

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: *Комп'ютерна система оцифрування потокового відеосигналу
для безпілотників*

Виконав: студент 4 курсу, групи СІ-41
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Коношенко Я. А.

(прізвище та ініціали)

Керівник

(підпис)

Лецишин Ю.З

(прізвище та ініціали)

Нормоконтроль

(підпис)

Луцик Н.С.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

Цуприк Г.Б

(прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Осухівська Г.М.
(підпис) (прізвище та ініціали)
«27» січня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студента Коношенко Ярослави Антонівни
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система оцифрування потокового відеосигналу для безпілотників

Керівник роботи доцент, к.т.н. Лецишин Юрій Зіновійович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 27 » січня 2025 року № 4/7-53

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз технічного завдання на розробку комп'ютерної системи оцифрування потокового відеосигналу

2. Проєктування комп'ютерної системи оцифрування потокового відеосигналу

3. Програмування та тестування комп'ютерної системи оцифрування потокового відеосигналу

4. Безпека життєдіяльності, основи охорони праці.

Висновки

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

1. Схема структурна

2. Схема електрична принципова

3. Схема монтажна (з'єднань)

4. Блок схема алгоритму роботи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Пилипець М.І. д.т.н. проф. каф. МТ</i>		

7. Дата видачі завдання 27.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Розробка технічного завдання.</i>	<i>27.01.2025р. – 02.02.2025р.</i>	<i>Викон.</i>
2.	<i>Робота над розділом аналіз технічного завдання на розробку комп'ютерної системи оцифрування потокового відеосигналу для безпілотників</i>	<i>02.02.2025р. – 03.03.2025р.</i>	<i>Викон.</i>
3.	<i>Робота над основним розділом проектування комп'ютерної системи оцифрування потокового відеосигналу для безпілотників</i>	<i>04.03.2025р. – 01.05.2025р.</i>	<i>Викон.</i>
4.	<i>Робота над основним розділом програмування та тестування комп'ютерної системи оцифрування потокового відеосигналу для безпілотників</i>	<i>02.05.2025р. – 01.06.2025р.</i>	<i>Викон.</i>
5.	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>02.06.2025р. – 06.06.2025р.</i>	<i>Викон.</i>
6.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>06.06.2025р. – 09.06.2025р.</i>	<i>Викон.</i>
7.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>9.06.2025р. – 15.06.2025р.</i>	
8.	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>16.06.2025р. – 22.06.2025р.</i>	<i>Викон.</i>
9.	<i>Захист кваліфікаційної роботи бакалавра</i>	<i>26.06.2025р.</i>	<i>Викон.</i>

Студент

(підпис)

Коношенко Я. А.

(прізвище та ініціали)

Керівник роботи

(підпис)

доцент, к.т.н. Лецишин Ю. З.

(прізвище та ініціали)

АНОТАЦІЯ

Коношенко Я.А. Комп'ютерна система оцифрування потокового відеосигналу для безпілотників: робота на здобуття кваліфікаційного ступеня бакалавра: спец. 123 — комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2025.

Ключові слова: комп'ютерна система, передача даних, оцифрування, відеопотік, безпілотники.

Кваліфікаційна робота присвячена дослідженню можливості отримання оцифрованого відео з систем моніторингу безпілотниками. Робота охоплює аналіз можливих рішень, що відповідають сучасним стандартам передачі даних та кріпленні на безпілотниках. В розробці використовуються друкувана плата, обчислювальний модуль Raspberry Pi 4 та мікрокомп'ютер STM32. Для отримання даних, а саме кадрів, використовується під'єднана камера.

Для отримання даних для оцифрування система під'єднується по мережі завдяки SSH до персонального комп'ютера. Кадри оцифровуються у порядку черги за системою FIFO (First In First Out). Дані конвертуються у потрібний колірний режим та виводяться через додаток VLC. В результаті отримується потоковий відеосигнал, який передає дані в режимі реального часу. Результати цього дослідження можуть мати застосування у системах моніторингу, що використовують безпілотники.

ANNOTATION

Konoshenko Y.A. Computerized system for digitizing streaming video signals for drones: Bachelor's Graduation Thesis: speciality 123 — computer engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2025.

Keywords: computer system, data transmission, digitization, video stream, drones.

The qualification work is devoted to the study of the possibility of obtaining digitized video from monitoring systems by drones. The work includes an analysis of possible solutions that meet modern standards for data transmission and mounting on drones. The development uses a printed circuit board, a Raspberry Pi 4 computing module, and an STM32 microcomputer. A connected camera is used to obtain data, namely frames.

To obtain data for digitization, the system is connected via SSH to a personal computer. The frames are digitized on a first-come, first-served basis using the FIFO (First In First Out) system. The data is converted to the desired color mode and displayed via the VLC application. The result is a streaming video signal that transmits data in real time. The results of this study can be used in monitoring systems that use drones.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ НА РОЗРОБКУ КОМП'ЮТЕРНОЇ СИСТЕМИ ОЦИФРУВАННЯ ПОТОКОВОГО ВІДЕОСИГНАЛУ	11
1.1 Аналіз предметної області	11
1.2 Аналіз вимог до комп'ютерної системи	13
1.3 Аналіз можливих рішень поставленого завдання	16
РОЗДІЛ 2 ПРОЄКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ОЦИФРУВАННЯ ПОТОКОВОГО ВІДЕОСИГНАЛУ	22
2.1 Розробка узагальненої структури комп'ютерної системи	22
2.2 Обґрунтування вибору апаратного забезпечення проектного комп'ютерного пристрою	25
2.3 Обґрунтування вибору програмного забезпечення проектного комп'ютерного пристрою	29
2.3.1 Апаратна частина	29
2.3.2 Програмна частина	33
2.4 Проектування комп'ютерного пристрою	35
РОЗДІЛ 3 ПРОГРАМУВАННЯ ТА ТЕСТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ОЦИФРУВАННЯ ПОТОКОВОГО ВІДЕОСИГНАЛУ	44
3.1 Прошивання Compute Module Raspberry Pi 4	44
3.2 Розробка програмного коду для STM32	47
3.3 Тестування	53

					КС КРБ 123.211.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	Комп'ютерна система оцифрування потокового відеосигналу для безпілотників		
Розроб.		Коношенко Я.А.					
Перевір.		Лецишин Ю.З.					
Реценз.		Цуприк Г.Б.					
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркушів
						6	
					ТНТУ, каф. КС, гр. СІ-41		

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.....57

4.1 Долікарська допомога при ураженні електричним струмом57

4.2 Естетичне оформлення та ергономічне дослідження робочого місця оператора.....59

ВИСНОВКИ.....62

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....63

Додаток А Технічне завдання

Додаток Б Лістинг програми

Додаток В Перелік елементів

					КС КРБ 123.211.00.00 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК СКОРОЧЕНЬ

АПП – Автоматичне регулювання підсилення

ARM – Advanced RISC Machine (або Acorn RISC Machine)

BNC – Bayonet Neill-Concelman

CM – Compute Module

eMMC – embedded MultiMediaCard

EMI – Electromagnetic Interference

FIFO – First In, First Out

GPIO – General Purpose Input/Output

GPS – Global Positioning System

JTAG – Joint Test Action Group

NEON SIMD – NEON Single Instruction, Multiple Data (архітектура розширення ARM)

NTSC – National Television System Committee

NVMe – Non-Volatile Memory Express

PAL – Phase Alternating Line

PCB – Printed Circute Board

RCA – Radio Corporation of America

RPi – Raspberry Pi

SD – Secure Digital

SECAM – Sequential Color with Memory

SIMD – Single Instruction, Multiple Data

SSH – Secure Shell

					КС КРБ 123.211.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

TCP – Transmission Control Protocol

TSP – Real Time Streaming Protocol

TTX – тактико-технічні характеристики

UART – Universal Asynchronous Receiver-Transmitter

UDP – User Datagram Protocol

USB – Universal Serial Bus

V4L2 – Video4Linux version 2

WiFi – Wireless Fidelity

					КС КРБ 123.211.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У сучасному світі технологій безпілотні літальні апарати перетворилися з експериментальних пристроїв на невід'ємну частину багатьох сфер людської діяльності. Від сільськогосподарського моніторингу до оборонних операцій, від пошуково-рятувальних місій до кінематографічних зйомок – БПЛА революціонували підхід до отримання та аналізу візуальної інформації з повітря. Однак за цим технологічним прогресом йде складна інженерна задача: як ефективно оцифрувати, обробити та передати високоякісний відеосигнал в умовах жорстких обмежень авіаційного середовища.

Сьогодні, коли кожен кадр відео може нести критично важливу інформацію – чи то координати для рятувальної операції, чи дані для точного землеробства, чи розвідувальні відомості – якість оцифрування відеосигналу стає питанням не просто технічної досконалості, але й реальної ефективності застосування безпілотних систем. Традиційні підходи до оцифрування, розроблені для стаціонарних умов, виявляються недостатніми для динамічного, енергообмеженого та вібраційно навантаженого середовища польоту БПЛА.

Саме це протиріччя між зростаючими вимогами до якості відеоданих та специфічними обмеженнями бортового обладнання визначає актуальність даного дослідження.

Мета кваліфікаційної роботи полягає в розробці комп'ютерної системи оцифрування потокового відеосигналу, яка не просто адаптована до умов роботи на безпілотнику, але й оптимізована для максимальної ефективності в цих умовах.

					КС КРБ 123.211.00.00 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ НА РОЗРОБКУ КОМП'ЮТЕРНОЇ СИСТЕМИ ОЦИФРУВАННЯ ПОТОКОВОГО ВІДЕОСИГНАЛУ

1.1 Аналіз предметної області

Комп'ютерна система для оцифрування потокового відеосигналу для безпілотників слугує важливим мостом між застарілим аналоговим відео обладнанням і сучасною цифровою інфраструктурою.

Дана система перетворює аналогові відеосигнал в цифровий відео формат в цифровий, забезпечуючи покращену обробку, зберігання, передачу та інтеграцію з сучасними системами управління безпілотниками та даними.

Метою даної системи є конвертація аналогового або чистого відеосигналу в камерах, що знаходяться на дронах, в зжатиї оцифрований формат, який підходить для передачі даних з мінімальною затримкою на наземне базування або ж хмарний сервер для використання в режимі реального часу.

Для початку реалізації потрібно розуміти, в яких сферах можливо застосовувати дану систему.

Одною зі сфер застосування даної системи є військова та оборонна. На теперішній час дуже багато військового обладнання використовує аналогові відео системи, що потребують інтеграції з сучасними цифровими командними центрами.

Також, варто додати, що при використанні перетворення аналогових сигналів можливо шифрувати сигнал, що є суттєвою перевагою на полі бою чи передачі конфіденційних даних, які можливо передати безпечними

					КС КРБ 123.211.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата	Аналіз технічного завдання на розробку комп'ютерної системи оцифрування потокового відеосигналу			Літ.	Арк.	Аркушів	
Розроб.		Коношенко Я.А.									
Перевір.		Лецишин Ю.З.								11	
Реценз.		Цуприк Г.Б.						ТНТУ, каф. КС, гр. СІ-41			
Н. Контр.		Луцик Н.С.									
Затверд.		Осухівська Г.М.									

протоколами передачі [2].

При передачі даних важливо також взяти до уваги багатоплатформну сумісність. Оцифрування потокового відеосигналу дозволяє аналоговим каналам безпілотників працювати з цифровими наземними станціями управління та тактичними дисплеями. В режимі реального часу таким чином можливо проводити розвідку, через те, що забезпечується негайна цифрова обробка даних для ідентифікації цілей та оцінки загроз [3].

Поза військовим призначенням варто розглянути комерційне (цивільне) та промислове застосування. Наразі безпілотники використовуються як моніторингові системи – таким чином можливе забезпечення огляду територій та миттєва передача даних з місць, які не оглядаються звичайною камерою (див. рис. 1.1). Нафтові вишки, лінії електропередач та промислові об'єкти часто використовують старі аналогові системи камер, які значно виграють від цифрового перетворення.



Рисунок 1.1 – Дрон для охорони та моніторингу Reactive Drone RDM3

Також комп'ютерна система для оцифрування потокового відеосигналу допоможе службам швидкого реагування під час пошуково-рятувальних операцій. Перетворення аналогових каналів передачі дозволяє координувати

					КС КРБ 123.211.00.00 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

роботу з цифровими системами реагування на надзвичайні ситуації.

Якщо відійти до більш примітивних структур, можливо звернутися до сільськогосподарських структур. Застарілі сільськогосподарські дрони з аналоговими камерами можна модернізувати для роботи з сучасним програмним забезпеченням для управління фермерськими господарствами.

На останок з цивільних призначень системи можливо виділити безпеку та спостереження. Існуючі аналогові системи безпеки можуть бути вдосконалені за допомогою можливостей цифрової обробки.

1.2 Аналіз вимог до комп'ютерної системи

Розробка комп'ютерної системи для оцифрування потокового відео для дронів вимагає всебічного розуміння як технічних специфікацій, так і експлуатаційних обмежень. Ця система повинна безперешкодно долати розрив між аналоговими відеотехнологіями та сучасною цифровою інфраструктурою, працюючи в складному середовищі безпілотних літальних апаратів [4].

Складність операцій з дронами вимагає, щоб будь-яка бортова система відповідала суворим вимогам щодо розміру, ваги, енергоспоживання та стійкості до впливу навколишнього середовища, одночасно забезпечуючи високоякісну обробку відео. Ці вимоги повинні бути збалансовані з міркуваннями вартості і термінами розробки, що робить вибір відповідних апаратних платформ і підходів до проектування критично важливими для успіху проєкту.

Система повинна підтримувати широкий спектр форматів вхідного аналогового відеосигналу, щоб забезпечити сумісність з існуючими системами камер безпілотників. Основною вимогою є підтримка стандартних композитних відеосигналів, включаючи формати NTSC (National Television System Committee) і PAL (Phase Alternating Line), причому NTSC підтримує

					КС КРБ 123.211.00.00 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

525 рядків при 29,97 кадрах в секунду, а PAL - 625 рядків при 25 кадрах в секунду. Крім того, система повинна підтримувати формат SECAM (Sequential Color with Memory - послідовний колір з пам'яттю) для ширшої міжнародної сумісності.

Вимоги до якості вхідного сигналу вимагають, щоб система обробляла відеосигнали з амплітудою від 0,7 В до 1,4 В від піку до піку, що є типовим для стандартів композитного відео. Система повинна підтримувати цілісність сигналу в цьому діапазоні, забезпечуючи при цьому автоматичне регулювання підсилення для компенсації коливань сигналу, які можуть виникати через довжину кабелю, зміни температури або старіння компонентів. Вхідний опір повинен бути належним чином підібраний на рівні 75 Ом, щоб запобігти віддзеркаленню сигналу та зберегти якість відео.

Підсистема захоплення повинна підтримувати кілька одночасних вхідних каналів, з мінімальною вимогою до двох каналів для стерео або багатокутових сценаріїв захоплення. Кожен канал повинен мати можливість незалежного налаштування для різних форматів відео та параметрів обробки. Система повинна забезпечувати можливість зйомки в реальному часі з покадровою синхронізацією для запобігання часових артефактів в оцифрованому вихідному сигналі.

Основна функціональність аналого-цифрового перетворення повинна відповідати суворим критеріям продуктивності, щоб забезпечити якість відео, придатну для оперативного використання. Аналого-цифровий перетворювач повинен здійснювати дискретизацію з достатньою частотою, щоб захопити всю смугу вхідного сигналу, що вимагає частоти дискретизації не менше 27 МГц для відео стандартної роздільної здатності. Перетворювач повинен забезпечувати мінімум 8-бітну роздільну здатність для кожного кольорового каналу, при цьому бажано використовувати 10-бітну роздільну здатність для підвищення точності передачі кольору та зменшення шуму квантування [2].

Можливості перетворення кольорового простору повинні підтримувати

					КС КРБ 123.211.00.00 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

перетворення між різними форматами, включаючи колірні простори YUV, RGB і YCbCr. Система повинна реалізовувати належну гамма-корекцію та перетворення колірної матриці, щоб забезпечити точне відтворення кольорів на різних пристроях відображення та системах запису. Повинні бути реалізовані алгоритми автоматичного балансу білого та кольорокорекції для компенсації варіацій характеристик аналогового сигналу.

Конвеєр обробки повинен підтримувати роботу в режимі реального часу з максимальною затримкою не більше 100 мілісекунд від аналогового входу до цифрового виходу. Ця вимога до низької затримки є критично важливою для додатків, що вимагають негайного візуального зворотного зв'язку, таких як ручне пілотування дронів або операції спостереження в режимі реального часу. Система повинна підтримувати постійну частоту кадрів без пропусків кадрів під час нормальної роботи, навіть при різних навантаженнях на обробку.

Можливості цифрового виводу повинні підтримувати кілька стандартних форматів для забезпечення сумісності з різними системами наземних станцій та обладнанням для запису. Система повинна забезпечувати можливість кодування у форматі H.264 як мінімальну вимогу, а кодування у форматі H.265 (HEVC) є кращим для покращення ефективності стиснення та використання смуги пропускання. Вихід сирого відео в нестиснених форматах, таких як YUV420 або RGB24, також повинен бути доступний для додатків, що вимагають максимальної якості зображення.

Можливості потокового передавання повинні підтримувати декілька мережних протоколів, включаючи протокол потокового передавання в реальному часі (RTSP), протокол користувацьких дейтаграм (UDP) і протокол керування передаванням (TCP). Система повинна реалізовувати потокове передавання з адаптивним бітрейтом для автоматичного регулювання якості відео залежно від доступної пропускну здатності мережі та якості з'єднання. Можливості багатоадресного потокового передавання необхідні для

					КС КРБ 123.211.00.00 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

підтримки одночасної передачі на кілька наземних станцій без дублювання накладних витрат на обробку.

Підтримка роздільної здатності відео повинна включати стандартну роздільну здатність (720x480 NTSC, 720x576 PAL) як мінімальні вимоги, з можливістю масштабування до форматів високої роздільної здатності, таких як 720p і 1080p. Має підтримуватися перетворення частоти кадрів, що дозволяє виводити відео з частотою, відмінною від вхідного сигналу, з належною часовою інтерполяцією для збереження плавності відтворення руху.

1.3 Аналіз можливих рішень поставленого завдання

На основі попередньо прописаних вимог до функціональних вимог, можливо визначити найбільш оптимальну базу, що буде служити як основа даної системи.

Обчислювальний модуль Raspberry Pi 4 (див. рис. 1.2) слугує основною обчислювальною платформою, що вимагає певних параметрів конфігурації для задоволення вимог до продуктивності системи. Модуль повинен мати мінімум 4 ГБ оперативної пам'яті, щоб підтримувати одночасну обробку відео, потокове передавання та системні операції. Конфігурація з 8 ГБ оперативної пам'яті є кращою для додатків, що вимагають великої буферизації або декількох одночасних відеопотоків [5, 6].



Рисунок 1.2 – Compute module Raspberry Pi 4

					КС КРБ 123.211.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Вимоги до пам'яті передбачають використання високошвидкісного накопичувача eMMC або NVMe ємністю не менше 32 ГБ для операційної системи, додатків і тимчасової буферизації відео. Підсистема зберігання даних повинна підтримувати стабільну швидкість запису щонайменше 50 МБ/с, щоб задовольнити вимоги до високоякісного відеозапису. Для розширення можливостей запису та передачі даних слід передбачити додатковий знімний накопичувач на SD-карті або USB-інтерфейси.

Процесорна підсистема повинна надійно працювати в різних температурних умовах, характерних для середовища безпілота. Вимоги до управління тепловим режимом включають належну конструкцію розсіювання тепла для підтримки температури процесора нижче 70°C під час безперервної роботи. Система повинна реалізовувати механізми теплового дроселювання для запобігання перегріву при збереженні прийняттого рівня продуктивності.

Аналогові вхідні відео інтерфейси повинні забезпечувати підключення професійного рівня з належним узгодженням імпедансу та обробкою сигналу. Система потребує щонайменше двох композитних відеовходів, реалізованих через високоякісні роз'єми BNC або RCA, залежно від вимог застосування. Вхідні ланцюги повинні включати з'єднання змінного струму, затискачі та фільтрацію для забезпечення чистого захоплення сигналу за різних умов експлуатації.

Цифрові вихідні інтерфейси повинні підтримувати як дротове, так і бездротове підключення. Підключення Ethernet є обов'язковим для високошвидкісних додатків, що вимагають можливості Gigabit Ethernet для підтримки декількох відеопотоків високої чіткості. Підключення WiFi повинно підтримувати стандарт 802.11ac або новіші стандарти з дводіапазонною роботою для гнучкості в різних радіочастотних середовищах [3].

Інтерфейси USB необхідні для підключення периферійних пристроїв і

					КС КРБ 123.211.00.00 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

зовнішніх накопичувачів. Мінімум два порти USB 3.0 повинні бути доступні для високошвидкісної передачі даних і підключення пристроїв. Інтерфейси GPIO повинні бути доступними для інтеграції з системами керування дронами та зовнішніми датчиками, для чого потрібно щонайменше 20 конфігурованих цифрових контактів вводу/виводу.

Характеристики енергоспоживання є критично важливими для інтеграції дронів, де час автономної роботи безпосередньо впливає на тривалість місії. Система повинна працювати в межах бюджету енергоспоживання не більше 15 Вт, при цьому типове робоче енергоспоживання не повинно перевищувати 10 Вт при виконанні стандартних операцій обробки відео. Система повинна підтримувати діапазони вхідної напруги від 5 до 28 В постійного струму, щоб відповідати різним системам живлення дронів.

Функції керування живленням повинні включати інтелектуальне масштабування потужності на основі вимог до обробки, з можливістю зниження енергоспоживання в періоди простою або при обробці відеопотоків з низькою роздільною здатністю. Система повинна реалізовувати процедури плавного вимкнення для запобігання пошкодженню даних під час несподіваної втрати живлення, що вимагає резервного живлення або підтримки суперконденсатора для критичних операцій вимкнення.

Регулювання напруги повинно підтримувати стабільну роботу в зазначеному діапазоні вхідної напруги, а рівень пульсацій і шумів не повинен перевищувати 50 мВ від піку до піку. Для забезпечення надійного запуску та роботи системи за різних умов електроживлення необхідні схеми скидання та виявлення зникнення живлення.

Обмеження за розміром і вагою мають першорядне значення для інтеграції дронів, де кожен грам і кубічний сантиметр впливає на польотні характеристики і час автономної роботи. Загальна вага всієї системи, включаючи корпус і роз'єми, не повинна перевищувати 200 грамів. Фізичні

					КС КРБ 123.211.00.00 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

розміри повинні відповідати максимальним габаритам 120 мм x 80 мм x 30 мм, щоб забезпечити можливість інтеграції в типові відсіки корисного навантаження безпілота.

Вимоги до механічного кріплення повинні підтримувати стандартні методи кріплення безпілота, включаючи точки кріплення з різьбленням М3 і М4 з відповідною відстанню для віброізоляційних кріпильних систем. Корпус повинен забезпечувати захист від електромагнітних завад (ЕМІ), щоб запобігти перешкодам для систем керування дронами та GPS-приймачів, що вимагає відповідного електропровідного покриття або металеві конструкції корпусу.

Специфікації роз'ємів повинні передбачати використання роз'ємів авіаційного або військового стандарту для критично важливих інтерфейсів, щоб забезпечити надійне з'єднання в умовах вібрації та впливу навколишнього середовища. Всі зовнішні з'єднання повинні бути належним чином захищені від потрапляння вологи, зберігаючи при цьому електричні характеристики.

Можливості обробки в реальному часі повинні гарантувати, що операції перетворення та потокової передачі відео не призводять до неприйнятних затримок для оперативного використання. Максимальна наскрізна затримка від аналогового входу до мережевого виходу не повинна перевищувати 100 мілісекунд за нормальних умов роботи. Стабільність частоти кадрів повинна підтримувати вхідну частоту кадрів з менш ніж 0,1% падінням кадрів протягом безперервного 24-годинного періоду роботи.

Вимоги до одночасної роботи визначають, що система повинна підтримувати одночасну обробку декількох відеоканалів зі збереженням специфікацій продуктивності для кожного каналу. Система повинна підтримувати щонайменше два одночасних відеопотоки з незалежними параметрами обробки та напрямками виводу.

Продуктивність мережевого потокового передавання повинна підтримувати кілька одночасних потоків з адаптивним контролем якості на

					КС КРБ 123.211.00.00 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

основі доступної пропускної здатності. Система повинна автоматично регулювати параметри стиснення для підтримки прийнятної якості відео, запобігаючи перевантаженню мережі або переповненню буфера.

Програмна платформа повинна базуватися на дистрибутиві Linux, оптимізованому для вбудованих систем та обробки відео в реальному часі. Операційна система повинна підтримувати функції апаратного прискорення Raspberry Pi CM4, включаючи можливості кодування та декодування відео на графічному процесорі. Повинні бути доступні можливості планування в реальному часі для забезпечення стабільної продуктивності обробки відео при різних навантаженнях на систему.

Підтримка драйверів пристроїв повинна включати комплексні можливості апаратного інтерфейсу для пристроїв відеозахоплення, мережевих інтерфейсів та периферійних підключень. Система повинна підтримувати стандартні інтерфейси Video4Linux (V4L2) для пристроїв відео захоплення та стандартні мережеві протоколи для потокової передачі даних.

Вимоги безпеки вимагають, щоб операційна система реалізовувала відповідні заходи безпеки, включаючи можливості безпечного завантаження, зашифровані варіанти зберігання даних та мережеві протоколи безпеки. Для запобігання несанкціонованому доступу до системи або зміні конфігурації необхідно впровадити автентифікацію користувачів та контроль доступу.

Вимоги, викладені вище, створюють всеосяжну основу для розробки надійної та ефективної комп'ютерної системи для оцифрування потокового відео з дронів. Ці вимоги балансують між вимогливим робочим середовищем платформ безпілотників та необхідністю якісної обробки відео і надійної роботи системи. Специфікація Raspberry Pi CM4 як обчислювальної платформи забезпечує оптимальне поєднання обчислювальних можливостей, енергоефективності та економічної ефективності при дотриманні жорстких обмежень щодо розміру та ваги для інтеграції з безпілотниками.

Успішне виконання цих вимог призведе до створення системи, здатної

					КС КРБ 123.211.00.00 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

безперешкодно поєднувати аналогові та цифрові відеотехнології, забезпечуючи при цьому надійність і продуктивність, необхідні для критично важливих операційних додатків. Комплексний характер цих вимог гарантує, що всі аспекти проектування системи, від вибору обладнання до реалізації програмного забезпечення, будуть належним чином розглянуті та визначені для досягнення цілей проекту в рамках реалістичних обмежень за вартістю та графіком.

					КС КРБ 123.211.00.00 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЄКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ОЦИФРУВАННЯ ПОТОКОВОГО ВІДЕОСИГНАЛУ

2.1 Розробка узагальненої структури комп'ютерної системи

Розробка комп'ютерної системи для оцифрування потокового відео в додатках для безпілотників є складним інженерним завданням, яке вимагає ретельного розгляду апаратної архітектури, програмної реалізації та системної інтеграції. Узагальнена структура цієї системи базується на стратегічному поєднанні обчислювального модуля Raspberry Pi Compute Module 4 (CM4) як основного процесора та спеціально розробленої друкованої плати (PCB), яка забезпечує спеціалізовані можливості захоплення аналогового відео та інтерфейси, специфічні для дронів. Такий архітектурний підхід забезпечує баланс між вимогами до продуктивності, вартісними обмеженнями, складністю розробки та експлуатаційною надійністю, зберігаючи при цьому гнучкість, необхідну для різноманітних сценаріїв розгортання [5, 6].

Узагальнена структура системи оцифрування відео з дронів має модульну архітектуру, яка розділяє завдання між обчислювальними функціями загального призначення та спеціалізованими вимогами до обробки аналогового відео. Таке розділення дозволяє оптимізувати кожну підсистему незалежно, зберігаючи при цьому чіткі інтерфейси між компонентами. Архітектура реалізує конвеєрний підхід до обробки відео, коли аналогові сигнали проходять через спеціальне обладнання для захоплення, а потім обробляються обчислювальною платформою загального призначення і передаються через стандартні мережеві інтерфейси.

Структура системи складається з трьох основних функціональних блоків: аналогового інтерфейсу обробки, ядра цифрової обробки та

					КС КРБ 123.211.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Коношенко Я.А.			Проектування комп'ютерної системи оцифрування потокового відеосигналу			Літ.	Арк.	Аркушів	
Перевір.		Лецишин Ю.З.								22	
Реценз.		Цуприк Г.Б						ТНТУ, каф. КС, гр. СІ-41			
Н. Контр.		Луцик Н.С.									
Затверд.		Осухівська Г.М.									

комунікаційного інтерфейсу. Аналоговий інтерфейс, реалізований на спеціальній друкованій платі, відповідає за формування сигналу, аналого-цифрове перетворення та початкову цифрову обробку відеосигналів. Ядро цифрової обробки, зосереджене на Raspberry Pi CM4, виконує високорівневу обробку відео, кодування, потокове передавання та функції управління системою. Інтерфейс зв'язку забезпечує дротовий і бездротовий зв'язок для потокової передачі відео, керування системою та інтеграції з системами керування польотом дронів.

Модульний підхід дає кілька важливих переваг для застосування безпілотників. По-перше, він уможливорює незалежну оптимізацію енергоспоживання для кожної підсистеми, що дозволяє системі впроваджувати інтелектуальні стратегії управління живленням, які можуть збільшити час польоту. По-друге, розділення аналогової і цифрової обробки зменшує електромагнітні перешкоди між чутливими аналоговими схемами і високошвидкісними цифровими комутаційними схемами. По-третє, модульна структура полегшує обслуговування та модернізацію, дозволяючи замінювати або модернізувати окремі підсистеми, не впливаючи на всю систему [7].

Спеціальна друкована плата слугує основою архітектури системи, забезпечуючи спеціалізовані інтерфейси та можливості обробки сигналів, недоступні на стандартних обчислювальних платформах. Друкована плата має багатошарову конструкцію, яка ретельно розділяє аналогову та цифрову частини схеми, щоб мінімізувати перешкоди та зберегти цілісність сигналу. Аналогова секція містить високоякісні схеми відеозахоплення з належним узгодженням імпедансу, формуванням сигналу та фільтрацією шуму для забезпечення оптимальної якості аналого-цифрового перетворення.

Конструкція друкованої плати включає спеціальні схеми керування живленням, які забезпечують чисте, регульоване живлення чутливих аналогових компонентів, зберігаючи при цьому ефективність для безпілотників, що живляться від акумуляторів. Схеми імпульсних регуляторів

					КС КРБ 123.211.00.00 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечують різні рівні напруги, необхідні для різних підсистем, мінімізуючи при цьому енергоспоживання і тепловиділення. Плата має комплексні схеми захисту, включаючи захист від перенапруги, захист від неправильної полярності та захист від електростатичних розрядів, щоб забезпечити надійну роботу в суворих умовах експлуатації безпілота.

Інтерфейсні схеми на спеціальній друкованій платі забезпечують необхідний зв'язок між Raspberry Pi CM4 та зовнішніми системами, включаючи контролери польоту дронів, GPS-приймачі та телеметричні системи. Ці інтерфейси реалізують відповідне перетворення рівня сигналу, ізоляцію та фільтрацію для забезпечення надійного зв'язку, запобігаючи втручанню в роботу критично важливих систем дрона. Друкована плата також містить індикатори стану, контрольні точки та діагностичні інтерфейси, які полегшують налагодження та технічне обслуговування системи.

Механічна конструкція друкованої плати враховує унікальні вимоги до інтеграції безпілота, включаючи вібростійкість, терморегуляцію та електромагнітну сумісність. Плата використовує відповідні матеріали та технології виготовлення, щоб витримувати механічні навантаження, пов'язані з роботою безпілота, зберігаючи при цьому електричні характеристики. При виборі компонентів акцент робиться на автомобільні або військові деталі, щоб забезпечити надійність в екстремальних умовах навколишнього середовища.

На основі вище описаного можливо розробити структурну схему системи (див. рис. 2.1). На ній зображений основний принцип роботи комп'ютерної системи для оцифрування потокового відосигналу для безпілотників.

					КС КРБ 123.211.00.00 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

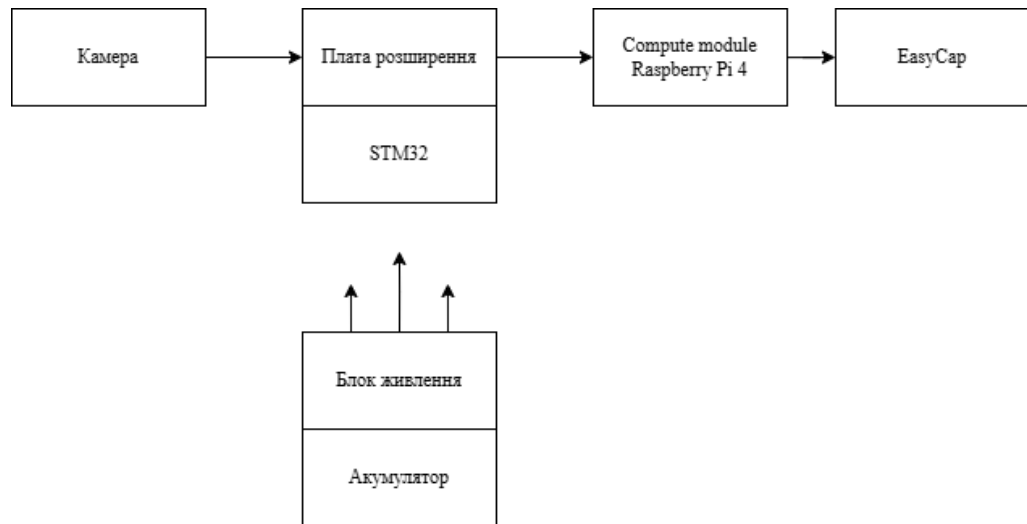


Рисунок 2.1 – Структурна схема комп’ютерної системи

Дана структурна схема містить в собі блок живлення, що підключається до блока розширення та заживлює всю систему. Блок розширення знаходиться напряду над платою обчислювального модуля та дозволяє підключити Ethernet та камеру за допомогою шлейфів. Блок з STM32, яка містить в собі код, необхідний для оцифрування, знаходиться прям над блоком розширення, не конфліктуючи з Raspberry Pi. Камера підключена як сторонній модуль, що підключається до основної системи [8].

2.2 Обґрунтування вибору апаратного забезпечення проектованого комп’ютерного пристрою

Вибір Raspberry Pi Compute Module 4 як основної обчислювальної платформи - це ретельно продуманий баланс між вимогами до продуктивності, обмеженнями енергоспоживання та складністю розробки. Чотириядерний процесор ARM Cortex-A72 CM4, що працює на частоті 1,5 ГГц, забезпечує достатню обчислювальну потужність для обробки відео в реальному часі, зберігаючи при цьому рівень енергоспоживання, сумісний з акумуляторними системами дронів. Архітектура процесора включає в себе розширені функції, такі як позачергове виконання, передбачення розгалужень та комплексна

					КС КРБ 123.211.00.00 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

підтримка операцій з плаваючою комою, які забезпечують ефективну роботу алгоритмів обробки відео.

Нижче наведена таблиця 2.1 з ТТХ (тактико-технічні характеристики) обчислювального модуля Raspberry Pi 4 [6].

Таблиця 2.1 – Технічні характеристики обчислювального модуля
Raspberry Pi 4

Назва	Опис
Процесор	Broadcom BCM2711, чотирьохядерний Cortex-A72 (ARM v8), 64-біт, 1,5 ГГц
SDRAM	1 ГБ, 2 ГБ, 4 ГБ або 8 ГБ LPDDR4-3200
eMMC	0 ГБ ("Lite"), 8 ГБ, 16 ГБ або 32 ГБ
Можливості підключення	вбудований Gigabit Ethernet PHY з підтримкою IEEE1588, 1 інтерфейс USB 2.0, 1 інтерфейс PCIe Gen 2, GPIO інтерфейс, інтерфейс SD-карти для SD-карти або зовнішнього eMMC (для використання тільки з варіантами Compute Module 4 без eMMC)
Мультимедіа	H.265 (декодування 4Kp60); H.264 (декодування 1080p60, кодування 1080p30); Графіка OpenGL ES 3.0
Живлення	5В постійного струму
Діапазон робочих температур	від -20°C до +85°C
Габаритні розміри	55 мм × 40 мм

Інтегрований графічний процесор VideoCore VI забезпечує критично важливе апаратне прискорення операцій кодування та декодування відео. Таке графічне прискорення необхідне для досягнення продуктивності в реальному часі при перетворенні та стисненні відеоформатів, мінімізуючи при цьому навантаження на процесор та енергоспоживання. Можливості апаратного прискорення підтримують декілька відеокодеків, включаючи H.264 і H.265, що дозволяє системі адаптуватися до різних вимог до пропускної здатності та

якості, не вимагаючи програмного кодування, яке споживало б надмірні обчислювальні ресурси.

Варіанти конфігурації пам'яті від 1 ГБ до 8 ГБ забезпечують гнучкість для різних вимог додатків, зберігаючи при цьому економічну ефективність. Конфігурації з 4 ГБ і 8 ГБ забезпечують достатній обсяг пам'яті для складних операцій обробки відео, включаючи буферизацію кадрів, перетворення форматів і роботу з декількома одночасними відеопотоками. Уніфікована архітектура пам'яті дозволяє ефективно розподіляти ресурси пам'яті між операціями CPU та GPU, зменшуючи вимоги до пропускної здатності пам'яті та підвищуючи загальну продуктивність системи [9].

Форм-фактор Compute Module 4 забезпечує значні переваги для інтеграції в безпілотники порівняно з традиційними одноплатними комп'ютерами. Компактні розміри 55 мм x 40 мм мінімізують вимоги до простору, а стандартизований формат роз'єму SO-DIMM забезпечує надійне високошвидкісне підключення до спеціальних плат-носіїв. Такий підхід усуває проблеми надійності, пов'язані з безліччю окремих роз'ємів, забезпечуючи доступ до всіх інтерфейсів процесора через єдину точку підключення.

CM4 надає широкі можливості підключення, включаючи кілька високошвидкісних послідовних інтерфейсів, шини I2C і SPI, виводи GPIO і спеціальні інтерфейси камер, які полегшують інтеграцію з системами безпілотників і зовнішніми периферійними пристроями. Наявність двох спеціальних послідовних інтерфейсів для камер (CSI) дозволяє підключати додаткові системи камер або сенсорні модулі, не використовуючи пропускну здатність USB і не вимагаючи додаткового інтерфейсного обладнання.

Інтерфейси USB 3.0 і Gigabit Ethernet забезпечують високошвидкісне підключення для потокового відео і передачі даних. Вбудовані модулі WiFi і Bluetooth забезпечують бездротове з'єднання для управління системою і передачі телеметрії з низькою пропускну здатністю. Ці можливості

					КС КРБ 123.211.00.00 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

підключення забезпечують гнучкість, необхідну для адаптації системи до різних платформ дронів та експлуатаційних вимог без необхідності модифікації апаратного забезпечення [10].

Архітектура процесора ARM забезпечує виняткову продуктивність на ват, що є критично важливим для безпілотників з живленням від акумуляторів. 28-нм технологічний процес і вдосконалені функції керування живленням дозволяють процесору ефективно працювати в широкому діапазоні рівнів продуктивності, автоматично масштабуючи тактову частоту і рівень напруги залежно від вимог до обробки. Ця можливість динамічного масштабування живлення подовжує час роботи від батареї, забезпечуючи при цьому достатню продуктивність для вимогливих операцій з обробки відео.

Теплові характеристики CM4 добре підходять для застосування в безпілотниках, де активні системи охолодження зазвичай недоступні. Процесор генерує мінімальне тепло під час нормальної роботи і реалізує механізми теплового захисту, які запобігають пошкодженню в умовах перегріву. Компактний форм-фактор і низька теплова маса забезпечують швидку теплову реакцію на зміну умов навколишнього середовища, що важливо для роботи безпілотника, яка може бути пов'язана зі швидкими змінами висоти і температури [11].

Комплексні можливості керування живленням включають підтримку різних режимів сну та вимкнення живлення, що дозволяє значно подовжити термін служби батареї в періоди зниженої активності. Система може бути налаштована на вимкнення другорядних підсистем на етапах крейсерського польоту, зберігаючи при цьому критично важливі функції обробки відео, оптимізуючи загальне енергоспоживання системи для забезпечення максимальної тривалості польоту.

Екосистема Raspberry Pi надає широкі інструменти розробки, документацію та підтримку спільноти, які значно скорочують час розробки та зменшують ризики порівняно з альтернативними вбудованими платформами.

					КС КРБ 123.211.00.00 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

Наявність комплексних наборів для розробки програмного забезпечення, інструментів для крос-компіляції та інтерфейсів для налагодження дозволяє ефективно розробляти програмне забезпечення та оптимізувати систему. Велика спільнота розробників та широкі онлайн-ресурси надають цінну підтримку для усунення несправностей та оптимізації.

Апаратна сумісність зі стандартними інструментами та інтерфейсами розробки спрощує процес розробки та зменшує потребу в спеціалізованому обладнанні. Наявність стандартних налагоджувальних інтерфейсів, включаючи JTAG та UART, дозволяє проводити комплексне налагодження системи та аналіз продуктивності за допомогою широкодоступних інструментів. Така сумісність знижує витрати на розробку і дозволяє швидко створювати та тестувати прототипи [12].

Налагоджений ланцюжок поставок і виробнича екосистема, що оточує продукти Raspberry Pi, забезпечує довгострокову доступність і стабільну ціну, що є важливими факторами для розробки комерційних продуктів. Масштабне виробництво та глобальна мережа дистрибуції забезпечують постійну доступність та конкурентоспроможні ціни протягом усього життєвого циклу продукту.

2.3 Обґрунтування вибору програмного забезпечення проектного комп'ютерного пристрою

2.3.1 Апаратна частина

Вибір мови C як основної мови програмування для системи оцифрування відео з дронів зумовлений вимогами до продуктивності в реальному часі, притаманними програмам обробки відео. Перетворення та потокова передача відео в реальному часі вимагають детермінованого часу виконання та мінімальної затримки обробки, що найкраще досягається за допомогою низькорівневих мов програмування, які забезпечують прямий

					КС КРБ 123.211.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

доступ до апаратного забезпечення та передбачувані характеристики продуктивності.

Мова С надає програмісту прямий контроль над керуванням пам'яттю, завантаженням процесора та апаратними інтерфейсами, що дозволяє оптимізувати критичні шляхи коду для досягнення максимальної продуктивності. На відміну від мов вищого рівня, які покладаються на збір сміття або інтерпретацію під час виконання, мова С створює скомпільований машинний код, який виконується безпосередньо на процесорі без проміжних шарів, що можуть спричинити непередбачувані затримки. Ця характеристика є важливою для дотримання вимог до затримок менше 100 мілісекунд, визначених для системи оцифрування відео [13, 14].

Можливість реалізувати точне керування синхронізацією за допомогою програмування мовою С дозволяє синхронізувати операції захоплення, обробки та передачі відео з точністю, необхідною для професійних відеододатків. Точна синхронізація та синхронізація з точністю до кадру може бути досягнута завдяки ретельному програмуванню обробників переривань, функцій таймерів та процедур апаратного інтерфейсу, які було б складно або неможливо надійно реалізувати на мовах вищого рівня.

Можливості керування пам'яттю в мові С дозволяють оптимізувати розподіл і керування відеобуферами, що є критично важливим для обробки великих відеокадрів і декількох паралельних потоків. Статичний розподіл пам'яті усуває непередбачувані затримки, пов'язані з динамічним розподілом пам'яті в інших мовах, а арифметика з вказівниками дозволяє ефективно маніпулювати структурами відеоданих. Ці можливості необхідні для підтримки постійної частоти кадрів і запобігання пропуску кадрів під час тривалих періодів роботи.

Широкі вимоги до апаратного інтерфейсу системи оцифрування відео потребують мови програмування, яка забезпечує прямий доступ до апаратних регістрів, обробників переривань та низькорівневих системних функцій. Мова

					КС КРБ 123.211.00.00 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

C є найкращою в цій галузі завдяки своїй здатності безпосередньо взаємодіяти з апаратним забезпеченням через ввід/вивід з відображенням пам'яті, вбудований асемблерний код та структуровані механізми обробки переривань.

Апаратні інтерфейси відеозахоплення вимагають точного керування синхронізацією та низькорівневих маніпуляцій з регістрами для конфігурування АЦП, керування маршрутизацією сигналів та керування операціями передачі даних. Мова C надає необхідні інструменти для реалізації драйверів пристроїв та рівнів абстракції апаратного забезпечення, які ефективно взаємодіють з користувацьким обладнанням, зберігаючи при цьому портативність та ремонтпридатність. Можливість реалізації операцій на бітовому рівні та накладання структур дозволяє ефективно пакувати та розпаковувати формати відеоданих з мінімальними накладними витратами на обробку.

Керування інтерфейсом GPIO для інтеграції в систему безпілотної авіації вимагає реакції на зовнішні сигнали в реальному часі та точного контролю синхронізації протоколів зв'язку. Програмування мовою C дозволяє реалізувати підпрограми вводу/виводу, керовані перериваннями, які можуть реагувати на зовнішні події за мікросекунди, зберігаючи при цьому стабільність і продуктивність системи. Ці можливості необхідні для інтеграції з системами керування польотом дронів та телеметричними інтерфейсами.

Програмування мережевого інтерфейсу для потокового відео вимагає низькорівневого програмування сокетів і можливостей реалізації протоколів, які найефективніше реалізуються мовою C. Можливість конструювати власні мережеві пакети, реалізовувати стеки протоколів і оптимізувати управління мережевими буферами забезпечує гнучкість, необхідну для реалізації адаптивних алгоритмів потокового мовлення і можливостей мовлення на багато пунктів призначення.

Вимоги до енергоспоживання та обчислювальної ефективності додатків для безпілотної авіації роблять оптимізацію продуктивності критично важливим

					КС КРБ 123.211.00.00 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

фактором при проектуванні. Програмування мовою C дозволяє реалізовувати високооптимізовані алгоритми, які мінімізують завантаження процесора і вимоги до пропускну здатності пам'яті, що безпосередньо призводить до зниження енергоспоживання і збільшення часу роботи від батареї.

Алгоритми обробки відео, реалізовані на мові C, можуть використовувати переваги специфічних для процесора оптимізацій, включаючи інструкції SIMD (Single Instruction, Multiple Data), оптимізацію кеш-пам'яті та конвеєрну оптимізацію. Процесор ARM Cortex-A72 в CM4 включає розширення NEON SIMD, які можуть значно прискорити операції обробки відео при правильному використанні за допомогою програмування на C та оптимізації вбудованої збірки [15].

Схеми доступу до пам'яті в додатках для обробки відео мають значний вплив на продуктивність кеш-пам'яті та загальну ефективність системи. Програмування на мові C забезпечує контроль, необхідний для оптимізації структур даних і реалізації алгоритмів для максимального використання кешу і мінімальних вимог до пропускну здатності пам'яті. Ці оптимізації можуть призвести до значного підвищення продуктивності та зниження енергоспоживання порівняно з реалізаціями на мовах вищого рівня.

Можливості оптимізації компілятора для C-коду вдосконалювалися протягом десятиліть розвитку, забезпечуючи зрілі та складні методи оптимізації, які можуть значно покращити продуктивність коду. Сучасні компілятори GCC та Clang включають вдосконалені методи оптимізації, спеціально розроблені для вбудованих систем та додатків реального часу, що дозволяє автоматично оптимізувати код мови C для цільових апаратних платформ.

Модульна архітектура системи оцифрування відео вимагає чітких інтерфейсів між програмними компонентами та апаратними рівнями абстракції, які можуть бути пристосовані до майбутніх апаратних змін або модернізації. Програмування мовою C полегшує це завдяки чітко визначеним

					КС КРБ 123.211.00.00 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

функціональним інтерфейсам, структурованим методам програмування та реалізації рівнів апаратної абстракції, які ізолюють специфічний для апаратного забезпечення код від логіки програми.

Вимоги до інтеграції операційної системи в систему на базі Linux найефективніше задовольняються за допомогою програмування мовою C, яка надає власні інтерфейси до функцій ядра, драйверів пристроїв та системних служб. Розгалужена екосистема Linux для розробки вбудованих систем включає розвинені фреймворки на C для обробки відео, мережових комунікацій та управління системою, які можуть значно скоротити час розробки та зменшити ризики.

Модульна архітектура системи оцифрування відео вимагає чітких інтерфейсів між програмними компонентами та апаратними рівнями абстракції, які можуть бути пристосовані до майбутніх апаратних змін або модернізації. Програмування мовою C полегшує це завдяки чітко визначеним функціональним інтерфейсам, структурованим методам програмування та реалізації рівнів апаратної абстракції, які ізолюють специфічний для апаратного забезпечення код від логіки програми.

Вимоги до інтеграції операційної системи в систему на базі Linux найефективніше задовольняються за допомогою програмування мовою C, яка надає власні інтерфейси до функцій ядра, драйверів пристроїв та системних служб. Розгалужена екосистема Linux для розробки вбудованих систем включає розвинені фреймворки на мові C для обробки відео, мережових комунікацій та управління системою, які можуть значно скоротити час розробки та зменшити ризики.

2.3.2 Програмна частина

Міркування крос-платформної переносимості є важливими для потенційної майбутньої адаптації системи до різних апаратних платформ або операційних систем. Код на мові C може бути розроблений з відповідними

					КС КРБ 123.211.00.00 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

рівнями абстракції, щоб забезпечити перенесення на різні архітектури процесорів або операційні системи з мінімальними змінами. Така переносимість забезпечує довгострокову гнучкість і знижує ризик застарівання технології [16].

Доступність бібліотечної екосистеми для програмування на мові C включає зрілі та добре протестовані бібліотеки для обробки відео, мережових комунікацій та розробки вбудованих систем. Такі бібліотеки, як FFmpeg для обробки відео, libav для реалізації кодеків та різноманітні мережеві бібліотеки надають надійні базові компоненти, які можуть прискорити розробку, забезпечуючи при цьому надійність та продуктивність [17].

Довготривала експлуатаційна придатність та підтримка системи оцифрування відео вимагає такої мови програмування та підходу до розробки, які можуть пристосуватися до вимог, що змінюються, та тривалого життєвого циклу продукту. Програмування мовою C має низку переваг у цій сфері, зокрема широку доступність для розробників, велику кількість документації та навчальних ресурсів, а також розвинену екосистему інструментів для розробки.

Читабельність коду та зручність супроводу у мові C можна досягти завдяки дисциплінованим практикам програмування, включаючи структуровані методи програмування, вичерпне коментування та модульний підхід до проектування. Хоча мова C вимагає більш явного програмування порівняно з мовами вищого рівня, ця явність насправді може покращити довгострокову супроводжуваність, роблячи поведінку системи зрозумілою та передбачуваною.

Довготривала експлуатаційна придатність та підтримка системи оцифрування відео вимагає такої мови програмування та підходу до розробки, які можуть пристосуватися до вимог, що змінюються, та тривалого життєвого циклу продукту. Програмування мовою C має низку переваг у цій сфері, зокрема широку доступність для розробників, велику кількість документації

					КС КРБ 123.211.00.00 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

та навчальних ресурсів, а також розвинену екосистему інструментів для розробки.

Реалізація системи оцифрування відео з дронів відбувається за допомогою модульного підходу до розробки, який використовує архітектурне розділення між обчислювальною платформою CM4 та аналоговим інтерфейсом друкованої плати на замовлення. Ця методологія дозволяє паралельно розробляти апаратні та програмні компоненти, мінімізуючи ризики інтеграції та залежність від графіка розробки.

Розробка апаратного забезпечення починається з детального електричного проектування друкованої плати, включаючи захоплення схеми, вибір компонентів та оптимізацію компонування друкованої плати. Процес проектування включає в себе інструменти моделювання та аналізу для перевірки цілісності сигналу, енергоспоживання та електромагнітної сумісності, перш ніж переходити до створення фізичних прототипів. Створення прототипів на ранніх стадіях дає змогу перевірити критичні аналогові характеристики та виявити будь-які конструктивні недоліки, що можуть вплинути на продуктивність системи.

Реалізація системи оцифрування відео з дронів відбувається за допомогою модульного підходу до розробки, який використовує архітектурне розділення між обчислювальною платформою CM4 та аналоговим інтерфейсом друкованої плати на замовлення. Ця методологія дозволяє паралельно розробляти апаратні та програмні компоненти, мінімізуючи ризики інтеграції та залежність від графіка розробки.

2.4 Проектування комп'ютерного пристрою

На основі вище обґрунтованих рішень та побудованої структурної схеми можливо розробити схеми електричні принципів для друкованих плат.

Процес розробки схеми починається з всебічного аналізу

					КС КРБ 123.211.00.00 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

функціональних вимог, що впливають зі специфікацій системи. Основною вимогою є захоплення аналогових композитних відеосигналів і перетворення їх у цифровий формат, придатний для обробки Raspberry Pi CM4. Це вимагає реалізації аналогових інтерфейсних схем з відповідним узгодженням ліній передачі аналогових сигналів, формуванням сигналу та можливостями аналого-цифрового перетворення.

Вимоги до системи живлення вимагають ефективного регулювання напруги від акумуляторних систем дронів, які зазвичай працюють при напрузі від 12 до 28 В постійного струму. Конструкція повинна забезпечувати кілька регульованих рівнів напруги, включаючи 5 В для аналогових схем, 3,3 В для цифрової логіки і потенційно 1,8 В для спеціалізованих інтерфейсних схем. Енергоефективність має вирішальне значення для мінімізації впливу на час польоту безпілотної системи, що вимагає використання імпульсних регуляторів напруги, що забезпечує ефективне перетворення напруги в широкому діапазоні.

Вимоги до інтерфейсів включають підключення до Raspberry Pi CM4 за допомогою високошвидкісних цифрових інтерфейсів, аналогових відеовходів через роз'єми, а також різні інтерфейси управління та стану для інтеграції з системами дронів. Конструкція також повинна включати відповідні схеми захисту для забезпечення надійної роботи в умовах електромагнітних завад, характерних для платформ безпілотної системи.

Секція керування живленням є основою всієї системи і вимагає ретельного проектування для забезпечення стабільної напруги для живлення пристрою. На схемі (див. рис. 2.2) показана реалізація топології імпульсного регулятора з використанням понижувального перетворювача TPS54620, який забезпечує первинну шину напруги 5В від високовольтної акумуляторної збірки системи дрона.

Конфігурація TPS54620 включає необхідні компоненти для стабільної роботи блоку живлення: котушка індуктивності (L1) визначає пульсації

					КС КРБ 123.211.00.00 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

струму і перехідні характеристики, а вихідні конденсатори (C10, C11, C12) забезпечують вихідну фільтрацію і накопичення енергії для перехідних процесів навантаження. Мережа зворотнього зв'язку (R6, R7) встановлює вихідну напругу 5 В. Для ізоляції лінії напруги 5 В використано гальванічну розв'язку живлення (ROE0505S).

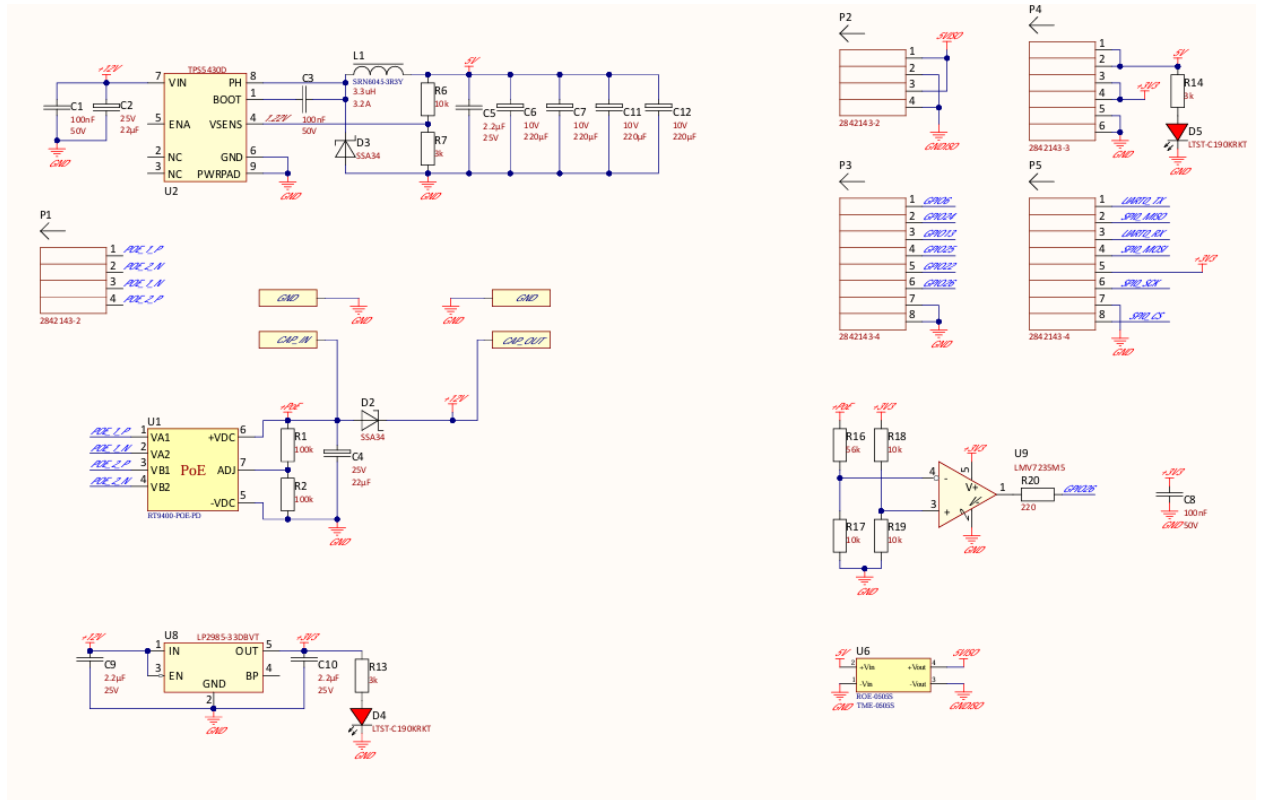


Рисунок 2.2 – Схема електрична принципова блоку живлення

Схеми вхідного захисту і фільтрації запобігають пошкодженню від перехідних процесів напруги і електромагнітних перешкод, які можуть бути присутніми в системі живлення дрона. Мережа вхідних конденсаторів (C1, C2) забезпечує локальне накопичення енергії та високочастотну фільтрацію, в той час як схема ввімкнення дозволяє здійснювати контрольований запуск і вимкнення системи живлення. Сигнал про наявність живлення забезпечує індикацію стану для системи керування, що дозволяє правильно визначити послідовність ініціалізації системи.

У лінії напруги 3,3 В використовується підхід лінійного регулятора для

застосувань, що вимагають низького рівня шуму. Лінійний стабілізатор AMS1117-3.3 забезпечує чисте живлення 3,3 В, придатне для чутливих аналогових і цифрових схем. Хоча лінійний підхід менш ефективний, ніж імпульсні регулятори, він усуває комутаційні шуми, які можуть впливати на аналогові схеми обробки відеосигналу.

Реалізація регулятора включає відповідні вхідні та вихідні ємності для стабільності та перехідної характеристики. Вхідний конденсатор (C13) забезпечує зменшення впливу комутаційного шуму від попередніх ланцюгів живлення. Вихідний конденсатор забезпечує стабільну вихідну напругу та усуває шуми в діапазоні резонансної частоти конденсатора.

Індикація стану за допомогою світлодіодних ланцюгів забезпечує візуальний зворотний зв'язок про роботу системи живлення. Світлодіоди індикатора живлення (D4, D5) використовують струмообмежувальні резистори для забезпечення відповідної яскравості при мінімальному енергоспоживанні. Ці індикатори полегшують налагодження системи та надають інформацію про робочий стан під час інтеграції та тестування системи.

Секція вхідного аналогового відеосигналу є одним з найбільш важливих аспектів схеми, що безпосередньо впливає на якість оцифрованого відеосигналу. Вхідні ланцюги повинні забезпечувати належне узгодження ліній, щоб запобігти віддзеркаленню сигналу, яке погіршує якість відео. Це досягається завдяки ретельному підбору вхідних кінцевих компонентів і міркувань щодо конструкційних рішень лінії передачі.

Схеми фільтрації відокремлюють відеосигнал від будь-яких постійних перешкод, присутніх у джерелі, зберігаючи при цьому повну смугу пропускання відеосигналу. Вибір конденсатора зв'язку вимагає балансування низькочастотної характеристики з урахуванням фізичних обмежень за розміром. Мережа зв'язку також включає схеми відновлення постійного струму, які встановлюють належний рівень постійного струму для наступних

					КС КРБ 123.211.00.00 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

етапів обробки.

Схеми автоматичного регулювання підсилення (АРП) компенсують коливання амплітуди вхідного сигналу, які можуть виникати через різне обладнання джерела або втрати в кабелі. Реалізація АРП відстежує амплітуду відеосигналу і регулює коефіцієнт підсилення програмованих підсилювачів, щоб підтримувати постійний рівень сигналу для оптимальної продуктивності аналого-цифрового перетворення [18].

Схеми фільтрів згладжування перешкоджають високочастотному шуму та завадам, які спотворюють оцифрований відеосигнал через ефекти згладжування. Конструкція фільтра повинна забезпечувати правильне ослаблення частот вище частоти Найквіста, зберігаючи при цьому пласку АЧХ в межах смуги пропускання відеосигналу. Схема реалізує багатоступеневий підхід з використанням як пасивних, так і активних компонентів фільтра.

Вибір топології фільтра враховує компроміс між продуктивністю фільтра, енергоспоживанням і складністю схеми. Активні фільтри з використанням операційних підсилювачів забезпечують вищу продуктивність, але споживають більше енергії, тоді як пасивні фільтри мають нижче енергоспоживання, але можуть вимагати більшої кількості компонентів. Остаточна конструкція являє собою оптимізований компроміс для вимог застосування безпілота.

Схеми буферних підсилювачів ізолюють схеми фільтрів від впливу навантажень наступних етапів обробки. Ці буфери підтримують цілісність сигналу, забезпечуючи при цьому необхідну здатність керування інтерфейсами аналого-цифрового перетворювача. Конструкція буфера передбачає відповідні характеристики смуги пропускання та частоти дискретизації для збереження цілісності відеосигналу [19].

За таким самим принципом створюємо схеми електричні принципи для блоку розширення (див. дод. Б) та блоку з STM32 (див. рис. 2.3).

					КС КРБ 123.211.00.00 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

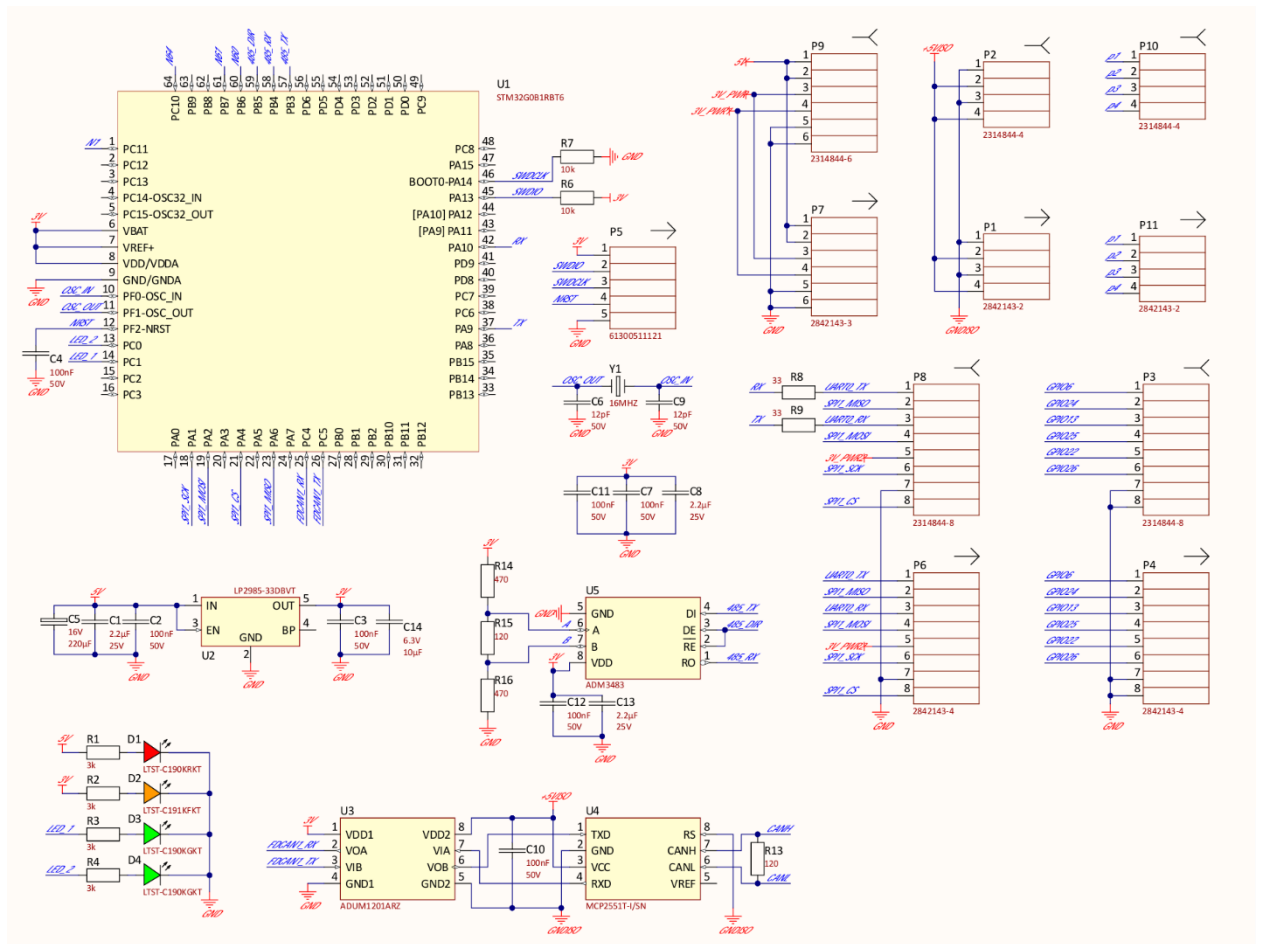


Рисунок 2.3 – Схема електрична принципова блоку з STM32

Також було розроблено схему електричну принципову для блоку розширення (див. рис. 2.4).

Таким чином було розроблено базові схеми для створення комп'ютерної системи для оцифрування потокового відеосигналу для безпілотників.

Перед друкуванням блоків потрібно також розробити схему з'єднань, а саме прокласти з'єднання між ключовими блоками, для того, щоб надалі було зрозуміло, як саме елементи поєднуються між собою (див. рис. 2.5).

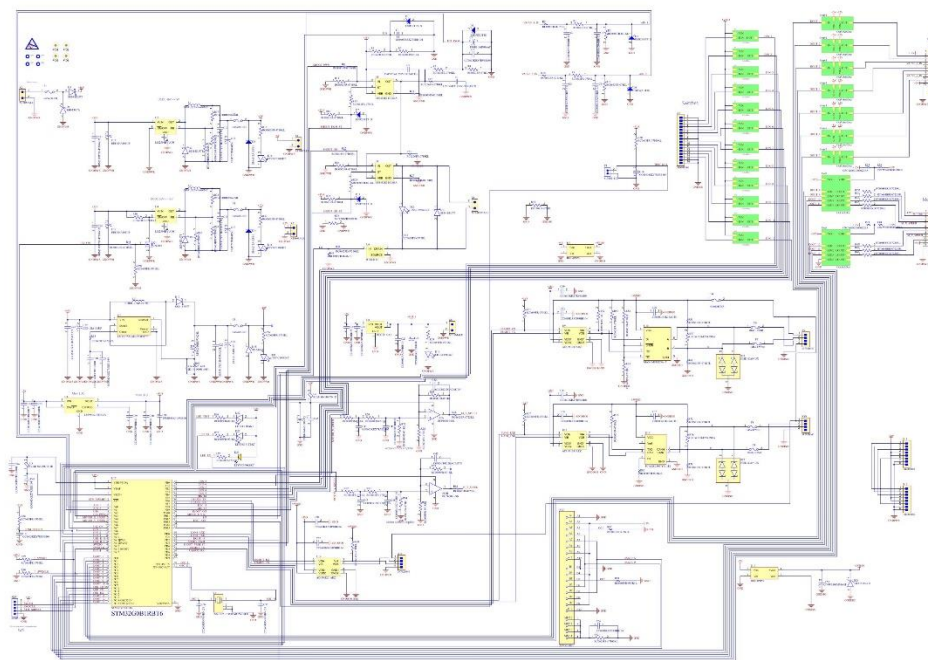


Рисунок 2.4 – Схема електрична принципова блоку розширення

Далі потрібно експортувати файл у форматі Gerber та надіслати на друк, після чого плату можливо запаяти.

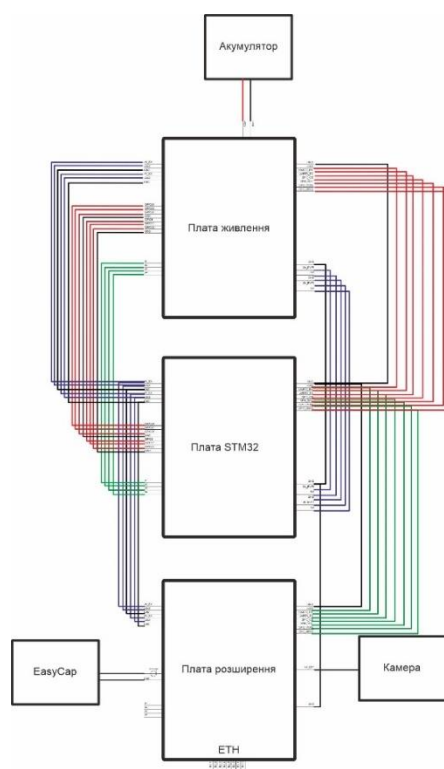


Рисунок 2.5 – Схема з'єднань комп'ютерної системи

					КС КРБ 123.211.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

Після цього, потрібно розробити блок-схему програмного коду, який буде прошиватися в STM32. Розглянути розроблену блок-схему можливо переглянути у додатку на рисунку 2.6.

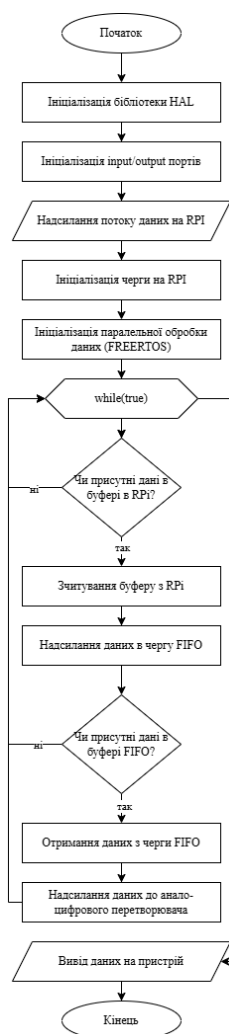


Рисунок 2.6 – Блок-схема програмного коду

На початку виконання програми відбувається ініціалізація бібліотеки HAL, що слугує абстрактним шаром між апаратним та програмним забезпеченням, та input/output портів друкованої плати. Після цього з підключеної камери починається надсилання потоку даних в буфер RPi. Паралельно на RPi ініціалізується черга для зберігання даних, що приходять з камери та ініціалізація паралельної обробки даних (FREERTOS).

Після здійснених ініціалізацій починається цикл пересилання даних:

якщо в буфері на RPi є дані (кадри з камери), вони зчитуються та надсилаються в чергу FIFO, що на блоку STM32. Якщо у буфері на RPi даних немає, цикл починається наново та очікує даних у буфері. Після успішного надсилання даних у чергу йде перевірка чи вони присутні у буфері FIFO: якщо так, то дані надсилаються до аналогово-цифрового перетворювача. Там аналоговий потік даних перетворюється на цифровий та виводиться на підключений пристрій. Якщо у черзі даних немає, цикл повторюється з початку [20, 21, 22].

					КС КРБ 123.211.00.00 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 ПРОГРАМУВАННЯ ТА ТЕСТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ОЦИФРУВАННЯ ПОТОКОВОГО ВІДЕОСИГНАЛУ

3.1 Прошивання Compute Module Raspberry Pi 4

Для того, щоб зашити систему, потрібно для початку взяти адаптер для зчитування карти пам'яті eMMC (див. рис. 3.1). Дану карту пам'яті можливо прошити окремо від всієї системи, забезпечуючи зручне використання.



Рисунок 3.1 – Адаптер для eMMC

Підключивши даний адаптер з картою пам'яті до персонального комп'ютера, можливо прошити eMMC образом для Raspberry Pi CM4.

Після завантаження образу, першим відкривається вікно, на якому потрібно обрати систему, яка буде зашиватися на модуль. В даному випадку була обрана система BullsEye ARM64.

Далі, натиснувши кнопку «Next», відкривається вікно загальних налаштувань (див. рис. 3.3).

					КС КРБ 123.211.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Коношенко Я.А.			Програмування та тестування комп'ютерної системи оцифрування потокового відеосигналу			Літ.	Арк.	Аркушів	
Перевір.		Лецишин Ю.З.								44	
Реценз.		Цуприк Г.Б						ТНТУ, каф. КС, гр. CI-41			
Н. Контр.		Луцик Н.С.									
Затверд.		Осухівська Г.М.									



Рисунок 3.2 – Вибір системи для прошивки Raspberry Pi CM4

В ньому потрібно зазначити ім'я користувача («Username») та пароль («Password»). Дана інформація потім знадобиться при підключенні до пристрою.

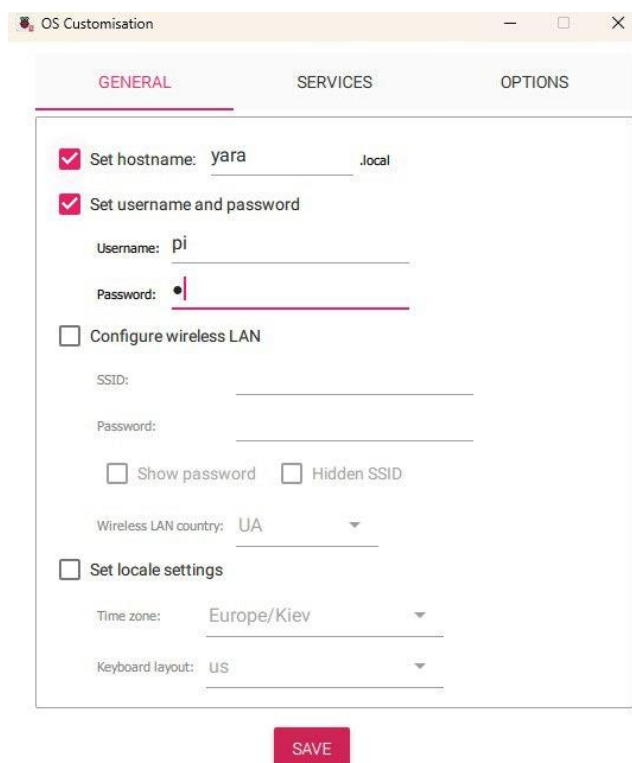


Рисунок 3.3 – Загальні налаштування прошивки

Після цього потрібно перейти на вкладку «Services» та необхідно

					КС КРБ 123.211.00.00 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

включити SSH («Enable SSH») (див. рис. 3.4). Також, для більшої безпеки, треба включити аутентифікацію по паролю («Use password authentication»).

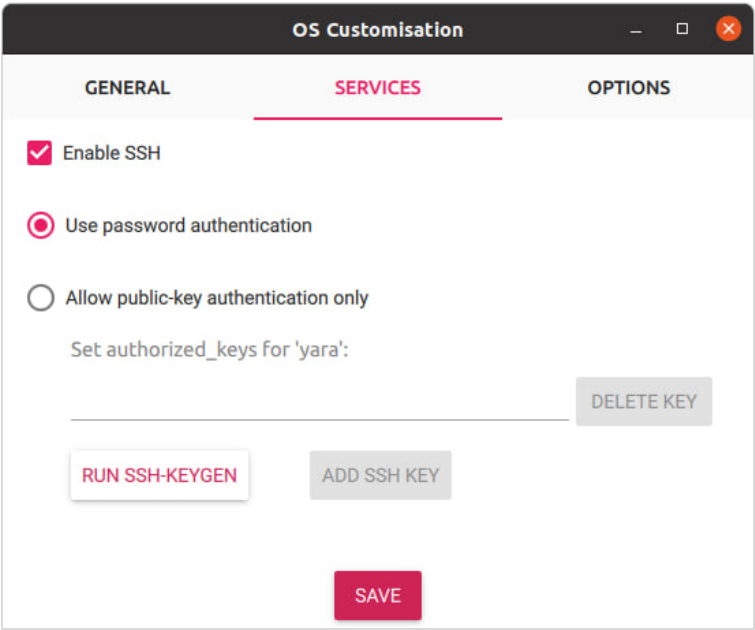


Рисунок 3.4 – Підключення сервісів системи

Після натискання кнопки «Save» потрібно підтвердити всі зміни та натиснути кнопку «Yes» (див. рис. 3.5). Якщо потрібно внести ще якісь зміни, або ж скинути їх до стандартних налаштувань, це вікно пропонує дані опції.



Рисунок 3.5 – Підтвердження змін в операційній системі Raspberry Pi

Далі відкривається вікно підтвердження встановлення операційної системи на карту пам'яті eMMC (див. рис. 3.6). Після цього можливо витягнути карту з адаптера та вставити на блок розширення.

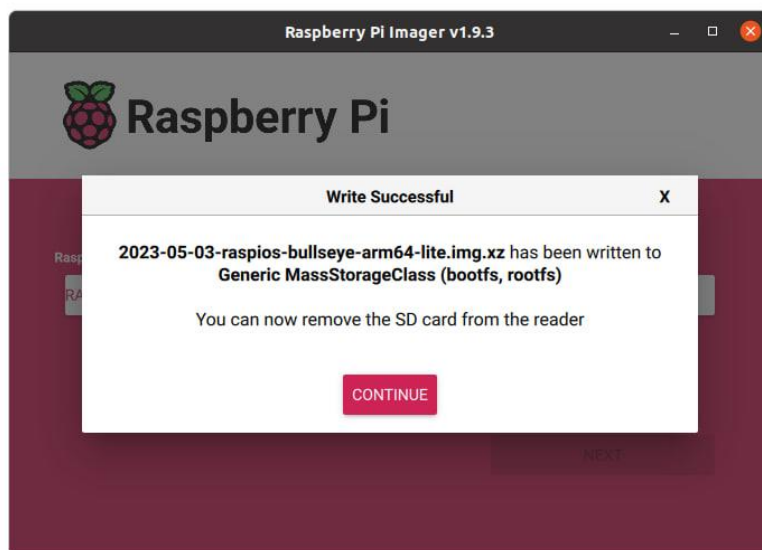


Рисунок 3.6 – Підтвердження встановлення системи

Таким чином подалі до Raspberry Pi CM4 можливо буде під'єднатися по SSH по зазначеним логіну та пароллю.

3.2 Розробка програмного коду для STM32

На основі обґрунтування, прописаного в розділі 2.4, можна зробити висновок, що написання програмного коду для оцифрування потокового відеосигналу краще за все зробити мовою C.

Програмний код складається з двох частин – бібліотека та основний код. У бібліотеці прописуються основні функції. На рисунку 3.7 можливо побачити структуру буфера FIFO [23].

```

typedef struct FIFO_ { //Структура
    S32 in; // Наступний In Index
    S32 out; // Наступний Out Index
    U8 *buf; // Буфер
    S32 buf_size; // Розмі
    INT item_size; // Size in bytes for each element in the
buffer.
} FIFO;

```

Рисунок 3.7 – Лістинг структури буфера FIFO

Наступна функція, що оголошується вже в основному коді, є функція `fifo_add` (див. рис. 3.8), що додає кадри, отримані з камери, в буфер FIFO до черги.

```

BOOL _fifo_add( FIFO *fifo, const void *item )
{
    U32 offset;
    test_param( NULL == fifo );
    test_param( NULL == item );
    if( _fifo_free( fifo ) >= fifo->item_size )
    {
        offset = fifo->in & ( fifo->buf_size - 1 );
        fifo->in += fifo->item_size;
        _memcpy( fifo->buf + offset,
                item,
                fifo->item_size );
        return TRUE;
    }
    return FALSE;
}

```

Рисунок 3.8 – Лістинг додавання кадрів в чергу буфера FIFO

Для того, щоб буфер отримав дані з черги, була написана функція `fifo_get`, що ініціалізується одразу після `fifo_add` (див. рис. 3.9). Функція отримує кадри з черги, до якої їх додала функція `fifo_add` та передає на опрацювання далі.

```

BOOL _fifo_get( FIFO *fifo, void *item )
{
    U32 offset;

    test_param( NULL == fifo );
    test_param( NULL == item );

    if( _fifo_available( fifo ) >= fifo->item_size )
    {
        offset = fifo->out & ( fifo->buf_size - 1 );
        fifo->out += fifo->item_size;
        _memcpy( item, fifo->buf + offset, fifo->item_size );
        return TRUE;
    }
    return FALSE;
}

```

Рисунок 3.9 – Лістинг отримання кадрів з черги буфера FIFO

Для перевірки наявності кадрів для відправки на опрацювання, використовується функція `fifo_available` (див. рис. 3.10). Дана функція перевіряє кількість кадрів на вхід та кількість кадрів на вихід, дозволяючи зрозуміти, чи є кадри в черзі на опрацювання.

```

S32 _fifo_available( FIFO *fifo )
{
    test_param( NULL == fifo );
    S32 i = ( fifo->in - fifo->out );
    if( i < 0 )
    {
        i = -i;
    }
    return i;
}

```

Рисунок 3.10 – Лістинг функції перевірки наявності кадрів

Далі, функція `memcpy` копіює дані з буфера, тобто дістає кадри з обчислювального модуля (див. рис. 3.11).

					КС КРБ 123.211.00.00 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

```

void *_memcpy( void *dst, const void *src, INT n )
{
    test_param( NULL == dst );
    test_param( NULL == src );
    test_param( n < 1 );

    register U8 *pdst = (U8 *)dst;
    register U8 *psrc = (U8 *)src;

    for( ; 0 < n; n-- )
    {
        *(U8 *)pdst++ = *(U8 *)psrc++;
    }
    return pdst;
}

```

Рисунок 3.11 – Лістинг функції копіювання даних з буферу

Останнє, на рисунку 3.12 описана функція, що створює саму систему FIFO, а саме прописує її параметри та ініціалізує.

```

BOOL __fifo_init_static( FIFO *fifo, void *pbuf, U32
fifo_size, U32 item_size )
{
    //Sanity check
    test_param( NULL == fifo );
    test_param( NULL == pbuf );
    test_param(fifo_size < 2 ); // Size is too small. It must
be larger than 2
    #if FIFO_2X
        test_param((fifo_size & (fifo_size - 1)) != 0 ); // Must be
a power of 2
    #endif
    test_param( item_size > fifo_size );
    {
        //return FALSE;
    }
    fifo->buf = (U8 *)pbuf;
    fifo->buf_size = fifo_size;
    fifo->item_size = item_size;
    __fifo_flush( fifo );
    return TRUE;
}

```

Рисунок 3.12 – Лістинг ініціалізації системи FIFO

					КС КРБ 123.211.00.00 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Також, важливо прописати програмний код для коректної роботи блоку з STM32, що не дає спрацювати апаратному перезавантаженню системи (див. рис. 3.13).

```
import RPi.GPIO as GPIO
from time import sleep

WD_PIN = 27 #6
SLEEP = 0.5
UP = 1
DOWN = 0

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False) # Turn off warnings output
GPIO.setup(WD_PIN, GPIO.OUT) # Set GPIO to output

print(GPIO.input(WD_PIN))

try:
    while True:
        GPIO.output(WD_PIN, UP) # set GPIO to
1/GPIO.HIGH/True
        print(GPIO.input(WD_PIN))
        sleep(SLEEP) # wait half a second
        GPIO.output(WD_PIN, DOWN) # set GPIO to
0/GPIO.LOW/False
        print(GPIO.input(WD_PIN))
        sleep(SLEEP) # wait half a second

    except KeyboardInterrupt: # trap a CTRL+C keyboard
interrupt
        GPIO.cleanup()
```

Рисунок 3.13 – Лістинг призупинення автоматичного перезавантаження

Для коректного завершення роботи використовується лістинг, наведений на рисунку 3.14.

За допомогою наведених скриптів, написаних на мові програмування Python, система буде працювати коректно та без перебоїв.

					КС КРБ 123.211.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

```

import RPi.GPIO as GPIO
from time import sleep
import os

POFF_PIN = 22
DOWN = 0
UP = 1

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False) # Turn off warnings output
GPIO.setup(POFF_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP) #
Set GPIO to input

sleep(2)
print(GPIO.input(POFF_PIN))

def shutdown(channel):
    sleep(0.5)
    if GPIO.input(POFF_PIN) == UP:
        print("Poweroff ...")
        os.system("sudo poweroff")

GPIO.add_event_detect(POFF_PIN, GPIO.RISING,
callback=shutdown, bouncetime=2000)
while True:
    sleep(1)

```

Рисунок 3.14 – Лістинг коректного завершення роботи апаратної частини

Скрипти працюють у фоновому режимі за допомогою сервісів, використовуючи systemd [24].

					КС КРБ 123.211.00.00 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

3.3 Тестування

Для того, щоб розпочати тестування, потрібно зібрати всі блоки та підключити до персонального комп'ютера. За структурною схемою, що була описана в розділі 2.1, підключаємо блок розширення до Raspberry Pi CM4. Далі, потрібно підключити блок живлення та блок з STM32.

Підключивши акумулятор до блоку живлення, потрібно під'єднати Ethernet для стабільного підключення до мережі та вдалого підключення по SSH.

Наступним кроком є підключення системи до персонального комп'ютера через перетворювач EasyCAP. На рисунку 3.15 можливо побачити повністю зібрану систему разом з камерою та об'єктивом.



Рисунок 3.15 – Зібране та підключене апаратне забезпечення

Після підключення до персонального комп'ютера, потрібно підключитися до системи по SSH за допомогою програмного забезпечення MobaXterm по налаштованих параметрах, описаних в розділі 3.1.1. Підключення вимагає вводу імені користувача та пароля. Результат успішного підключення

до системи можливо побачити на рисунку 3.16.

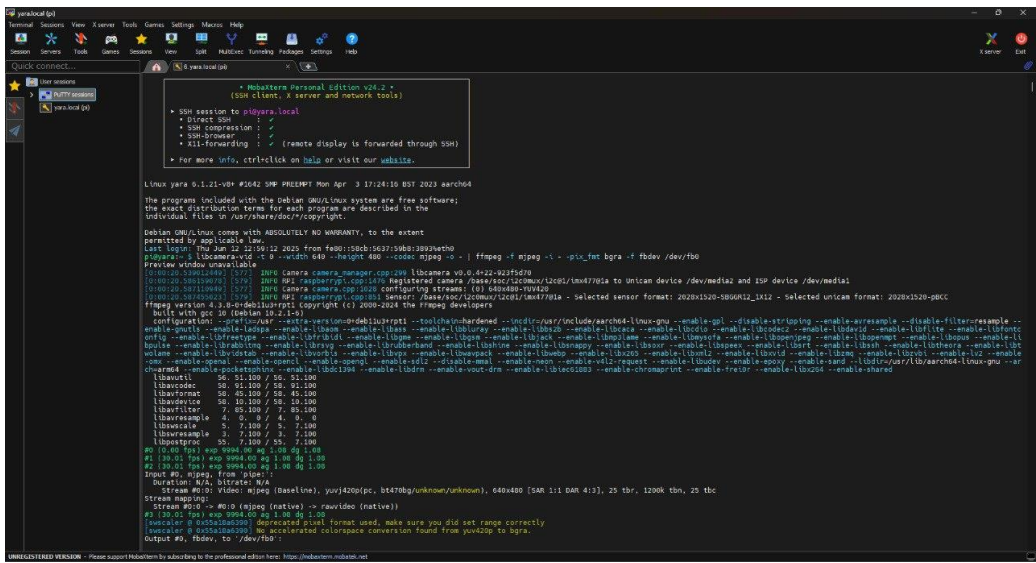


Рисунок 3.16 – Результат успішного підключення до системи

Для того, щоб камера зчитувалася, потрібно налаштувати її config файл. Це необхідно для того, щоб Raspberry Pi могла її зчитати як пристрій, який надсилає дані. Відкрити редагування даного файлу можливо за допомогою наступної команди.

```
sudo nano /boot/config.txt
```

Для правильного налаштування потрібно налаштувати параметри, описані на рисунку 3.17.

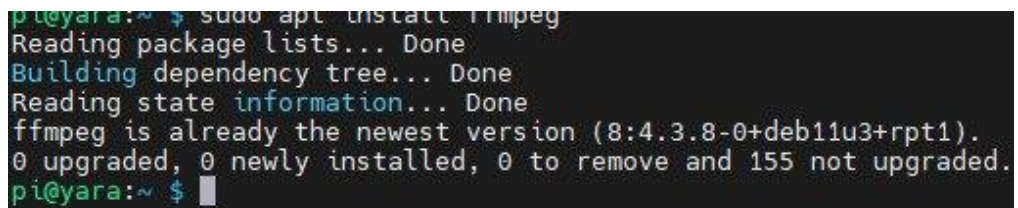
```
hdmi_force_hotplug=1
framebuffer_width=640
framebuffer_height=480
sdtv_mode=1
sdtv_aspect=1
enable_tvout=1
#camera_auto_detect=1
dtoverlay=imx477,caml
dtparam=caml_reg
```

Рисунок 3.17 – Лістинг даних, що потрібно зазначити в config файлі

Оскільки кадровий буфер підтримує формат RGBA, а камера видає потік даних у SRGGB10, необхідно конертувати формат. Для цього потрібно встановити пакет libcamera-vid у зв'язці з ffmpeg. Дане можливо зробити за допомогою наступних команд.

```
sudo apt update  
sudo apt install ffmpeg
```

Процес виконання даних команд можливо побачити на рисунку 3.18.



```
pi@yara:~$ sudo apt install ffmpeg  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
ffmpeg is already the newest version (8:4.3.8-0+deb11u3+rpt1).  
0 upgraded, 0 newly installed, 0 to remove and 155 not upgraded.  
pi@yara:~$
```

Рисунок 3.18 – Завантаження необхідних пакетів

Зображення захоплюється за допомогою libcamera-vid та конвертує його у формат, який є сприйнятним для ffmpeg. Формати, які підтримуються даним пакетом є mjpeg та yuv420. Далі воно кодує дані у формат RGBA та передає в кадровий буфер.

Для цього необхідно ввести наступну команду:

```
libcamera-vid -t 0 --width 640 --height 480 --codec yuv420 -  
o - | ffmpeg -f rawvideo -pix_fmt yuv420p -s 640x480 -i - -pix_fmt  
bgra -f fbdev /dev/fb0
```

Перевірити, чи камера працює та зчитується можливо через термінал, де буде постійним потоком видавати потік кадрів (див. рис. 3.19). Щоб відкрити оцифроване відео, можливо використати програму VLC або GStreamer. Зробити це можливо за допомогою наступної команди.

```
gst-launch-1.0 v4l2src device=/dev/video0 ! videoconvert !  
autovideosink
```

В даному випадку використовувалася програма VLC. На рисунку 3.19

					КС КРБ 123.211.00.00 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

можливо побачити успішне виконання системи.

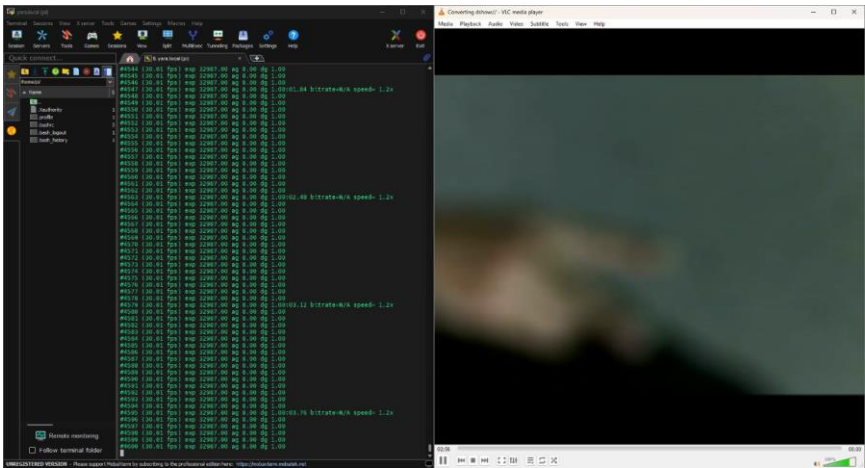


Рисунок 3.19 – Успішне виконання системи

Таким чином було протестовано розроблену комп’ютерну систему для оцифрування потокового відеосигналу для безпілотників.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська допомога при ураженні електричним струмом

Використання електричної енергії є невід'ємною складовою наукових досліджень, зокрема при роботі з комп'ютерною технікою. Незважаючи на відносно низькі рівні напруги в побутових ПК (220 В), неправильна експлуатація або технічні несправності можуть призвести до ураження електричним струмом. Дія електричного струму на організм людини залежить від типу струму, напруги, тривалості його проходження, шляху проходження, індивідуальних особливостей і оточуючого середовища [25].

Електричний струм, проходячи через тіло людини, викликає низку фізіологічних ефектів: судоми, зупинку серця, термічні опіки, електроліз тканин. У контексті експлуатації ПК виникають наступні ситуації:

- пошкодження ізоляції проводів живлення або блоку живлення;
- дотик до неізольованих елементів при розібраному корпусі ПК;
- використання несправних подовжувачів або розеток;
- висока вологість у приміщенні або відсутність заземлення.

Особлива увага приділяється дотриманню наступних правил при користуванні ПК чи ПЕОМ, відповідно до яких необхідно:

- використовувати справні мережеві фільтри з функцією захисту від перенапруги;
- уникати відкритого контакту з внутрішніми компонентами ПК, особливо при ввімкненому живленні;
- регулярно перевіряти заземлення комп'ютерної техніки;
- дотримуватись інструкцій виробника при використанні лабораторного устаткування;

					КС КРБ 123.211.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	Безпека життєдіяльності, основи охорони праці		
Розроб.		Коношенко Я.А.					
Перевір.		Лецишин Ю.З.					
Консульт.		Пилипець М.І.					
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркушів
						57	
					ТНТУ, каф. КС, гр. СІ-41		

- не працювати на вологих поверхнях або з мокрими руками.

Важливо пам'ятати, що навіть при низьких напругах можливе ураження струмом — особливо при підвищеній вологості або пошкодженій ізоляції.

Згідно з Порядком надання домедичної допомоги постраждалим при ураженні електричним струмом або блискавкою Наказу Міністерства охорони здоров'я України від 09 березня 2022 року № 441, першочергові кроки до та при наданні домедичної допомоги включають [26]:

1) перед наданням допомоги переконатися у відсутності небезпеки для себе, оточуючих, постраждалого та тільки за її відсутності перейти до наступного кроку;

2) якщо постраждалий у свідомості, заспокоїти та пояснити свої наступні дії;

3) здійснити виклик екстреної медичної допомоги та дотримуватись вказівок диспетчера прийому виклику;

4) при ураженні постраждалого електричним струмом:

- якщо постраждалий без свідомості, впевнитись, що дія електричного струму на постраждалого припинена;

- всі дії щодо припинення дії електричного струму слід здійснювати за умови проходження відповідного навчання або здійснити виклик за єдиним телефонним номером системи екстреної допомоги населенню 112;

- якщо дія електричного струму на постраждалого припинена, слід надати йому домедичну допомогу, відповідно до наявних пошкоджень;

5) забезпечити постійний нагляд за постраждалим до приїзду бригади екстреної (швидкої) медичної допомоги;

6) при погіршенні стану постраждалого до приїзду бригади екстреної (швидкої) медичної допомоги повторно здійснити виклик екстреної медичної допомоги;

7) за можливості зібрати у постраждалого чи оточуючих максимально можливу інформацію стосовно обставин отримання травми. Всю отриману

					КС КРБ 123.211.00.00 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

інформацію передати працівникам бригади екстреної (швидкої) медичної допомоги або диспетчеру служби екстреної медичної допомоги [26].

Знання та практичні навички надання долікарської допомоги при ураженні електричним струмом є необхідними для кожного, хто працює з електронним обладнанням, особливо в рамках розробки наукових програмних продуктів. Врахування електробезпеки під час роботи з ПК не лише зберігає здоров'я, а й запобігає втратам обладнання та зупинці процесу моніторингу навколишнього середовища за допомогою безпілота.

4.2 Естетичне оформлення та ергономічне дослідження робочого місця оператора

Питання естетичного та ергономічного оформлення робочого місця має прямий вплив як на здоров'я і самопочуття операторів, так і на ефективність виконаної роботи. Особливо це важливо для операторів, чия діяльність пов'язана з комп'ютерними технологіями, оскільки тривала робота за комп'ютером у некомфортних умовах може призвести до фізичних і психологічних проблем. Оператор комп'ютера, який не відчувається комфортно, не може виконувати завдання з належною якістю, що негативно позначається на результатах роботи.

Робоче місце — це зона, де оператор проводить значну частину свого робочого часу. Раціональна організація робочих місць допомагає зменшити фізичний дискомфорт, знизити рівень втоми та підвищити продуктивність праці. За результатами досліджень, правильно організоване робоче місце може підвищити продуктивність на 15-25% [27].

Правильна організація робочого місця сприяє зменшенню втоми, підвищенню працездатності та зниженню загального дискомфорту. Дослідження показують, що при ергономічно організованому робочому місці продуктивність праці може збільшитися на 15-25% [28]. Тому правильне

					КС КРБ 123.211.00.00 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

оформлення робочого місця є важливим аспектом у забезпеченні здоров'я працівників і підвищенні їхньої ефективності.

Організація робочого місця включає кілька ключових аспектів:

- правильне розміщення робочого місця у виробничому приміщенні з урахуванням природного освітлення. Робочі місця повинні бути орієнтовані таким чином, щоб світло з вікон падало збоку, переважно зліва, щоб уникнути відблисків на екрані і не спричиняти дискомфорту;
- вибір ергономічного робочого положення та меблів, що відповідають антропометричним характеристикам людини. Врахування таких факторів допомагає створити комфортні умови для роботи, зменшуючи навантаження на спину, шию та очі;
- розташування обладнання на робочих місцях має бути раціональним, щоб працівник мав легкий доступ до всіх необхідних інструментів, не перевантажуючи простір;
- урахування особливостей трудової діяльності. Для працівників, які працюють з відеотерміналами, вимоги до організації робочого місця строго регламентуються стандартами, такими як НПАОП 0.00-1.28-10, що визначають оптимальні відстані між екранами та меблями [29].

Згідно з вимогами, площа для одного робочого місця з персональним комп'ютером має становити не менше 6 м², а об'єм приміщення — не менше 20 м³. Важливо дотримуватись таких норм:

- відстань від робочого місця до стін зі світловими прорізами повинна бути не менше 1 метра;
- між бічними поверхнями відеотерміналів має бути відстань не менше 1,2 метра;
- між тильною поверхнею одного відеотермінала і екраном іншого повинна бути відстань не менше 2,5 метра;
- ширина проходів між рядами робочих місць повинна бути не менше 1 метра [30].

					КС КРБ 123.211.00.00 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Ці вимоги є частиною стандартів, які регулюють правильне компонування робочих місць для забезпечення ефективності праці, зниження рівня втоми та покращення здоров'я операторів.

Під час роботи оператора, його здоровий стан має позіціонуватися в першому порядку, оскільки від його швидкої реакції залежить точність моніторингу та сприянню дій у випадку помічення порушень.

Правильна ергономіка робочого місця оператора БПЛА має першорядне значення для збереження здоров'я та підтримання високої працездатності протягом тривалих польотних сесій. Оператори часто проводять за пультом управління по 8-12 годин підряд, що створює значне навантаження на опорно-руховий апарат, зір та нервову систему.

Неправильно організоване робоче місце може призвести до розвитку професійних захворювань, включаючи синдром зап'ястного каналу, остеохондроз шийного та поперекового відділів хребта, синдром комп'ютерного зору. Ці проблеми не лише негативно впливають на здоров'я оператора, але й значно знижують якість управління БПЛА, що може мати критичні наслідки для безпеки польотів [31].

					КС КРБ 123.211.00.00 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розроблено комп'ютерну систему, а саме:

1) Розроблено схеми електричні принципові та схема мотажна (з'єднань), що дозволяє надалі створити апаратне забезпечення для системи.

2) Розроблено апаратне забезпечення, а саме спеціальну друковану плату, що працює на основі обчислювального модуля Raspberry Pi 4. Дана система також містить в собі мікроком'ютер STM32, що відповідає за опрацювання кадрів. Система також містить в собі камеру.

3) Розроблено програмне забезпечення системи з урахуванням специфічних вимог до обробки відеопотоку в реальному часі. Основою архітектури програмного забезпечення стала реалізація ефективної системи черги FIFO (First In, First Out), яка забезпечує послідовну обробку кадрів без втрат та з мінімальними затримками.

4) Налаштовано з'єднання системи по мережі завдяки використанню SSH. Система виводить оцифроване відео з камери на персональний комп'ютер.

5) Проведено комплексне тестування системи, яке підтвердило коректну роботу всіх підсистем, мінімальні затримки передачі відеопотоку, швидкість передачі кадрів та стабільність роботи в тривалому режимі експлуатації.

6) Продемонстровано практичне призначення комп'ютерної системи оцифрування потокового відеосигналу для безпілотників, а саме можливість з'єднання по локальній мережі для отримання відео з систем спостереження.

					КС КРБ 123.211.00.00 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жаровський Р.О., Луцик Н.С., Осухівська Г.М., Паламар А.М., Тиш Є.В. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп’ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2024. 39 с.
2. Лецишин Ю., Герасименко А., Герасименко О. Комп’ютерна система моніторингу радіозв’язку від БПЛА // Матеріали 12-ї науково-технічної конференції „Інформаційні моделі, системи та технології “. Тернопіль, 2024. С.140.
3. Безрук В., Чернега В., Поморова О. Цифрова обробка відеосигналів: теорія та практичні застосування. Харків : Харків. нац. ун-т радіоелектроніки, 2019. С.89-112
4. Rodriguez A., Martinez C., Thompson D. Low-latency Video Streaming for UAV Applications: Architecture and Implementation. Journal of Real-Time Image Processing. 2020. Vol. 14, no. 4. P. 1–15.
5. Upton E., Halfacree G. Raspberry Pi User Guide. Indianapolis, IN : Wiley, 2016. P.78-99
6. Raspberry Pi Ltd. Raspberry Pi Compute Module 4 Datasheet. Cambridge, UK : Cambridge University Press, 2020. P.33
7. Паламар М.І., Стрембіцький М.О., Паламар А.М. Проектування комп’ютеризованих вимірювальних систем і комплексів. Навчальний посібник. Тернопіль: ТНТУ. 2019. 150 с.
8. Voloskyi V., Leshchyshyn Y., Romanyshyn N., Palamar A., Tarasenko L. Method and algorithm for efficient cell balancing in the lithium-ion battery control system. CEUR Workshop Proceedings, The 1st International Workshop on Bioinformatics and Applied Information Technologies (BAIT 2024), Zboriv, Ukraine, October 02-04, 2024. Vol. 3842. P. 258-267.

					КС КРБ 123.211.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

9. Tymkiv P., Leshchyshyn Y. Algorithm Reliability of Kalman Filter Coefficients Determination for Low-Intensity Electroretinosignal. IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM). 2019. P.1-5.

10. Ярмусь О., Лецишин Ю., Варавін А. Методи та засоби моделювання швидкості Wi-Fi мережі, Матеріали XI науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопіль: ТНТУ. 2023. С.11

11. Leschyshyn Y., Scherbak L., Nazarevych O., Gotovych V., Tymkiv P., Shymchuk G. Multicomponent Model of the Heart Rate Variability Change-point. IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH). 2019. P. 110–113.

12. Method of Gas Consumption Change-point Detection based on Seasonality Multicomponents Model / Leshchyshyn Yu., Nazarevych O., Lupenko S., Gotovych V., Shymchuk G., Shabliy N. 2020 10th International Conference on Advanced Computer Information Technologies (ACIT). 2020. P. 152-155.

13. Leschyshyn Y., Semchyshyn O. Periodically correlated heart rate variability detection by Neyman - Pearson criterion. 9th International Conference - The Experience of Designing and Applications of CAD Systems in Microelectronics. 2007. P. 139–140.

14. Barr M. Programming Embedded Systems: With C and GNU Development Tools. Sebastopol, CA : O'Reilly Media, 2006. P.45-78

15. Gonzalez R. C., Woods R. E. Digital Image Processing. Upper Saddle River, NJ : Pearson Prentice Hall, 2018. P.234-278

16. Луцків А., Лупенко С., Пасічник В. Паралельні та розподільнені обчислення. Навчальний посібник. Львів: Видавництво «Магнолія 2006», 2024. 566 с.

					КС КРБ 123.211.00.00 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

17. Kharchenko A., Bodnarchuk I., Yatcysyn V. The Method for Comparative Evaluation of Software Architecture with Accounting of Trade-offs. American Journal of Information Systems. 2014. Vol. 2, No. 1. P. 20-25.

18. Horowitz P., Hill W. The Art of Electronics. Cambridge, UK : Cambridge University Press, 2015. P.345-378

19. Кардаш І., Лецишин Ю., Варавін А. Критерії ефективності роботи для задачі моніторингу локальної мережі / Матеріали XI науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопіль: ТНТУ. 2023. С.154

20. Bentley J. Programming Pearls (2nd Edition). Reading, MA : Addison-Wesley Professional, 2000. P.45.78

21. FFmpeg Development Team. FFmpeg Documentation: Hardware Acceleration. FFmpeg Project. URL: <https://ffmpeg.org/ffmpeg.html> (date of access: 09.05.2025).

22. Gallmeister B. O. POSIX.4: Programming for the Real World. Sebastopol, CA : O'Reilly Media, 1995. P.267-298

23. Залісковий Ю., Лецишин Ю., Варавін А. Методи проведення моніторингу і аналізу мережевої інфраструктури Інтернет провайдерів, Матеріали XI науково-технічної конференції «Інформаційні моделі, системи та технології». Тернопіль: ТНТУ. 2023. С.152

24. Плухатиров І. Програмування мікроконтролерів STM32 у режимі GPIO, 2023. YouTube. URL: https://www.youtube.com/watch?v=KSJyuL4-6Ao&ab_channel=IhorPluhatyrov (дата звернення: 14.05.2025).

25. Желібо Є. П. Безпека життєдіяльності. навч. посіб. / Є. П. Желібо, Н. М. Заверуха, В. В. Зацарний. 6-те вид. Київ : Каравела, 2023. 344 с.

26. Про затвердження порядків надання домедичної допомоги особам при невідкладних станах: наказ МОЗ України від 09.03.2022 №441. URL: <https://zakon.rada.gov.ua/laws/show/z0356-22/conv#n800> (дата звернення: 03.06.2025)

					КС КРБ 123.211.00.00 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

27. Директива від 30.11.1989 № 89/654/ЄЕС Щодо мінімальних вимог стосовно безпеки і охорони здоров'я у робочих зонах (перша окрема Директива у значенні частини 1 статті 16 Директиви 89/391/ЄЕС). URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=73689 (дата звернення: 03.06.2025)

28. Сьомка С. В. Ергономіка та ергодизайн: підручник / С. В. Сьомка; за ред. М ва культури України. – Київ: Ліра К, 2022. – 618 с. – ISBN 978 617 7507 19 1.

29. НПАОП 0.00-1.28-10. Про затвердження правил охорони праці під час експлуатації електронно-обчислювальних машин. URL: https://dnaop.com/html/31562/doc-%D0%9D%D0%9F%D0%90%D0%9E%D0%9F_0.00-1.28-10 (дата звернення: 03.06.2025)

30. ДСТУ 7951:2015. Дизайн і ергономіка. Крісло оператора. Загальні ергономічні вимоги: нац. стандарт України / Укр. н.-д. ін-т дизайну та ергономіки. – Київ: УкрНДНЦ, 2016. – 10 с. URL: https://dnaop.com/html/61769/doc_ДСТУ_7951_2015 (дата звернення: 03.06.2025)

31. Про затвердження переліку професійних захворювань: постанова від Кабінету Міністрів України від 13.05.2020. URL: <https://zakon.rada.gov.ua/laws/show/1662-2000-%D0%BF#Text> (дата звернення: 03.06.2025)

					КС КРБ 123.211.00.00 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС

_____ Осухівська Г.М.

“ ____ ” _____ 2025 р

Комп'ютерна система оцифрування потокового відеосигналу для безпілотників

ТЕХНІЧНЕ ЗАВДАННЯ

на 9 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІ-41

_____ к.т.н., доц. Лещишин Ю.З.

_____ Коношенко Я.А.

« ____ » _____ 2025 р.

« ____ » _____ 2025 р.

Тернопіль 2025

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютерна система оцифрування потокового відеосигналу для безпілотників».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.211.00.00

1.2 Виконавець

Студентка групи СІ-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Коношенко Я.А.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/7-53 від 27.01.2025 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 27.01.2025 р.

Плановий термін завершення виконання кваліфікаційної роботи – 23.06.2025 р.

1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи – наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

Комп'ютерна система, що розробляється, призначена для:

- Передачі відеопотоку на наземну станцію з безпілотників.
- Оцифруванню даних для легшої подальшої обробки та роботу з ними.

2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка та налаштування комп'ютерної системи для оцифрування потокового сигналу для безпілотників на основі обчислювального модуля Raspberry Pi 4.

2.3 Характеристика об'єкту

Розробляювану комп'ютерну систему призначено для використання в системах моніторингу та безпеки.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

3.1.1 Вимоги до структури та функціонування системи

Комп'ютерна система повинна складатися з:

- Апаратного забезпечення.
- Програмного коду.

3.1.2 Вимоги до способів та засобів зв'язку між компонентами системи

Основна вимога, яка ставиться до способів та засобів інформаційного обміну – це їх узгодженість.

3.1.3 Вимоги до режимів функціонування системи

Для системи визначено два режими функціонування:

- нормальний режим функціонування;
- аварійний режим функціонування.

Основним режимом функціонування є нормальний режим.

Для забезпечення нормального режиму функціонування системи необхідно виконувати вимоги і дотримуватись умов експлуатації програмного забезпечення і комплексу технічних засобів системи, вказані у відповідних технічних документах (технічна документація, інструкції з експлуатації і т. д.).

Аварійний режим функціонування системи характеризується відмовою одного або декількох компонент програмного і (або) технічного забезпечення. При цьому функції роботи системи продовжують підтримувати роботу системи моніторингу в межах базових налаштувань.

3.1.4 Вимоги по діагностуванню системи

Для діагностування системи використовуються інструменти діагностування основних процесів системи, які вмонтовані в операційну

систему обчислювального модуля і програмне забезпечення, а також засоби для діагностики апаратного забезпечення.

Інструменти повинні забезпечувати зручний інтерфейс для можливості перегляду діагностичних подій, моніторингу процесу виконання програм.

3.1.5 Перспективи розвитку, проектування системи

Дана система може бути розширена завдяки використанню додаткових програмних компонентів, а також передбачено додавання елементів штучного інтелекту, а саме розпізнавання об'єктів та відстані до них.

3.2 Показники призначення

Система повинна передбачати можливість масштабування. Можливості масштабування повинні забезпечуватися засобами використовуваного базового програмного і технічного забезпечення.

3.2.1 Вимоги до надійності

Система повинна забезпечувати працездатність та відновлення своїх функцій при виникненні наступних ситуацій:

- при збоях в системі електропостачання апаратної частини;
- при помилках в роботі апаратних засобів;
- при помилках, пов'язаних з програмним забезпеченням (ОС і драйвери пристроїв).

Для захисту апаратури від стрибків напруги і комутаційних завад повинні застосовуватися мережні фільтри.

3.3 Вимоги до безпеки

Зовнішні елементи технічних засобів системи, що перебувають під напругою, повинні мати захист від випадкового дотику, а самі технічні засоби мати занулення або захисне заземлення .

Система електроживлення повинна забезпечувати захисне вимикання при перевантаженнях і коротких замиканнях в колах навантаження, а також аварійне ручне вимикання.

Загальні вимоги пожежної безпеки повинні відповідати нормам на побутове електрообладнання. У разі пожежі не мають виділятися отруйні гази і дим. Після зняття електроживлення має бути доступне застосування будь-яких засобів пожежогасіння.

3.3.1 Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Мікроклімат в приміщеннях повинен відповідати нормам виробничого мікроклімату по ДСН 3.3.6.042-99:

- температуру повітря в межах від +10°C до +35°C;
- відносну вологість повітря при 25°C в межах від 30% до 80%;
- атмосферний тиск 760 ± 25 мм рт. ст.

Періодичне технічне обслуговування використовуваних технічних засобів має проводитися відповідно до вимог технічної документації, але не рідше ніж один раз на рік.

Періодичне технічне обслуговування і тестування технічних засобів повинні включати обслуговування і тестування всіх використовуваних засобів, датчики, контролери, системи передачі даних, пристрої безперебійного живлення.

На підставі результатів тестування технічних засобів повинні проводитися аналіз причин виникнення виявлених дефектів і прийматися заходи по їх ліквідації.

3.4 Вимоги до захисту інформації від несанкціонованого доступу

Система повинна забезпечувати захист від несанкціонованого доступу на рівні не нижче встановленого вимогами, що пред'являються до категорії 1Д по класифікації документа, що діє, “Автоматизовані системи. Захист від несанкціонованого доступу до інформації. Класифікація автоматизованих систем”.

Компоненти підсистеми захисту від НСД повинні забезпечувати:

- ідентифікацію користувача;
- перевірку повноважень користувача при роботі з системою;
- розмежування доступу користувачів.

Рівень захищеності від несанкціонованого доступу засобів обчислювальної техніки, що здійснюють обробку конфіденційної інформації, повинен відповідати вимогам класу захищеності згідно вимогам документу “Засоби обчислювальної техніки. Захист від несанкціонованого доступу до інформації. Показники захищеності від несанкціонованого доступу до інформації”.

3.4.1 Вимоги по збереженню інформації при аваріях

Інформація, при виникненні аварійних ситуацій повинна бути збережена на резервних носіях.

3.4.2 Вимоги по стандартизації і уніфікації

Система повинна відповідати вимогам ергономіки і зручності користування за умови комплектування високоякісним обладнанням (ЕОМ, монітор і інше обладнання), що має необхідні сертифікати відповідності і безпеки.

3.4.3 Вимоги до функцій (завдань), що виконуються системою:

- забезпечення зручного інтерфейсу;
- забезпечення зберігання зв'язку;

- забезпечення якості передачі даних;
- забезпечення високої швидкодії.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:
 - а) Структурна схема.
 - б) Блок схема роботи комп'ютеризованої системи.
 - в) Схема електрична принципова.
 - г) Схема електрична монтажна (з'єднань).

*Примітка: У комплект документації можуть вноситися зміни та доповнення в процесі розробки.

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання.	27.01.2025р. – 02.02.2025р.
2	Робота над розділом аналіз технічного завдання на розробку комп'ютерної системи оцифрування потокового відеосигналу для безпілотників	02.02.2025р. – 03.03.2025р.
3	Робота над основним розділом проектування комп'ютерної системи оцифрування потокового відеосигналу для безпілотників	04.03.2025р. – 01.05.2025р.
4	Робота над основним розділом програмування та тестування комп'ютерної системи оцифрування потокового відеосигналу для безпілотників	02.05.2025р. – 01.06.2025р.
5	Безпека життєдіяльності, основи охорони праці	02.06.2025р. – 06.06.2025р.

6	Оформлення пояснювальної записки і графічного матеріалу	06.06.2025р. – 09.06.2025р.
7	Перевірка на академічний плагіат, перевірка керівником та консультантами	9.06.2025р. – 15.06.2025р.
8	Попередній захист кваліфікаційної роботи бакалавра	16.06.2025р. – 22.06.2025р.
9	Захист кваліфікаційної роботи бакалавра	26.06.2025р.

6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

Додаток Б

Лістинг програми

```
#ifndef FIFO_H
#define FIFO_H 20191213

#include "board.h"

typedef struct FIFO_ { //struct
    S32 in; // Next In Index
    S32 out; // Next Out Index
    U8 *buf; // Buffer
    S32 buf_size; // Size buffer
    INT item_size; // Size in bytes for each element in the
buffer.
} FIFO;

#ifdef __cplusplus
extern "C" {
#endif

/**
 * @brief fifo_init_static - Init a statically allocated FIFO
buffer
 * @param fifo - The pointer to the structure holding FIFO
data
 * @param pbuf - Pointer to the memory used to store actual
fifo items
 * @param fifo_size - The number of items to store on the
FIFO
 * @param item_size - The size in bytes of each item on the
buffer
 * @return - @p TRUE on success, @p FALSE otherwise
 */
BOOL _fifo_init_static( FIFO *fifo, VOID *pbuf, U32
fifo_size, U32 item_size );

/**
 * @brief fifo_flush
 * @param fifo
 * @return - TRUE or FALSE
 */
BOOL _fifo_flush( FIFO *fifo );

/**
```

```

    * @brief fifo_available
    * @param fifo
    * @return - TRUE or FALSE
    */
S32 _fifo_available( FIFO *fifo );

/**
 * @brief fifo_free
 * @param fifo
 * @return - TRUE or FALSE
 */
S32 _fifo_free( FIFO *fifo );

/**
 * @brief fifo_add
 * @param fifo
 * @param item
 * @return - TRUE or FALSE
 */
BOOL _fifo_add( FIFO *fifo, const VOID *item );

/**
 * @brief fifo_get
 * @param fifo
 * @param item
 * @return - TRUE or FALSE
 */
BOOL _fifo_get( FIFO *fifo, VOID *item );

/**
 * @brief fifo_adds
 * @param fifo
 * @param item
 * @param num
 * @return - TRUE or FALSE
 */
BOOL _fifo_adds( FIFO *fifo, const void *item, INT num );

/**
 * @brief fifo_gets
 * @param fifo
 * @param item
 * @param num
 * @return - TRUE or FALSE
 */
BOOL _fifo_gets( FIFO *fifo, VOID *item, INT num );

```

```

/**
 * @brief test
 */
void _fifo_test( VOID );

#ifdef    __cplusplus
}
#endif

#endif    /* FIFO_H */

#include "_fifo.h"
#include "board.h"

void *_memcpy( void *dst, const void *src, INT n )
{
    test_param( NULL == dst );
    test_param( NULL == src );
    test_param( n < 1 );

    register U8 *pdst = (U8 *)dst;
    register U8 *psrc = (U8 *)src;

    for( ; 0 < n; n-- )
    {
        *(U8 *)pdst++ = *(U8 *)psrc++;
    }

    return pdst;
}

#define FIFO_2X                1 // fast mode

// #define FIFO_ALIGN(value, alignment) ((value + alignment -
1) & ~(alignment - 1))
// #define ADVANCE_QUEUE(index, size, max_size) ((index +
size) % max_size)

BOOL _fifo_init_static( FIFO *fifo, void *pbuf, U32
fifo_size, U32 item_size )
{
    //Sanity check
    test_param( NULL == fifo );
    test_param( NULL == pbuf );
    test_param(fifo_size < 2 ); // Size is too small. It
must be larger than 2
    #if FIFO_2X
        test_param((fifo_size & (fifo_size - 1)) != 0 ); // Must
be a power of 2
    #endif
}

```

```

#endif
    test_param( item_size > fifo_size );
    {
        //return FALSE;
    }
    fifo->buf = (U8 *)pbuf;
    fifo->buf_size = fifo_size;
    fifo->item_size = item_size;
    _fifo_flush( fifo );
    return TRUE;
}

BOOL _fifo_flush( FIFO *fifo )
{
    test_param( NULL == fifo );
    fifo->in = 0;
    fifo->out = fifo->in;
    return TRUE;
}

S32 _fifo_available( FIFO *fifo )
{
    test_param( NULL == fifo );
    S32 i = ( fifo->in - fifo->out );
    if( i < 0 )
    {
        i = -i;
    }
    return i;
}

S32 _fifo_free( FIFO *fifo )
{
    test_param( NULL == fifo );
    return ( fifo->buf_size - _fifo_available( fifo ) );
}

BOOL _fifo_add( FIFO *fifo, const void *item )
{
    U32 offset;

    test_param( NULL == fifo );
    test_param( NULL == item );
    if( _fifo_free( fifo ) >= fifo->item_size )
    {
        offset = fifo->in & ( fifo->buf_size - 1 );
        fifo->in += fifo->item_size;
        _memcpy( fifo->buf + offset,
                item,

```

```

        fifo->item_size );
    return TRUE;
}
return FALSE;
}

BOOL _fifo_get( FIFO *fifo, void *item )
{
    U32 offset;

    test_param( NULL == fifo );
    test_param( NULL == item );

    if( _fifo_available( fifo ) >= fifo->item_size )
    {
        offset = fifo->out & ( fifo->buf_size - 1 );
        fifo->out += fifo->item_size;
        _memcpy( item, fifo->buf + offset, fifo->item_size
);
        return TRUE;
    }
    return FALSE;
}

BOOL _fifo_adds( FIFO *fifo, const void *item, INT num )
{
    register INT j, offset;
    register U8 *p = (U8 *)item;
    BOOL ret = FALSE;

    test_param( NULL == fifo );
    test_param( NULL == item );
    test_param( num < 0 );

    if( _fifo_free( fifo ) >= ( fifo->item_size * num ) )
    {
        for( j = 0; j < num; j++ )
        {
            offset = fifo->in & ( fifo->buf_size - 1 );
            fifo->in += fifo->item_size;

            _memcpy( fifo->buf + offset,
                (U8 *) ( p + j * fifo->item_size ),
                fifo->item_size );
        }
        ret = TRUE;
    }

    return ret;
}

```

```

    BOOL _fifo_gets( FIFO *fifo, void *item, INT num )
    {
        register INT j, offset;
        U8 *p = (U8 *)item;
        BOOL ret = FALSE;

        test_param( NULL == fifo );
        test_param( NULL == item );
        test_param( num < 1 );

        if( _fifo_available( fifo ) >= ( fifo->item_size * num
) )
        {
            for( j = 0; j < num; j++ )
            {
                offset = fifo->out & ( fifo->buf_size - 1 );
                fifo->out += fifo->item_size;

                _memcpy( p + j * fifo->item_size,
                        fifo->buf + offset,
                        fifo->item_size );
            }
            ret = TRUE;
        }

        return ret;
    }

void _fifo_test( void )
{
    U8 test_buf[ 64 ];
    FIFO fifo;
    U8 fifo_buf[ 32 ];
    INT i;

    if( TRUE != _fifo_init_static( &fifo, fifo_buf, 32, 1 )
)
    {
        while( 1 ){};
    }

    test_buf[ 0 ] = 0; //clear before
    if( TRUE == _fifo_get( &fifo, &test_buf[ 0 ] ) )
    {
        while( 1 ){};
    }

    test_buf[ 0 ] = 8;
    if( TRUE != _fifo_add( &fifo, &test_buf[ 0 ] ) )
    {

```

```

        while( 1 ){};
    }

    test_buf[ 0 ] = 0; //clear before
    if( TRUE != _fifo_get( &fifo, &test_buf[ 0 ] ) )
    {
        while( 1 ){};
    }

    for( i = 0; i < 5; i++ ) //init before
    {
        test_buf[ i ] = i + 1;
    }
    if( TRUE != _fifo_adds( &fifo, &test_buf[ 0 ], 5 ) )
    {
        while( 1 ){};
    }

    i = _fifo_available( &fifo );

    for( i = 0; i < 5; i++ ) //clear before
    {
        test_buf[ i ] = 0;
    }
    if( TRUE == _fifo_gets( &fifo, &test_buf[ 0 ], 8 ) )
    {
        while( 1 ){};
    }
    if( TRUE != _fifo_gets( &fifo, &test_buf[ 0 ], 5 ) )
    {
        while( 1 ){};
    }
    else
    {
        for( i = 0; i < 5; i++ )
        {
            if( test_buf[ i ] != ( i + 1 ) )
            {
                while( 1 ){};
            }
        }
    }

    //max limit test
    if( TRUE == _fifo_adds( &fifo, &test_buf[ 0 ], 100 ) )
    {
        while( 1 ){};
    }

    // overflow test
    for( i = 0; i < 32; i++ ) //init before
    {
        test_buf[ i ] = i + 1;
    }

```

```

    }
    if( TRUE != _fifo_adds( &fifo, &test_buf[ 0 ], 31 ) )
    {
        while( 1 ){};
    }
    if( TRUE != _fifo_gets( &fifo, &test_buf[ 0 ], 30 ) )
    {
        while( 1 ){};
    }
    if( TRUE != _fifo_adds( &fifo, &test_buf[ 0 ], 31 ) )
    {
        while( 1 ){};
    }
    if( TRUE != _fifo_gets( &fifo, &test_buf[ 0 ], 30 ) )
    {
        while( 1 ){};
    }
    if( TRUE != _fifo_adds( &fifo, &test_buf[ 0 ], 30 ) )
    {
        while( 1 ){};
    }
    if( TRUE != _fifo_gets( &fifo, &test_buf[ 0 ], 30 ) )
    {
        while( 1 ){};
    }
    }

// 2byte
if( TRUE != _fifo_init_static( &fifo, fifo_buf, 32, 2 )
)
{
    while( 1 ){};
}
for( i = 0; i < ( 32 / 2 ); i += 2 ) //init before
{
    test_buf[ i + 0 ] = i / 2 + 1;
    test_buf[ i + 1 ] = 0;
}
if( TRUE != _fifo_adds( &fifo, &test_buf[ 0 ], 7 ) )
{
    while( 1 ){};
}
if( TRUE != _fifo_gets( &fifo, &test_buf[ 0 ], 6 ) )
{
    while( 1 ){};
}
if( TRUE != _fifo_adds( &fifo, &test_buf[ 0 ], 7 ) )
{
    while( 1 ){};
}
if( TRUE != _fifo_gets( &fifo, &test_buf[ 0 ], 6 ) )
{
    while( 1 ){};
}
}

```

```

if( TRUE != _fifo_adds( &fifo, &test_buf[ 0 ], 7 ) )
{
    while( 1 ){};
}
if( TRUE != _fifo_gets( &fifo, &test_buf[ 0 ], 7 ) )
{
    while( 1 ){};
}

// 3byte
if( TRUE != _fifo_init_static( &fifo, fifo_buf, 32, 3 )
)
{
    while( 1 ){};
}
for( i = 0; i < ( 32 / 3 ); i += 3 ) //init before
{
    test_buf[ i + 0 ] = i / 3 + 1;
    test_buf[ i + 1 ] = 0;
    test_buf[ i + 2 ] = 0;
}
if( TRUE != _fifo_adds( &fifo, &test_buf[ 0 ], 4 ) )
{
    while( 1 ){};
}
if( TRUE != _fifo_gets( &fifo, &test_buf[ 0 ], 3 ) )
{
    while( 1 ){};
}
if( TRUE != _fifo_adds( &fifo, &test_buf[ 0 ], 4 ) )
{
    while( 1 ){};
}
if( TRUE != _fifo_gets( &fifo, &test_buf[ 0 ], 3 ) )
{
    while( 1 ){};
}
if( TRUE != _fifo_adds( &fifo, &test_buf[ 0 ], 4 ) )
{
    while( 1 ){};
}
if( TRUE != _fifo_gets( &fifo, &test_buf[ 0 ], 3 ) )
{
    while( 1 ){};
}
}

```

```

import RPi.GPIO as GPIO
from time import sleep

```

```

WD_PIN = 27 #6
SLEEP = 0.5

```

```

UP = 1
DOWN = 0

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False) # Turn off warnings output
GPIO.setup(WD_PIN, GPIO.OUT) # Set GPIO to output

print(GPIO.input(WD_PIN))

try:
    while True:
        GPIO.output(WD_PIN, UP) # set GPIO to
1/GPIO.HIGH/True
        print(GPIO.input(WD_PIN))
        sleep(SLEEP) # wait half a second
        GPIO.output(WD_PIN, DOWN) # set GPIO to
0/GPIO.LOW/False
        print(GPIO.input(WD_PIN))
        sleep(SLEEP) # wait half a second

    except KeyboardInterrupt: # trap a CTRL+C keyboard
interrupt
        GPIO.cleanup()

import RPi.GPIO as GPIO
from time import sleep
import os

POFF_PIN = 22
DOWN = 0
UP = 1

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False) # Turn off warnings output
GPIO.setup(POFF_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP) #
Set GPIO to input
sleep(2)
print(GPIO.input(POFF_PIN))

def shutdown(channel):
    sleep(0.5)
    if GPIO.input(POFF_PIN) == UP:
        print("Poweroff ...")
        os.system("sudo poweroff")

GPIO.add_event_detect(POFF_PIN, GPIO.RISING,
callback=shutdown, bouncetime=2000)

while True:
    sleep(1)

```

Додаток В

Перелік елементів

<i>Поз. позначення</i>	<i>Найменування</i>	<i>Кіл.</i>	<i>Примітка</i>
	<u>Конденсатори</u>		
C3, C5, C13	CC0603KRX7R9BB104, 100nF 50V ±10%	3	
C1, C2, C6, C8, C10, C11	CC0603MRX5R7BB106, 16V 10uF ±20%	6	
C4	CL10B103KB8NNNC, 50V 10nF ±10%	1	
C12, C14	GRM188R61E106MA73J, 25V 10uF ±20%	2	
C15	T491C227K010AT, 220uF 10V ±10%	1	
	<u>Коннектори</u>		
CM1	DF40C-100DS-0.4, Коннектор ROHS 0.4 мм	1	
P6	2314844-6, 6-позиційна розетка, нижній або верхній вхідний роз'єм 0,079" (2,00 мм)	1	
P4, P7	2314844-4, 4-позиційна розетка, нижній або верхній вхідний роз'єм 0,079" (2,00 мм)	2	
P9, XT1	10051922-1210ELF, 3'єднувач FPC нижній 12 поз 0.50 мм	2	
P5	SFW15R-1STE1LF, 15-позиційний коннектор FFC, FPC, нижній роз'єм 0,039" (1,00 мм)	1	
P7, P8	2314844-8, 8-позиційна розетка, нижній або верхній вхідний роз'єм 0,079" (2,00 мм)	2	
P1	gb042-34p-h10, Коннектор 0.4 мм 34 позицій	1	
P6	F32R-1A7H1-11006, 3'єднувач FFC нижній 6 поз 0,5мм	1	
	<u>Світлодіоди</u>		
D2, D5	LTST-C190KGKT, 567.5нм~576.5нм Жовтий Зелений	2	
D6	LTST-C190KRKT, 631нм Дискретний ROHS	1	
D1	LTST-C190YKT, 588 нм Жовтий	1	
	<u>Діоди</u>		
D5	USBLC6-2SC6, TB-діод 5,25 В, 17 В, SOT23-6	1	
IC1, IC2	TPD4EUSB30DQAR, TB-діод 5,5 В, 8 В, 10 мкС	2	

					КС КРБ 123.211.00.01 ПЕ			
Зм	Арк	№ докум	Підпис	Дата	Комп'ютерна система оцифрування потокового відеосигналу для безпілотників	Літ	Аркуш	Аркушів
Розроб.	Коношенко Я.А.					Н		
Перевір.	Лецишин Ю.З.							
Консульт.								
Н. контр.	Туцик Н.С.							
Зав. каф.	Осухівська Г.М.				Перелік елементів	ТНТУ, каф. КС, гр. СІ-41		

<i>Поз. позначення</i>	<i>Найменування</i>	<i>Кіл.</i>	<i>Примітка</i>
	<u>Індуктори</u>		
L1	LQH43CN2R2M03L, 900 мА 2,2 мкГн ±20%	1	
	<u>Резистори</u>		
R5, R6, R9, R10, R15, R16, R19, R21	RC0603JR-1310KL, 75В ±1% 10 кОм 0603	8	
R2, R3, R14, R15	RC0603JR-071KL, 75В ±1% 1 кОм 0603	4	
R11	RC0603FR-0722K1L, 75В ±1% 22.1 кОм 0603	1	
R4, R12	RC0603JR-07220RL, 75В ±1% 220 Ом 0603	2	
R17, R18	RC0603JR-0727RL, 75В ±1% 27 Ом 0603	2	
R14	RC0603JR-07240RL, 75В ±1% 240 Ом 0603	1	
R1, R24, R25	RC0603JR-074K7L, 75В ±1% 4.7 кОм 0603	3	
R22	RC0603JR-072K2L, 75В ±1% 2.2 кОм 0603	1	
R20, R23	RC0603FR-075K1L, 75В ±1% 5.1 кОм 0603	2	
R13	RC0603JR-07100RL, 75В ±1% 100 Ом 0603	1	
	<u>Перемикачі</u>		
SA1	KLS7-MSS-1290BP, 0,3 А 4 В 6.7x2.8x1.4 мм	1	
SW1	B3U-3000P, 0.05А 12В 3x2.5x1.2 мм	1	
	<u>Трансформатори</u>		
T1	SM42P01EL, 230 мкГн Трансформатор ПОЕ	1	
	<u>Таймери</u>		
U1	TPL5010DDCR, SOT-23-Thin-6 Таймер ROHS	1	
	<u>Перетворювачі</u>		
U2	MT3410L, Вихідний струм 1,3 А, вхідна напруга 2,3-7 В, частота перемикання 1,5 МГц	1	
	<u>Транзистори</u>		
VT1	IRLML0030TRPBF, 30В 5.3А 1.3Вт 27 мОм@10В, 4.5В N-канальний	1	
Зм	Арк	№ докум	Підпис
			Дата
			КС КРБ 123.211.00.01 ПЕ
			Арк
			2

[illegible]