні деякі табличні значення. Так температуру будемо визначати в Кельвінах. Позначення на письмі буде T_0 . Температура буде рівною 290 Кельвіна. Швидкість світла це константа яка використовується у різних розрахунках. Вона рівна $3 * 10^8$ метрів за секнду. Позначатимемо цю сталу літерою c . Для обрахунку шумів нам знадобиться стала Больцмана, яка рівна $1.38 * 10^{-23}$. Позначатимемо літерою k .

У технічному завданні зазначено що сигнал який буде використовуватися в РЛС, є М послідовним.

2.1 Опис РЛС

Радіолокаційна станція - це система, що використовує радіохвилі для виявлення, визначення положення, відстеження та ідентифікації об'єктів у просторі. Вона є важливим компонентом цивільної сфери, зокрема в системах раннього попередження про стан погоди.[16]



Рис. 2.1 – Приклад РЛС

В цій сфері РЛС використовується для визначення швидкості вітру, опадів, напрямку хмар, що допомагає прогнозувати погоду і в разі небезпеки швидко на це відреагувати. Ця система виявляє забруднення повітря . Це дозволить контролювати розповсюдження отруйних речовин і сприяє вирішенню екологічних катастроф. Повітряну навігацію взагалі важко уявити без РЛС. Вони встановлені в аеропортах, використовуються для виявлення інших повітряних об'єктів, грозових хмар. Далі ця інформація передається пілотові, що підвищує безпеку при перевезеннях. Система GPS також використовує радіолокаційну систему. Вона дозволяє визначити точне місце знаходження ,що є корисним для навігації. [18]

Працює РЛС за принципом радіосигналів і відбиття. Основна ідея полягає в тому що радар надсилає короткий імпульс електромагнітної енергії та приймає відбиті сигнали від об'єктів у радіусі дії. Існує декілька методів виявлення відбитого сигналу. Частотний метод заснований на використанні частотних переданих безперервних сигналів. В цьому методі частота, що змінюється лінійно від f_1 до f_2 передається протягом періоду. Відбитий сигнал надходитиме лінійно модульованим у часі, що передує теперішньому на час затримки. Тому частота відбитого сигналу, прийнятого радаром буде залежною від пропорційного часу. Фазовий метод заснований на виділенні та аналізі різниці фаз між відбитим та випромінюваним сигналом. Імпульсний радар передає випромінюваний сигнал протягом короткого часу у вигляді коротких імпульсів. Потім переходить у режим прийому та відлуння що відбивається від цілі.[11]

Система РЛС складається з наступних основних компонентів:

Антенa: це пристрій, що формує діаграму спрямованості та приймає сигнали від повітряних(морських) об'єктів . Антени можуть мати різні форми та конфігурації, включаючи параболічні, фазові решітки, активні апертури тощо. Вони забезпечують фокусування сигналу для досягнення високої точності визначення положення об'єктів.

Передавач: це електронне устаткування, яке генерує радіосигнали та передає їх через антену. Ці сигнали відбиваються від об'єктів у просторі та повертаються до РЛС у вигляді ехо.

Приймач: це пристрій, що отримує відбиті сигнали (ехo) від об'єктів. Він аналізує ці сигнали для визначення відстані, напрямку та інших характеристик об'єктів. Приймач також реалізує процес розрізнення між різними об'єктами та шумами, що можуть впливати на сигнал.

Система обробки сигналу: це складний набір алгоритмів та програмного забезпечення, які виконують обробку та аналіз сигналів

Різні типи радарів використовуються в різних сферах. Морські радари встановлюються на кораблях або ж на берегових станціях і слугують для виявлення підводних човнів, кораблів або ж інших об'єктів на поверхні моря.

Для виявлення повітрях цілей використовуються бортові радары. Вони слугують для виявлення та відстеження, ідентифікації літаків, гелікоптерів. На метеорологічних станціях використовуються наземні і спеціальні радары наприклад: метеорадар.

Кожен тип радарів має свої унікальні характеристики та застосування. Ці РЛС можуть мати різні конфігурації та стаціонарні системи і відрізняються активним і пасивним радаром. Пасивні радары використовують вже наявні радіо сигнали, а активні ж самостійно генерують радіосигнал. Спільне ж у кожного із них те, що кожен забезпечує виявлення, ідентифікацію об'єктів для різних цілей.

2.2 Розрахунок загальних параметрів

Радіолокація ґрунтується на деяких фізичних властивостях радіохвиль. Однією із цих властивостей є сталість швидкості та прямолінійність поширення радіохвилі у вільному просторі. Ці параметри дозволять нам дізнатися роздільну здатність по дальності. Для цього необхідно визначити добуток швидкості світла та тривалість імпульсу, та поділити їх добуток на два. Чому саме на два? Тому що величина стиснутого імпульсу проходить відстань рівну подвоєній відстані до цілі. Якщо написати формулою то виходить такий

вираз:

$$\Delta r = \frac{c * t_i}{2}$$

Так як ця величина в нас присутня із технічного завдання, вона нас не сильно цікавить, але із цієї формули ми можемо дістати величину стиснутого імпульсу. Для цього скористаємося елементарними перетвореннями. Якщо перемножити навхрест в нас вийде :

$$2\Delta r = c * t_i$$

Тепер слід поділити обидві частини на швидкість світла. Після цих перетворень в нас виходить ось така формула.

$$t_i = \frac{2 * \Delta r}{c}$$

Тепер слід підставити числові значення.

$$t_i = \frac{2 \cdot 14}{3 \cdot 10^8} = 9.333 \cdot 10^{-8}$$

Ця величина підходить, оскільки дозволить нам розрізнити цілі, якщо вони знаходяться близько одна до одної. Для прикладу у випадку коли цілі будуть розташовані на відстані 150 метрів, ми б не змогли б розрізнити 5 літаків окремо, а замість цього бачили б лише одну велику область. Однак мінімальна відстань яку ми здатні розрізнити дві стоячі цілі, буде більшою за наше значення. Тобто якщо відстань між цілями перевищує наше значення, ми зможемо їх розрізнити.

Після того як ми дізнались величину стислого імпульсу, ми можемо дізнатися ширину спектру сигналу. Вона обернено пропорційна ширині спектру.

$$\Delta f = \frac{1}{t_i} = 1.071 \cdot 10^7$$

Із переліку технічного завдання ми можемо знайти період повторення зондуючого сигналу. Для цього потрібно знати максимальну відстань та швидкість світла. В нас все є щоб дізнатися цю величину.

$$T_{\Pi} = \frac{2 \cdot R_{max}}{c} = 2 \cdot 10^{-4} \text{ с}$$

2.3 Аналіз сигнал шум

Починаємо аналізувати сигнал шум. Відношення сигнал шум це відношення середньої потужності сигналу до середньої потужності шумів.

Сигнал шум визначає ефективність виявлення радіолокаційних сигналів. Ми можемо використовувати формулу, що по своїй суті є апроксимацією кривих виявлення.

$$q = 2 * \left(\frac{\ln\left(\frac{1}{P_{xt}}\right)}{\ln\left(\frac{1}{P_{\text{ВВ}}}\right)} - 1 \right) = 202.983$$

Це необхідне відношення сигнал шум при виявленні сигналу відстані, щоб ми забезпечили правильне виявлення ($P_{\text{ВВ}}$), а хибної тривоги (P_{xt}), нам

треба забезпечувати ось таку величину сигнал шум. Чим далі тим відношення сигнал шум буде гіршим. Це відношення сигнал шум у разях для зручності обрахунків слід перевести цю величину в децибелі. Для цього необхідно десять помножити на логарифм відношення сигнал шум.

$$10 * \log(q) = 23.075 \text{ Дб}$$

Використавши основну формулу радіолокації ми можемо знайти мінімальну потужність приймача.

$$R_{\max 1} = \sqrt[4]{\frac{P_{\text{пер}} * G^2 * \sigma_{\text{ц}} * \lambda^2}{(4 * \pi)^3 * P_{\text{пр min}}}}$$

Де $P_{\text{пер}}$ – потужність передавача
 G – коефіцієнт підсилення антени
 $\sigma_{\text{ц}}$ – ефективна площа розсіювання цілі
 λ – довжина хвилі

$P_{\text{пр min}}$ – мінімальна потужність приймача.

Постає питання що таке мінімальне потужність приймача. Це найменша потужність сигналу, яку може сприйняти приймач для успішного виявлення сигналу. Загалом це значення визначається як нижня межа потужності сигналу, яка є достатньою для виявлення і обробки радаром інформації про цільовий об'єкт.

Щоб знайти мінімальну потужність приймача слід зробити елементарні математичні перетворення. Після перетворень ми використаємо деякі табличні значення із технічного завдання, а саме коефіцієнт сигнал шум, температуру навколишнього середовища і сталу Больцмана. Формула для знаходження мінімальної потужності передавача буде така: добуток коефіцієнт шум, температура навколишнього середовища, стала Больцмана, ширина спектру сигналу на відношення сигнал шум поділене на два.

$$P_{\text{пр}} = \frac{q}{2} * K_{\text{ш}} * T_0 * k * \Delta f = 1.306 * 10^{-11}$$

При обробці складних сигналів у радіолокаційних системах з частотно-модульованим неперервним періодичним випромінюванням використовується оптимальний фільтр, який складається з двох основних частин. Спочатку сигнал стискається за допомогою процедури стиснення, що змінює його тривалість. Це може бути досягнуто, наприклад, застосуванням фільтру зі скінченною відповіддю або іншими методами обробки сигналу, що дозволяють стиснути сигнал у часовій площині.

Другою частиною оптимального фільтра є фільтрація окремого імпульсу. Під час цього процесу враховується лише оптимальна частина сигналу, що має найбільш значущість з точки зору покращення співвідношення сигнал/шум. В результаті цього фільтраційного процесу одиночний імпульс стискається, тобто його тривалість зменшується.

Цей підхід до оптимального фільтрування у радіолокаційних системах включає коефіцієнт стиснення сигналу. Це означає, що шумові компоненти розподіляються по всій частотній площині, а сигнал стискається до вузького піку. Таким чином, застосування оптимального фільтра в складних сигналах виробляє необхідний сигнал-шум в вихідному сигналі.

2.4 Розрахунок мінімальної потужності приймача

При використанні порогового пристрою на вході радіолокаційної системи ми отримуємо сигнал або відсутність сигналу. Однак на виході необхідно забезпечити присутність сигналу-шуму. З огляду на оптимальну обробку сигналу, ми повинні знати відношення сигнал-шум, приведені до входу приймача. Це дозволить визначити потужність сигналу у приймачі, оскільки кореляційна інформація та оптимальна фільтрація можуть змінювати величину сигнал шуму. Таким чином, ми повинні обчислити приведені відношення сигнал шуму.

Згідно з технічним завданням вид сигналу має м послідовний вигляд. Для обробки такого сигналу використовують формулу два в степені енн, таким чином ми дізнаємося коефіцієнт стиснення сигналу.

$$K_{сж}=2^5 - 1 = 31$$

Після знаходження коефіцієнта стиснення сигналу, ми маємо змогу знайти коефіцієнт сигнал шум приведене до виходу, тобто до обробки в оптимальному фільтрі.

$$q_1 = \frac{q}{K_{сж}} = 0.211 \text{ рази}$$

$$10 * \log(q_1) = -6.753 \text{ дБ}$$

Після того як ми знайшли коефіцієнт сигнал шум приведене, ми можемо дізнатися мінімальну потужність приймача. Будемо використовувати стару формулу, тільки із однією правкою що коефіцієнт сигнал шум тепер у нас преведений.

$$P_{pmin} = \frac{q_1}{2} * K_{ш} * \Delta f * T_0 * k = 1.359 * 10^{-14} \text{ Вт}$$

За допомогою вищезгаданої формули, ми можемо визначити потужність передавача в імпульсі, необхідну для ефективного виявлення мінімального сигналу з певною ймовірністю. Це значення визначається залежністю від ефективності, з якою потрібно виявити мінімальний сигнал і від відношення сигнал/шум.

2.5 Розрахунок параметрів антени

Метод ітерацій дозволив нам визначити оптимальну довжину хвилі для нашої системи. Довжина хвилі дорівнює

$$\lambda = 0.03 \text{ м}$$

Використовуючи довжину світла ми маємо змогу обчислити частоту світла.

$$f_n = \frac{c}{\lambda} = 1 * 10^{10} \text{ Гц}$$

Ця формула вказує на зворотну пропорційність між частотою світла та його довжиною хвилі. Коли довжина хвилі збільшується, частотам зменшується і навпаки.

Після знаходження довжини хвилі також ми маємо змогу знайти параметри системи антени. Можемо розрахувати площу антени. Для цього спочатку розрахуємо розмір сторони антени по азимуту та по куту місця.

$$da1 = \frac{\lambda * 60}{e} = 1.2 \text{ м}$$

Розмір сторони антени по азимуту відображає покриття антени. Цей параметр називають також горизонтальною або горизонтальною шириною променевої діаграми антени. Він визначається конструкцією антени та її напрямною характеристикою.

$$da2 = \frac{\lambda * 60}{e} = 0.124 \text{ м}$$

Розмір сторони антени по куту місця відображає вертикальну зону покриття антени. Цей параметр називають вертикальною шириною променевої діаграми антени. Після знаходження сторін антени маємо змогу знайти площу антени.

$$S = \frac{da1 * da2 * \pi}{4} = 0.117 \text{ м}^2$$

Після того як знайшли площу ми можемо знайти коефіцієнт підсилення антени

$$G = \frac{4 * \pi * S * h}{\lambda^2} = 1.389 \text{ рази}$$

Коефіцієнт підсилення антени вказує на здатність антени підсилити сигнали у певному напрямку. Високий коефіцієнт підсилення означає сильніший сигнал у заданому напрямку, що корисно для покращення зони покриття зв'язку

$$G1 = 10 * \log(G) = 31.426 \text{ дБ}$$

Із основного рівняння радіолокації знаходимо потужність передавача.

$$P_{\text{пер}} = \frac{R_{\text{max}}^4 * (4 * \pi)^3 * P_{\text{прmin}}}{G^2 * 1 * \lambda^2} = 1.258 * 10^4$$

Після того як нам відомі всі параметри, ми можемо підрахувати яка максимальна дальність нашого приймача. Єдине нам не вистачає ефективної апе-

ртури приймальної антени. Ефективна апертура приймальної антени - це фізичний параметр, який вимірюється в квадратних метрах і вказує на площу поверхні антени, яка здатна ефективно збирати електромагнітні хвилі з навколишнього простору. Для знаходження цього параметру нам треба знати довжину хвилі сигналу та коефіцієнт підсилення антени.

$$A = \frac{\lambda^2 * G}{4\pi} = 0.028$$

Після визначення ефективної апертури приймальної антени, ми можемо продовжити пошук максимальної дальності радіолокаційного зондування. Вираз для розрахунку максимальної дальності враховує такі фактори як потужність передавача, коефіцієнт підсилення антени, ефективна площа розсіювання цілі, щільність потоку енергії падаючої хвилі та відстань до цілі.

$$R_{max} = \left(\frac{P_{пер} * G * \sigma * A}{4\pi^2 * S} \right)$$

В цій формулі присутня ефективна площа розсіювання. Ця характеристика важлива для оцінки здатності цілі відбивати радіосигнали. Ефективна площа розсіювання визначається як відношення потужності електромагнітної енергії, яка відбивається цілю в напрямку радару до поверхні щільності потоку енергії падаючої хвилі. Загалом ЕПР визначають за табличним значенням або ж за формулою. Використаємо табличні значення деяких повітряних об'єктів.

Варіант повітряного об'єкт №1 $\sigma_{ц1} = 1$ м_{КВ}

Варіант повітряного об'єкт №2 $\sigma_{ц2} = 0.1$ м_{КВ}

Варіант повітряного об'єкт №3 $\sigma_{ц3} = 0.01$ м_{КВ}

Для врахування швидкості параметрів об'єктів у визначені дальності використовують канали швидкості, розраховані на основі максимальної швидкості та роздільної здатності.

$$\text{Розрахунок каналів швидкості } N\nu = \frac{V_{max}}{\Delta V} = 20$$

Після врахування усіх факторів, ми можемо здійснити розрахунок максимальної дальності виявлення для кожного варіанту повітряного об'єкту.

$$\text{Варіант повітряного об'єкт №1 } R_{\max 1} = \sqrt[4]{\frac{P_{\text{пер}} * (G^2 * \sigma_{\text{ц1}} * \lambda^2)}{(4 * \pi)^3 * P_{\text{прmin}}} = 3 * 10^4 \text{ м}}$$

$$\text{Варіант повітряного об'єкт №2 } R_{\max 2} = \sqrt[4]{\frac{P_{\text{пер}} * (G^2 * \sigma_{\text{ц2}} * \lambda^2)}{(4 * \pi)^3 * P_{\text{прmin}}} = 1.687 * 10^4 \text{ м}}$$

$$\text{Варіант повітряного об'єкт №3 } R_{\max 3} = \sqrt[4]{\frac{P_{\text{пер}} * (G^2 * \sigma_{\text{ц3}} * \lambda^2)}{(4 * \pi)^3 * P_{\text{прmin}}} = 9.487 * 10^3 \text{ м}}$$

Розуміння цих показників є важливим для проектування та оптимізації радарних систем, що допомагає досягнути більш ефективного та точного зондування навколишнього простору.

3 ПРОГРАМНА ЧАСТИНА СИСТЕМИ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ РЛС

Системи радіолокаційної візуалізації відіграють важливу роль у виявленні, ідентифікації та відстеженні об'єктів у радіолокаційних системах. Ефективна програмна частина дозволяє операторам радарів отримувати точні та зрозумілі данні, необхідні для прийняття важливих рішень.

Програмна частина включає в себе набір програм і алгоритмів які дозволяють збирати, обробляти і відображати дані з радіолокаційних датчиків. Одне з основних завдань програмної частини є обробка великих обсягів даних з радарів. Вона включає фільтрацію, сегментацію, видалення шумів та інші алгоритми обробки сигналів. Це дозволяє отримувати точні дані про об'єкти, їхнє положення та рух. Для того щоб візуально представити інформацію, дані після обробки передаються до модуля відображення. Цей модуль включає в себе графічні інтерфейси, дисплеї та інші засоби візуалізації.

Враховуючи важливу роль систем відображення радіолокаційної інформації, необхідне ефективне програмне забезпечення стає необхідною для забезпечення високої якості та точності отриманих даних. Ця система також дозволяє зробити процес виявлення та відстеження об'єктів більш автоматизованим та ефективним, зменшуючи навантаження на операторів та покращуючи їх робочий процес.

3.1 Загальна структура бібліотек QT для графічних інтерфейсів

QT - безкоштовна крос платформна структура для розробки програмного забезпечення, настільних мобільних пристроїв, для різних платформ таких як Windows, Linux, MacOS. Підтримується багатьма мовами програмування. Сама QT не є мовою програмування, а лише фреймворк написаний мовою програмування C++.[13] Основні призначення QT є сприяння розробці крос платформного програмного забезпечення з графічним інтерфейсом. Він надає розробникам потужні інструменти та компоненти. Основні призначення включа-

ють розробку програмного забезпечення яке буде працювати на різних операційних системах, без великих змін у коді. Платформа допомагає розробникам створювати інтерфейси зрозумілі для користувачів, так як має великий набір готових елементів та компонентів. Вона підтримує різні стилі оформлення і анімацію, що дозволить створювати сучасні та візуальні привабливі додатки. Система підтримує роботу з мережами, базами даних, мультимедіа.

Архітектура Qt одна із причин чому вона має таку популярність в розробників програмного забезпечення. Як було сказано раніше вона включає в себе набір компонентів, класів та модулів яка допомагає створювати між платформні графічні інтерфейси з високою продуктивністю та надійністю. Основною концепцією архітектури є «write once run anywhere».[13] Ця ідея означає що написавши код один раз і запустивши програму, платформа надає можливість запустити додаток на різних платформах без значних змін. Це досягається за допомогою мови програмування c++ і розширень які надає QT для реалізації функціональності. Архітектура QT базується на об'єктно орієнтованому програмуванні, що надає можливість розробникам легко моделювати складні зв'язки та структури у програмі. Класи та об'єкти Qt використовуються для створення інтерфейсів користувача, обробки подій, графіки, мережових операцій. Одним із ключових компонентів архітектури Qt є різноманітні модулі які охоплюють різні аспекти програмного забезпечення. До таких типів відносять модуль графічного відображення який дає змогу працювати з ростовою та векторною графікою, взаємодії з елементами інтерфейсу та користувачем для обробки подій. Архітектура QT також надає можливості інтеграції з іншими популярними бібліотеками дозволяючи розширювати функції програмного забезпечення та використовувати ресурси сторонніх розробників.

Розширені можливості бібліотеки QT роблять основні класи та компоненти необхідними для розробки графічного інтерфейсу користувача. Для прикладу клас Q Push Button дозволяє створювати кнопку з властивостями та обробками подій, тоді як клас Q Combo Box дозволяє вибирати елементи зі спадного списку. Для відображення тексту або зображень використовують клас Q

Label. Для вводу текстових даних треба вже буде використовувати клас Q Line Edit.[15] Компоненти, такі як Q Table View, надають можливість відобразити табличні дані з можливістю сортування та редагування. Класи Q Menu Bar та Q Menu спрощують створення зручних меню для додатків, тоді як Q File Dialog дозволяє користувачам зручно вибирати файли та папки через діалогові вікна. Клас Q Message Box дозволяє створювати діалогові вікна з повідомленнями для користувача, а клас Q Painter надає можливості для рисування на графічних елементах та створення власних графічних ефектів.

Ці класи та компоненти належать до основних будівельних блоків бібліотеки QT, які втілюють в собі надзвичайно потужну та вдосконалену архітектуру. Ця архітектура відкриває безліч можливостей для розробників, щоб втілити свої ідеї та створити графічні інтерфейси, які вражають елегантністю та виразністю.

Компоненти QT допомагають розробникам забезпечити не тільки естетично привабливий дизайн, але й функціональність, що задовольняє потреби користувачів і робить їх досвід використання додатків більш зручним та задоволеним. У бібліотеці QT існують глибокі взаємозв'язки між різними класами та компонентами, які допомагають створювати потужні інтерфейси. Наприклад, кнопки можна вставити в меню, яке, у свою чергу, можна розмістити на панелі меню.

Клас Q Label може відобразити текстові дані або зображення, які можна встановити за допомогою методів set Text і set QPixmap. Користувачі можуть ввести цей текст за допомогою класу Q Line Edit.

Ви також можете спостерігати за зв'язком між класами Q Table View і Q Abstract Table Model, де клас Q Abstract Table Model використовується для представлення табличних даних у Q Table View. Такі методи, як set Model() і resize Columns To Contents, дозволяють керувати відображенням даних і змінювати розмір стовпців у таблиці. Клас K Layout використовується для реда-

гування позиції елементів у вікні чи контейнері. Він може містити різні елементи, такі як кнопки, текст або навіть інші контейнери, і керувати їх положенням і зсувом за допомогою методів `set Spacing` і `set Alignment`.

Таким чином, завдяки зв'язкам між цими класами та компонентами можна створювати складні інтерфейси, які поєднують різні елементи, спілкуватися з користувачами та забезпечувати ефективне відображення та керування даними.

QT розділено на різні модулі та підсистеми, що охоплюють різні аспекти розробки графічного інтерфейсу користувача. Такий поділ дозволяє розробникам використовувати лише компоненти, необхідні для їхніх проєктів, зменшуючи витрати та підвищуючи ефективність. Взаємодія між модулями та підсистемами забезпечується відповідними інтерфейсами та з'єднаннями, що забезпечує правильну інтеграцію окремих компонентів та розширення функціональних можливостей. Ця модульна структура QT сприяє гнучкості, повторному використанню коду та полегшує розробку складних графічних інтерфейсів. Крім того, поділ QT на модулі та підсистеми забезпечує прозору організацію проєкту та полегшує його обслуговування. Кожен модуль або підсистема виконує певні завдання і має свою відповідність у розробці графічних інтерфейсів. Наприклад, модуль `Core` відповідає за основні функції та механізми, модуль `GUI` дозволяє створювати графічні елементи та керувати ними. Це дозволяє розробникам зосередитися на певній області розробки та полегшує командну роботу, коли кожен може працювати над власним модулем або підсистемою. Така структура сприяє ефективності та організованості проєкту та дозволяє розробникам швидко та легко реалізовувати свої ідеї та створювати якісні графічні інтерфейси.

Як вже зазначалось основним модулем є `QT Core` це серце QT і забезпечує базову функціональність для всіх інших модулів. Містить класи для роботи з подіями, полями, колекціями, таймерами, потоками та параметрами конфігурації. `Core` також містить сигнальні механізми та слоти, які дозволяють

об'єктам спілкуватися один з одним, спрощуючи розробку асинхронних операцій і обробку подій. Модуль GUI відповідає за графічні елементи інтерфейсу користувача. Містить класи для створення вікон, кнопок, тексту, зображень, меню, панелей інструментів і діалогових вікон і керування ними. Модуль графічного інтерфейсу користувача забезпечує події миші та клавіатури, маніпуляції графічним контекстом і малювання екрана. Модуль Widgets містить класи високого рівня для розробки графічних інтерфейсів. Він містить такі компоненти, як кнопки, поля введення, списки, таблиці, вкладки, дерева тощо, які можна розміщувати у вікнах і взаємодіяти з ними. Модуль Widgets дозволяє швидко створювати інтерактивні інтерфейси за допомогою готових елементів керування та елементів макета. Мережевий модуль забезпечує функції для взаємодії з мережею. Він надає класи для роботи з TCP/IP, UDP, HTTP, FTP та іншими мережевими протоколами. Мережевий модуль дозволяє виконувати операції з використанням мережевих розеток, надсилати та отримувати дані по мережі, створювати серверні та клієнтські програми, а також спілкуватися по мережі.

Комбінуючи функції та класи з цих модулів, розробники можуть створювати різноманітні інтерактивні програми, які можуть спілкуватися з мережею, відображати графіку та реагувати на дії користувача.

Однією з ключових особливостей QT є те, що він підтримує інтеграцію з кількома мовами програмування, включаючи C++, Python, Java тощо. Це дозволяє розробникам використовувати QT у поєднанні з іншими мовами або створювати гібридні програми, які поєднують різні технології.

QT також надає можливість інтеграції з веб-службами та API, дозволяючи вам викликати віддалені сервери, отримувати дані та взаємодіяти з іншими службами. Це дозволяє розробникам створювати програми, які використовують функції Інтернету, такі як отримання даних із баз даних, використання соціальних мереж або доступ до веб-служб. QT також має вбудовану підтримку графічних бібліотек, таких як Open GL, що дозволяє розробникам створювати 3D-графіку та використовувати потужні графічні можливості. Крім того, QT

також забезпечує інтеграцію з такими популярними базами даних, як My SQL, Postgre SQL, S Q Lite тощо, що дозволяє зручно зберігати та обробляти дані.

3.2 Склад бібліотек для реалізації графічного інтерфейсу РЛС

Розробка програмного забезпечення для радіолокаційних станцій вимагає врахування багатьох факторів, одним з найважливіших з яких є вибір і використання правильних бібліотек для реалізації графічного інтерфейсу радіолокаційної станції. Це необхідно для створення потужного, інтерактивного та ефективного інтерфейсу, який відображає важливі дані з радіолокаційної станції та надає користувачеві корисні інструменти для взаємодії з системою. Спеціалізовані бібліотеки для реалізації радіолокаційних графічних інтерфейсів дозволяють розробникам ефективно використовувати функціональність цих бібліотек для створення високоякісних візуальних інтерфейсів. Такі бібліотеки пропонують гнучкість і зручність для розробників, надаючи широкий спектр інструментів, які полегшують процес створення і конфігурації.

Ці бібліотеки можна використовувати для реалізації різних елементів інтерфейсу, таких як графіки, діаграми, таблиці, кнопки та перемикачі. Вони також підтримують анімацію та інтерактивність, дозволяючи користувачам взаємодіяти з системою за допомогою пристроїв введення, таких як миша або клавіатура.

Фреймворк QT є необхідною бібліотекою для реалізації радіолокаційних графічних інтерфейсів. Цей потужний інструментарій розробки надає багато можливостей для створення крос-платформних додатків з високоякісними інтерфейсами. QT значно спрощує процес розробки графічних інтерфейсів і базується на широкому спектрі модулів і класів, які надають розширені функціональні можливості для маніпулювання вікнами, кнопками, таблицями, графіками і багатьма іншими елементами інтерфейсу. Вона базується на широкому наборі модулів і класів, які надають розширену функціональність для маніпулювання вікнами, кнопками, таблицями, графіками та багатьма іншими елеме-

нтами інтерфейсу. Фреймворк QT забезпечує високу переносимість коду і дозволяє розробникам створювати додатки з еквівалентною функціональністю для різних операційних систем, таких як Windows, MacOS та Linux. Це робить QT особливо корисним для розробників, дозволяючи їм використовувати свої навички та створювати інтерфейси, які працюють на різних платформах без внесення значних змін. Фреймворк надає набір готових рішень та інструментів для розміщення елементів інтерфейсу, керування подіями та виконання багатьох інших завдань, пов'язаних з розробкою графічного інтерфейсу. Він також підтримує можливості анімації, що дозволяє створювати привабливі та динамічні ефекти в інтерфейсах, які приваблюють користувачів і заохочують їх взаємодіяти з системою.

Однією з ключових бібліотек QT, використаних для реалізації графічного інтерфейсу РЛС, є QT Widgets. Ця бібліотека надає широкий набір готових елементів інтерфейсу, таких як кнопки, текстові поля, списки, таблиці та інші. За допомогою цієї бібліотеки ми змогли створити інтуїтивно зрозумілий та зручний інтерфейс для користувача. QT Widgets дозволяє нам легко керувати вікнами, їх розмірами та видимістю. Ми могли створювати нові вікна, розташовувати елементи відповідно до потреб і налаштовувати їх зовнішній вигляд. Модуль також надає можливість взаємодії з користувачем через обробку подій, таких як натискання кнопок або введення тексту. Однією з важливих можливостей QT Widgets є підтримка зміни вікон. Ми могли реагувати на зміну розміру вікна та виконувати певні дії при цьому. Наприклад, ми могли перераховувати розміщення елементів, щоб забезпечити їх адаптивність до змін розміру вікна. Крім того, за допомогою QT Widgets ми могли оновлювати зміст вікон при необхідності. Це дозволяло нам динамічно відображати потрібну інформацію та змінювати стан елементів інтерфейсу залежно від поточного контексту.

Другою важливою бібліотекою в складі QT, використаною для реалізації графічного інтерфейсу РЛС, є QT Core. Цей модуль надає розширений набір

функцій та класів для роботи з різними аспектами програмування. QT Core надає нам потужні інструменти для роботи зі стрічками, контейнерами, подіями, потоками, файловою системою та іншими базовими елементами програмування. Ми використовували функції та класи QT Core для різних завдань у нашому проєкті. Наприклад, ми використовували функції для роботи зі стрічками, такі як отримання під рядків, з'єднання стрічок та форматування тексту. Це дозволяло нам маніпулювати даними та відображати їх у відповідному форматі. Ми також використовували контейнери QT Core, такі як списки та мапи, для зберігання та організації даних. Це спрощувало доступ до різних елементів та забезпечувало швидкий доступ до них. Модуль QT Core надавав нам можливість для роботи з подіями. Ми могли реагувати на події, такі як натискання кнопок або зміна значень, та виконувати відповідні дії. Це дозволяло нам створювати інтерактивний інтерфейс та забезпечувати взаємодію користувача з програмою. Крім того, ми використовували функції QT Core для роботи з файловою системою. Ми могли створювати, відкривати, зберігати та зчитувати файли, що дозволяло нам працювати з важливими даними та зберігати їх у відповідному форматі.

Для реалізації графічного відображення даних з РЛС використовується QT GUI. Ця бібліотека містить класи і функції, що дозволяють створювати та керувати графічним інтерфейсом. Завдяки модулю QT GUI ми змогли створити привабливий та функціональний графічний інтерфейс для нашої системи РЛС. Ми використовували цей модуль для розміщення та відображення різних візуальних елементів, таких як кнопки, тексти, списки, таблиці та графіки. За допомогою QT GUI ми могли визначати зовнішній вигляд елементів, встановлювати кольори, шрифти, розміри та інші властивості, щоб забезпечити зручну та привабливу візуалізацію даних. Крім того, модуль QT GUI дозволяв нам взаємодіяти з користувачем через різні події. Ми могли реагувати на натискання кнопок, переміщення миші або вводу тексту, і виконувати відповідні дії. Це дозволяло нам забезпечити інтерактивність та реагування на дії користу-

вача. Одним з важливих аспектів QT GUI було керування розміщенням та організацією елементів на екрані. Ми могли використовувати різні контейнери та механізми розмітки, щоб створювати гнучкі та ефективні інтерфейси. Ми також мали можливість працювати зі стилістичним аркушами, щоб налаштувати зовнішній вигляд елементів відповідно до вимог проекту.

Для взаємодії з графічним інтерфейсом та обробки подій використовується QT Event System. Ця система надає зручний механізм для обробки різних подій, що відбуваються в нашій системі РЛС. Завдяки функціоналу QT Event System, ми могли визначати обробники подій, які спрацьовують, коли відбувається певна подія. Наприклад, ми могли зареєструвати обробник для натискання кнопок, переміщення миші або введення з клавіатури, і виконувати певні дії відповідно до цих подій. За допомогою цього механізму, ми змогли забезпечити відповідний взаємозв'язок між графічним інтерфейсом та логікою нашої системи. Крім обробки вбудованих подій, QT Event System також дозволяв нам створювати власні події та передавати дані між об'єктами в системі. Ми могли створювати власні події, які відображали важливі зміни стану системи РЛС, і передавати необхідну інформацію з одного об'єкта до іншого. Це давало нам гнучкість та можливість забезпечувати високу рівень співробітництва між різними компонентами системи. QT Event System також надавав розширені можливості для обробки подій миші та клавіатури. Ми могли отримувати інформацію про позицію миші, її рух, натискання клавіші та інші подібні дані. Це було особливо корисним при реалізації функціоналу, який вимагав взаємодії з користувачем за допомогою миші або клавіатури.

3.3 Структурна схема програмного рішення

Структурна схема програмного рішення є невід'ємною та незамінною складовою процесу розробки програмного забезпечення. Вона візуалізує та систематизує архітектуру програми, надаючи зрозумілу структуру та організацію компонентів, модулів та їх взаємозв'язків. Структурна схема дозволяє розуміти як окремі елементи співпрацюють між собою, як вони обмінюються

даними сигналами та здійснюють взаємодію. Вона надає велику кількість інформації, необхідної для глибокого розуміння архітектурних рішень, які втілюються в програмі. Структурна схема програмного рішення нашої програми показана на Рис. 3.1

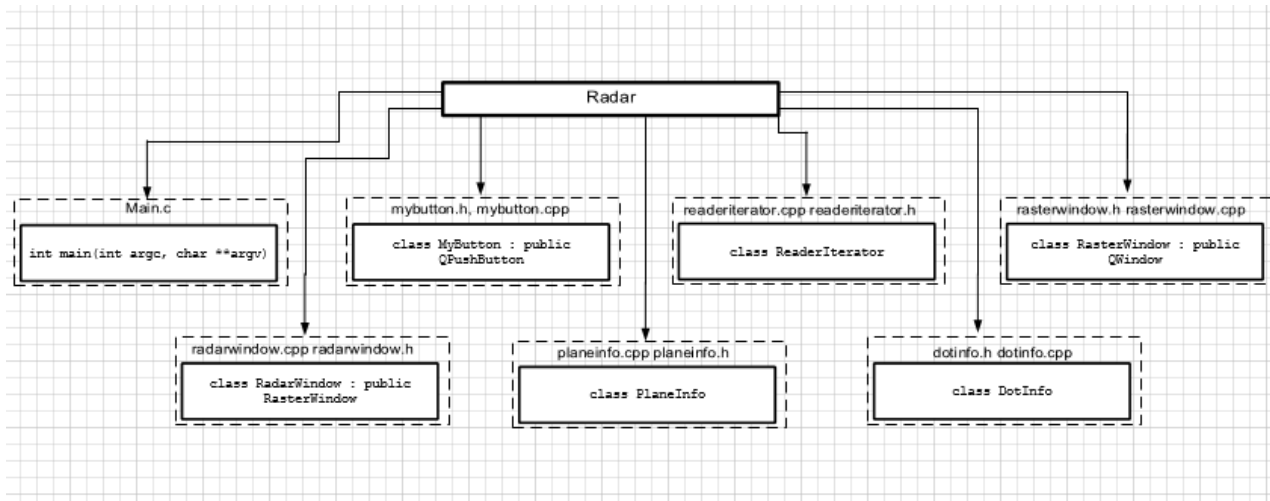


Рис. 3.1 – Схема структурна програми

Елемент `main.cpp` у схемі програмного рішення представляє головний модуль програми. Основною функцією `main` є включення необхідних заголовочних файлів, які містять оголошення класів і функцій, необхідних для програми. У файлі `main.cpp` також можуть бути присутні глобальні змінні, які використовуються в різних частинах програми. Функція `main` є точкою входу в програму і виконує послідовність операцій для запуску та виконання програми. Це може включати створення об'єктів, ініціалізацію змінних, взаємодію з користувачем, виклики функцій. Елемент `main.cpp` у схемі програмного рішення є головним модулем, який визначає основні кроки програми і організовує взаємодію з іншими компонентами системи. Він включає в себе необхідні функції та класи, а також координує роботу інших модулів програми.

Елемент `my button` на схемі програмного рішення відповідає за реалізацію елемента інтерфейсу, який може мати форму прямокутника. Цей елемент представлений у вигляді двох файлів: `mybutton.h` та `mybutton.cpp`. У файлі `mybutton.h` оголошується клас та його методи, які описують поведінку цього прямокутника. В цьому файлі можуть бути визначені змінні, функції-члени класу,

сигнали які використовуються для роботи з елементом `mybutton`. Також у цьому файлі можуть бути описані властивості, стилізація та інші параметри, які впливають на вигляд інтерфейсу цього елемента. Файл `mybutton.cpp` містить реалізацію методів класу, які були оголошені у файлі `mybutton.h`. В цьому файлі реалізується логіка кнопки, включаючи обробку подій, зміну стану кнопки та відображення графічних ефектів. Тут можуть бути визначені функції-члени класу, які забезпечують взаємодію з іншими елементами інтерфейсу або зовнішніми модулями.

Елемент `readeriterator` на схемі програмного рішення відповідає за реалізацію ітератора для читання даних. Ітератор є об'єктом, який надає доступ до послідовної колекції елементів і дозволяє послідовно перебирати ці елементи без розкриття деталей і реалізації колекції. У файлі `readeriterator.h` оголошується клас `readeriterator` та його методи, які описують поведінку ітератора. Цей клас може мати методи для переміщення до наступного елемента, перевірки кінця колекції, отримання значення поточного елемента. У файлі `readeriterator.cpp` виконуються операції, необхідні для роботи ітератора, такі як переміщення по колекції, збереження поточного стану ітератора, отримання значень елементів та інші операції, пов'язані з читанням даних. Елемент `readeriterator` виконує важливу функцію ітератора, що дозволяє послідовно читати дані з колекції без прямого доступу до її внутрішньої реалізації. Він забезпечує зручний і безпечний спосіб отримання даних з колекції, дозволяючи перебирати її елементи один за одним.

Елемент `rasterwindow` на схемі програмного рішення відповідає за реалізацію вікна зображення векторних графічних об'єктів які відображаються у форматі растрової графіки. Цей елемент включає файли `rasterwindow.h` і `rasterwindow.cpp`, де описуються клас і його методи для керування вікном та відображення графічних об'єктів. У файлі `rasterwindow.h` оголошується клас, який включає методи для створення вікна, обробки подій, малювання графічних об'єктів тощо. Також в цьому файлі можуть бути оголошені глобальні змінні, константи та додаткові класи, які використовуються в роботі з вікном.

У файлі `rasterwindow.cpp` реалізується функціонал класу `RasterWindow`. Цей файл містить код, який виконує різні операції з вікном, такі як створення вікна, ініціалізація графічних контекстів, обробка подій миші та клавіатури, малювання графічних об'єктів на вікні тощо. Код цього файлу реалізує логіку взаємодії з вікном і забезпечує коректне відображення графічних об'єктів. Елемент `rasterwindow` виконує важливу функцію в програмному рішенні, забезпечуючи візуалізацію векторних графічних об'єктів на растровому вікні. Він дозволяє користувачеві бачити і взаємодіяти з графікою, що відображається на екрані, і забезпечує необхідні можливості для створення, маніпулювання та відображення графічних об'єктів у програмі.

Елемент `planeinfo` на схемі програмного рішення відповідає за реалізацію класу `PlaneInfo`, який представляє інформацію про об'єкт. У цьому файлі оголошується клас, який містить різні властивості повітряного об'єкта, такі як час, радіус, висота, швидкість, індекс. Ці властивості описують стан і характеристики повітряного об'єкта в конкретний момент часу. У файлі `planeinfo.cpp` реалізується функціонал класу `PlaneInfo`. Цей файл містить визначення методів класу, таких як конструктор і деструктор, які ініціалізують об'єкт і здійснюють необхідні операції при його знищенні. Також в цьому файлі можуть бути реалізовані додаткові методи для роботи з властивостями об'єкта, наприклад, методи для отримання та зміни значень властивостей. Елемент `planeinfo` на схемі програмного рішення забезпечує зберігання та обробку інформації про повітряний об'єкт.

У файлі `dotinfo.h` оголошується клас, який містить різні властивості точки. Ці властивості включають координати x і y , висоту, швидкість та індекс точки. Вони описують стан і характеристики цієї точки. Клас `DotInfo` також має додаткові властивості, пов'язані з візуальним відображенням точки. Наприклад, `fadingTime` визначає час, через який точка зникне, а `timeToFade` показує, скільки часу залишилося до її зникнення. Елемент `dotinfo` на схемі програмного

рішення виконує функцію зберігання та обробки інформації про точки. Об'єкти класу DotInfo можуть використовуватись іншими компонентами програми для отримання, зміни та відображення даних про точки.

Елемент radarwindow на схемі програмного рішення відповідає за реалізацію класу RadarWindow, який представляє вікно радару. У файлі radarwindow.h оголошується клас, який містить методи та властивості для відображення та управління радаром. Цей клас може включати методи для малювання радарного екрану, відображення точок та інших об'єктів, а також методи для зміни налаштувань радару, обробки подій та взаємодії з користувачем. Файл radarwindow.cpp містить визначення методів класу, де реалізована функціональність цього класу. Елемент radarwindow на схемі програмного рішення виконує функцію управління та відображення радару в програмі. Він забезпечує інтерфейс між користувачем та функціональністю радару, дозволяючи користувачеві взаємодіяти з радаром, налаштовувати його параметри та спостерігати за відображеними на екрані об'єктами, такими як точки.

3.4 Переваги QT для реалізації програмного рішення

QT є потужним фреймворком для розробки програмного забезпечення, який надає розробникам широкі можливості і має свої переваги та недоліки.

Серед переваг QT можна виділити декілька функцій. Однією із них є багатофункціональність. Широкий спектр модулів і компонентів які будуть допомагати в розробці різноманітних функціональних можливостей. Це включає графічний інтерфейс користувача, мережеву взаємодію, роботу з базами даних. Завдяки цим готовим компонентам розробники можуть швидко та легко додавати потрібні функціональні можливості до своїх програм.

Іншою зі значних переваг QT його легкість використання. Він надає інтуїтивно зрозумілі інструменти, що спрощують процес розробки. Вбудований візуальний дизайнер зменшить необхідність у ручному кодуванні інтерфейсу що економить час та зусилля. Документація та приклади коду пояснить функціональність та використання різних компонентів та модулів. Багато прикладів

коду демонструють різні сценарії використання та надають практичні приклади. Це сприяє швидкому освоєнню QT та забезпечує зручні ресурси для вирішення завдань.

Суттєва перевага QT є його потужна підтримка дизайну. QT постачається з потужним візуальним дизайнером. Цей інструмент дозволяє розробникам перетягувати та розміщувати інтерфейси користувача за допомогою перетягування та розміщення елементів. Візуальний дизайнер значно спрощує процес створення та налаштування інтерфейсу користувача. Багатий набір готових компонентів і шаблонів також спрощують створення. Це включає в себе кнопки поля введення таблиці списки меню. Ці компоненти мають стильне та сучасне оформлення. Також QT надає підтримку вбудованих стилів та тем які дозволяють розробникам легко змінювати зовнішній вигляд свого додатка.

Однією з важливих переваг QT є його масштабність, що дозволяє розробникам ефективно створювати програмне забезпечення різної складності та обсягу. Тобто QT підходить для розробки як невеликих додатків з простим інтерфейсом, так і складних програмних рішень з великою кількістю функцій. Він може бути використаний для створення навіть найбільших та найскладніших додатків, які вимагають багатифункціональності та широкого спектру можливостей. Розподілена розробка дозволить розробникам працювати над проектом одночасно з різних місць та використовувати системи контролю версій для керування кодом.

3.5 Недоліки QT для реалізації програмного рішення

Хоча QT має безліч переваг, варто також враховувати його недоліки при реалізації програмного рішення.

Один з таких недоліків полягає у великому розмірі самого фреймворку. Це може призвести до збільшення розміру кінцевого програмного продукту, що може стати проблемою, особливо для мобільних додатків з обмеженим обсягом пам'яті.

Ще одним недоліком є необхідність вивчення та освоєння QT. Якщо ви новачок у програмуванні або не маєте досвіду роботи з фреймворком, це може зайняти певний час і вимагати додаткового зусилля. Потрібно ознайомитися з документацією, вчитися новим концепціям та особливостям . QT хоча і пропонує багато можливостей і компонентів, що робить його потужним, але це також може бути причиною складнощів. Навчання та освоєння всіх можливостей QT може зайняти певний час і вимагати додаткового зусилля.

Також слід враховувати, що використання QT робить ваше програмне забезпечення залежним від цього фреймворку. Якщо в майбутньому виникає необхідність перейти на інший фреймворк або технологію, це може вимагати значних зусиль і змін у кодї.

4 АПАРАТНА ЧАСТИНА СИСТЕМИ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ РЛС

Апаратна частина системи відображення інформації РЛС відіграє критичну роль у забезпеченні ефективного та точного функціонування радару. Вона включає в себе комплексний набір апаратних компонентів, які спільно працюють для прийому, обробки та відображення інформації, отриманої від радіосигналів. Ця апаратна частина включає в себе різноманітні складові, такі як антени для прийому радіосигналів, приймачі для збору та підсилення сигналів, аналогові та цифрові схеми для обробки сигналів, елементи відображення для візуалізації інформації, а також блоки управління та інтерфейсу для взаємодії з оператором.

Ефективність апаратної частини визначається як якість отримуваних сигналів, так і швидкість та точність обробки цих сигналів. Вона повинна забезпечувати широкий діапазон робочих частот, високу чутливість та низький рівень шумів, що дозволяє виявляти й відстежувати об'єкти на значній відстані з високою роздільною здатністю.

Крім того, апаратна частина повинна бути надійною, має бути витривалою до екстремальних умов експлуатації, включаючи високі температури, вологість, вібрації та інші небезпеки.

4.1 Розробка електричної схеми приймального пристрою радіолокаційної системи

Електрична схема приймального пристрою радіолокаційної системи представляє собою структуровану схему, що визначає спосіб підключення та взаємодії компонентів які приймають радіосигнали з антен та забезпечують їх подальшу обробку.

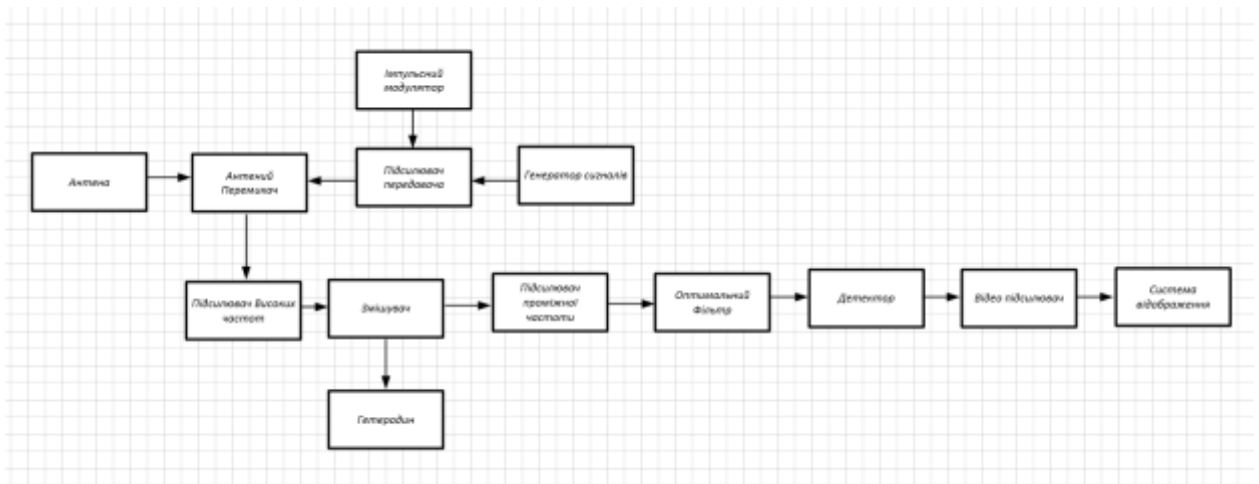


Рис. 4.1 – Електрична схема приймального пристрою

Електрична схема приймального пристрою складається з різноманітних елементів таких як:

Антенна: Антенна є вхідним елементом системи, вона приймає радіосигнали від об'єктів у просторі і перетворює їх на електричні сигнали.

Антенний перемикач: Це перемикач, який вибирає одну з кількох антен для отримання сигналу. В залежності від режиму роботи РЛС або конкретної задачі, може бути вибрана різна антена.

Підсилювач передавача: Це електронний пристрій, який збільшує потужність сигналу перед тим, як він буде переданий через антену. В приймальному пристрої РЛС, цей підсилювач служить для підсилення слабого вхідного сигналу, отриманого від антени.

Генератор сигналів: Це пристрій, який генерує радіочастотний сигнал, що використовується для передачі сигналів від РЛС до об'єктів у просторі.

Підсилювач високих частот: Цей підсилювач збільшує потужність високочастотного сигналу перед подальшою обробкою.

Змішувач: Це електронний пристрій, який комбінує вхідний високочастотний сигнал зі сигналом від генератора, що в результаті створює проміжну частоту.

Підсилювач проміжної частоти: Цей підсилювач збільшує потужність проміжно частотного сигналу перед подальшою обробкою.

Оптимальний фільтр: Це фільтр, який використовується для відсіювання небажаних частот у сигналі і залишення лише необхідних частот проміжної частоти.

Детектор: Це пристрій, який виявляє або витягує корисну інформацію зі сигналу проміжної частоти. В радіолокаційних системах, детектор може використовуватись для виявлення відображених радіосигналів від об'єктів у просторі.

Відео підсилювач: Цей підсилювач збільшує потужність вихідного сигналу з детектора перед подачею на систему відображення.

Система відображення: Це частина РЛС, яка відповідає за візуалізацію отриманої інформації. Це може бути екран, на якому відображаються відображені об'єкти або дані про їхні характеристики.

4.2 Розробка структурної схеми сублока відображення

Розробка структурної схеми сублока відображення є важливим етапом в процесі реалізації апаратної частини системи відображення інформації радіолокаційної системи. Цей підрозділ фокусується на проектуванні та створенні структурної схеми, яка відповідає вимогам функціональності та забезпечує правильне відображення інформації на екрані радару.

Структурна схема сублока відображення включає в себе компоненти, що відповідають за обробку, збереження та відображення отриманих даних. Це такі елементи як графічний процесор, пам'ять, інтерфейси звуку. Кожен компонент виконує свою функцію у процесі обробки та відображення даних на екран.

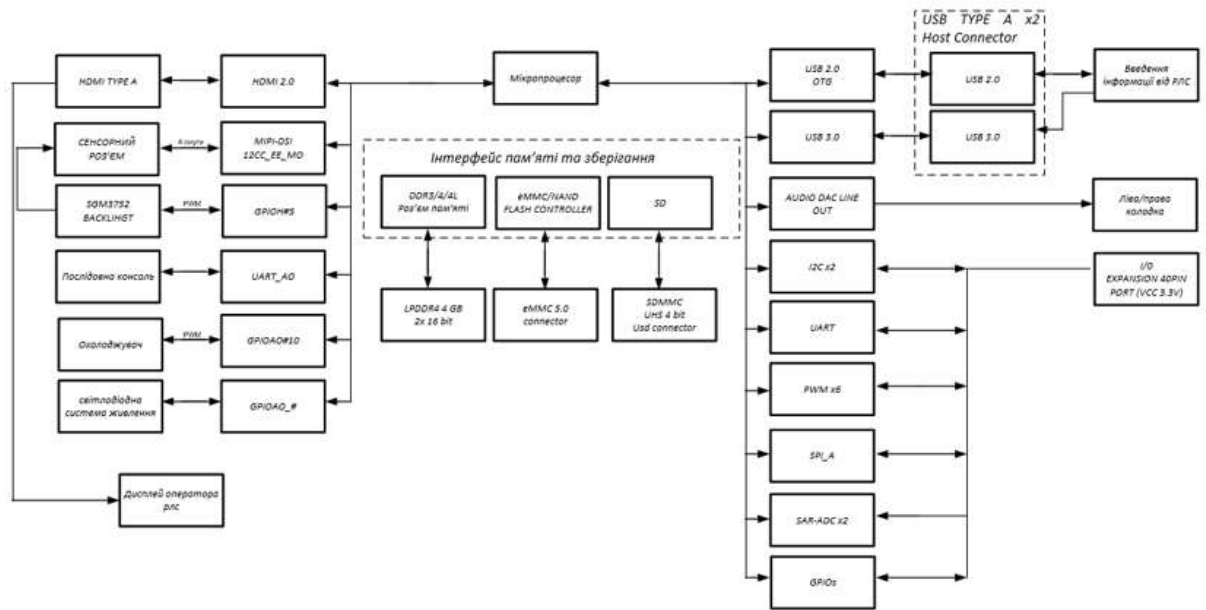


Рис 4.2 – Розробка структурної схеми субблока відображення

Головна мета електричної схеми приймального пристрою полягає у забезпеченні ефективного та точного прийому радіосигналів з антен та їх подальшому перетворенні у відповідні електричні сигнали, які підлягають подальшій обробці та відображенню на екрані радару. У розглянутій структурній схемі важливе місце займає дисплей оператора радіолокаційної системи, який під'єднується за допомогою роз'єму HDMI типу А. Використання інтерфейсу HDMI 2.0 забезпечує передачу відеосигналів високої якості з підтримкою роздільної здатності до 4К, що забезпечує чітке та деталізоване відображення інформації. Для введення інформації використовуються роз'єми USB, які забезпечують швидку передачу даних. У вашому випадку, використання роз'єму USB 3.0 дозволяє передавати інформацію ще швидше, що може бути корисним для підключення пристроїв з великим обсягом даних, наприклад, зовнішніх жорстких дисків. Охолоджувач підключений до порту GPIO А0#10, який є загальним вводом/виводом і дозволяє мікропроцесору взаємодіяти з охолоджувачем. Це з'єднання забезпечує керування температурою та забезпечує оптимальні умови для роботи системи.

4.3 Розробка принципової схеми

Принципова схема, відображає основні компоненти та їх взаємозв'язок для забезпечення правильної роботи системи відображення інформації. Вона передає важливі аспекти функціональності та структури системи.

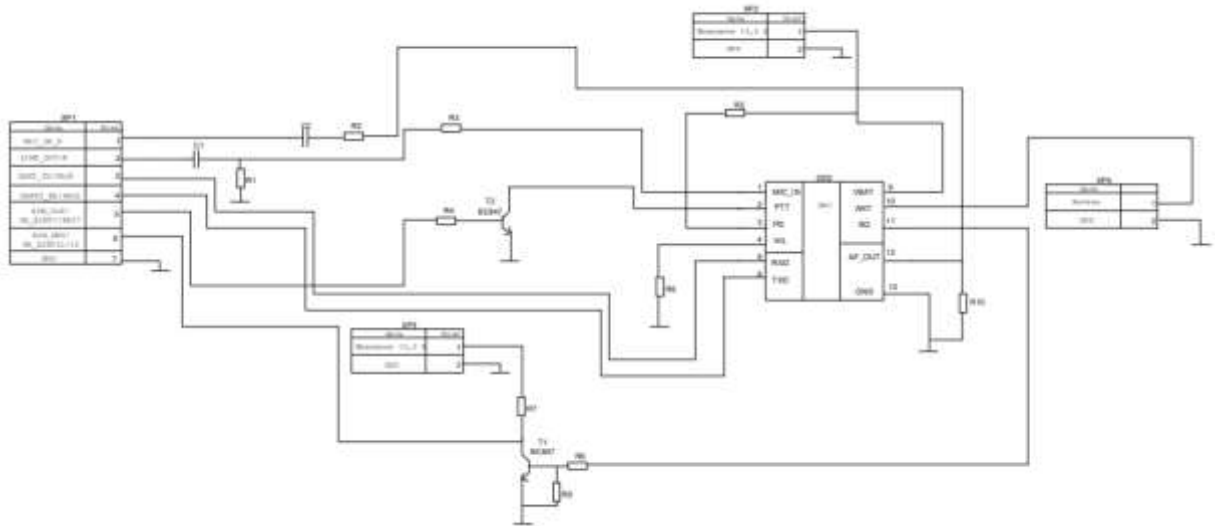


Рис. 4.3 – Принципова схема

Мікросхема, яка зображена на цій схемі, є ключовим елементом у пристрої, розробленому на основі технології 2.4G. Вона виконує роль мікропроцесора, який забезпечує обробку сигналів, передачу даних та керування виводами GPIO. Ця мікросхема має кілька особливостей, які роблять її ефективною та надійною. Перш за все, вона підтримує послідовний порт для прозорі передачі даних. Це означає, що дані можуть передаватись між передавачем і приймачем без будь-яких змін або обробки, забезпечуючи максимальну точність і надійність передачі. Крім того, ця мікросхема використовує аутентифікацію пакетів даних, що дозволяє уникнути отримання випадкових або недостовірних даних приймачем. Це забезпечує високу якість передачі і захист від спотворень чи помилок у даних. Мікросхема також підтримує керування комутацією, що дозволяє змінювати стан виводів GPIO на приймаючій стороні відповідно до отриманих команд. Це дає змогу керувати підключеними пристроями або виконувати різні дії залежно від отриманих сигналів.

В цій схемі використовуються два конденсатора з номіналом по 470 нФ. Крім того, на схемі присутні 10 резисторів з різними номіналами. У вузлі, що йде із порту та під'єднується до землі, резистори мають номінали 2,2 кОм та 1 кОм. На схемі ці резистори позначені як R1 та R10. Резистор, що йде до біполярного транзистора T2, має номінал 1 кОм. Заземлений резистор біля конденсатора C1 має розмірність 1 кОм. Резистори біля іншого транзистора мають такі номінали: R7 = 1 кОм, R9 = 56 кОм, а резистор R8 = 100 кОм. Тип мікросхеми є цифровою мікросхемою. На схемі позначено DD2.

Розробка принципової схеми мікропроцесора вимагає ретельного аналізу вимог до пристрою, вибору відповідного мікропроцесорного ядра, планування розташування компонентів на схемі, врахування енергоефективності, теплових показників та інших факторів. Оптимально розроблена принципова схема мікропроцесора гарантує ефективну та надійну роботу пристрою, швидку обробку даних і оптимальне використання ресурсів.

ВИСНОВКИ

У рамках даної дипломної роботи було проведено дослідження програмно апаратної системи відображення інформації радіолокаційної системи. Дослідження охоплювало аналіз міні комп'ютерів, розрахунок РЛС, програмну частину системи відображення інформації та апаратну частину системи. В результаті проведених робіт було отримано наступні висновки:

1. Загальний аналіз міні комп'ютерів показав їх ефективність та універсальність у застосуванні в системах імітатора РЛС. Ці міні комп'ютери забезпечують потужний обчислювальний потенціал, достатню кількість роз'ємів для підключення необхідних пристроїв.
2. Розрахунок параметрів РЛС дозволяє визначити оптимальні значення для роботи радіолокаційної системи. Це дозволяє досягти якості та точності відображення інформації.
3. Програмну частину системи відображення інформації можна досить добре описати на бібліотеках QT для графічних інтерфейсів. Вона має структуру, що дозволяє ефективно реалізовувати функціональні можливості систем.

Отримані результати показують, що програмно-апаратна система відображення інформації радіолокаційної системи є дієвим рішенням для досягнення точного інформаційного відображення. Ця система має потенціал для використання у різних сферах, таких як авіація, навігація, безпека та військова галузь. Перспективи розвитку даної системи полягають у вдосконаленні апаратних компонентів, удосконаленні алгоритмів обробки даних та розширенні функціональних можливостей програмного забезпечення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Радіолокаційні системи. Лабораторний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальностей 172 «Телекомунікації та радіотехніка» / В. О. Чмельов, П. Ю. Катін ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані – Київ : КПІ ім. Ігоря Сікорського, 2023. – 213 с.
2. Офіційний веб сайт Orange pi [Електронний ресурс] - <https://www.orangepi.org/>
3. Форум Orange pi [Електронний ресурс] - <http://www.orangepi.org/orangepibbsen/>
4. Orange Pi на Wikipedia [Електронний ресурс] - https://en.wikipedia.org/wiki/Orange_Pi
5. Офіційний веб сайт Raspberry Pi [Електронний ресурс] - <https://www.raspberrypi.org/>
6. Форум Raspberry pi [Електронний ресурс] - <https://www.raspberrypi.org/forums/>
7. Raspberry Pi на Wikipedia [Електронний ресурс] - https://en.wikipedia.org/wiki/Raspberry_Pi
8. Офіційний веб сайт Odroid-n2l [Електронний ресурс] - <https://www.hardkernel.com/shop/odroid-n2-with-4gbyte-ram/>
9. Форум Odroid-n2l [Електронний ресурс] - <https://forum.odroid.com/>
10. Odroid0n2l на Wikipedia [Електронний ресурс] - <https://en.wikipedia.org/wiki/Odroid-N2>
11. Теорія радіолокаційних систем (видання друге): підручник/ Б.Ф. Бондаренко, В.В. Вишнівський, В.П. Долгушин та іню; за заг.ред. С.В.

- Ленкова. – К. Видавничо-поліграфічний центр «Київський університет», 2011.- 383с.
- 12.Цифрова схемотехніка і архітектура комп'ютера(друге видання):підручник/Девід М. Харріс, Сара Л. Харис
- 13.Офіційний веб сайт QT [Електронний ресурс] - <https://www.qt.io/>
14. Офіційна документація QT [Електронний ресурс] - <https://doc.qt.io/>
15. Wikipedia від QT [Електронний ресурс] - <https://wiki.qt.io/Main>
- 16.Практичне використання РЛС та САПР Ревенко В.Ю.
- 17.Основи побудови радіолокаційних засобів розвідки повітряного простору : конспект лекцій / К. С. Васюта, О. В. Тесленко, В. М. Купрій, О. А. Малишев. – Х. : ХУПС, 2013. – 212 с.: іл.