

Ф29 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'ютеризована система для створення візуальних ефектів у кіно за допомогою маркерного методу захоплення руху актора.*

Виконав: студент 4 курсу, групи СІ-41

спеціальності 123 «Комп'ютерна інженерія»

Ковальський Олександр Станіславович

(шифр і назва спеціальності)

(підпис)

Ковальський О.С

(прізвище та ініціали)

Керівник

(підпис)

Жаровський Р.О

(прізвище та ініціали)

Нормоконтроль

(підпис)

Луцик Н.С

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

(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 Ковальського Олександра Станіславовича
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система для створення візуальних ефектів у кіно за допомогою маркерного методу захоплення руху актора.

Керівник роботи Жаровський Руслан Олегович, к.т.н доцент кафедри КС
(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 16.06.2025

3. Вихідні дані до роботи Технічне завдання

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

Вступ. 1. Аналіз технічного завдання

2. Розробка та аналіз вибраних структур

3. Програмна реалізація системи виявлення руху

4. Безпека життєдіяльності, основи охорони праці.

Висновки

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

1. Структурна схема системи

2. Схема електрична принципова

3. Блок-схема алгоритму роботи алгоритму програмного забезпечення

4. Модель тестового макету

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</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 – 02.02</i>	<i>Викон.</i>
2.	<i>Підбір джерел про створення вузільних ефектів у кіно за допомогою маркерного методу та захоплення руху</i>	<i>11.05.2025-20.05.2025</i>	<i>Викон.</i>
3.	<i>Опрацювання джерел по темі кваліфікаційної роботи</i>	<i>01.05.2025</i>	<i>Викон.</i>
4.	<i>Розробка структури та аналіз вибраного приладу</i>	<i>05.06.2025</i>	<i>Викон.</i>
5.	<i>Розробка програмного коду</i>	<i>07.06.2025</i>	<i>Викон.</i>
6.	<i>Тестування та оцінка приладу для захоплення руху</i>	<i>11.06.2025</i>	<i>Викон.</i>
7.	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>12.06.2025</i>	<i>Викон.</i>
8.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>20.06.2025</i>	<i>Викон.</i>
9.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>09.06.2025-15.06.2025</i>	<i>Викон.</i>
10	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>16.06.2025-22.06.2025</i>	<i>Викон.</i>
11	<i>Захист кваліфікаційної роботи бакалавра</i>	<i>23.06.2025</i>	<i>Викон.</i>

Студент

(підпис)

Ковальський О.С

(прізвище та ініціали)

Керівник роботи

(підпис)

Жаровський Р.О

(прізвище та ініціали)

АНОТАЦІЯ

Ковальський О.С. Комп'ютеризована система для створення візуальних ефектів у кіно за допомогою маркерного методу захоплення руху актора: робота на здобуття кваліфікаційного ступеня бакалавра: спец. 123 — комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2025.

Ключові слова: захоплення руху, маркерний костюм, Raspberry Pi 4, OpenCV, Python, візуальні ефекти, Blender, електронна схема

У кваліфікаційній роботі розглянуто процес розробки комп'ютеризованої системи для створення візуальних ефектів у кіно шляхом використання маркерного методу захоплення руху актора. В роботі виконано аналіз сучасних методів трекінгу руху, обґрунтовано вибір апаратної платформи Raspberry Pi 4, розроблено структурну схему системи, а також алгоритми запису та обробки анімаційних даних. Особливу увагу приділено розробці програмного забезпечення, що реалізує обробку відеопотоку з камери, виявлення маркерів та побудову координат руху. Система протестована в умовах моделювання реальної зйомки, результати підтверджують працездатність і точність розпізнавання. Отримані координати успішно використані для імпорту в 3D-середовище Blender. Розробка може бути інтегрована у практичні задачі створення VFX-контенту, анімації та освітні проєкти.

ANNOTATION

Kovalskiy O.S. A computerized system for creating visual effects in film using the marker method of capturing the actor's motion: Bachelor's Graduation Thesis: speciality 123 — computer engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2025.

Keywords: motion capture, marker suit, Raspberry Pi 4, OpenCV, visual effects, Python, Blender, coordinate analysis, 3D animation.

This bachelor qualification work presents the development of a computerized system for generating visual effects in film production using a marker-based motion capture method. The research includes analysis of modern motion tracking technologies, substantiates the choice of the Raspberry Pi 4 platform, and presents the design of the system's architecture and data processing algorithms. Particular attention is given to software development for capturing video stream, detecting motion markers, and calculating movement coordinates. The system was tested in conditions simulating real filming, and the results confirm its functionality and accuracy. The captured coordinates were successfully imported into the 3D environment Blender for visualization and animation. The proposed solution can be integrated into VFX production workflows, animation pipelines, or educational projects.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	
1.1 Методи і засоби, алгоритми захоплення	5
1.2 Аналіз існуючих методів.....	7
1.3 Вимоги до системи захоплення	11
РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА.....	15
2.1 Розробка та аналіз структури.....	15
2.2 Обґрунтування вибору обладнання	17
2.2.1 Розробка і аналіз структури системи	20
2.2.2 Обґрунтування засобів захоплення та об'єкта спостереження	23
2.2.3 Канали комунікації	26
2.3 Розробка і аналіз електричної схеми	29
2.3.1 Структура загальної електричної схеми	30
2.3.2 Підключення камери	31
2.3.3 Схема живлення системи	32
2.4 Вибір необхідного програмного забезпечення для реалізації проєкту	36
2.4.1 Операційна система	36
2.4.2 Програмні бібліотеки для камери	37
2.4.3 Мова програмування та бібліотеки для обробки.....	38
2.4.4 Blender та постобробка.....	39

					КС КРБ 123.209.00.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Ковальський О.С			Комп'ютеризована система для створення візуальних ефектів у кіно за допомогою маркерного методу захоплення руху актора.			Літ.	Арк.	Аркушів	
Перевір.		Жаровський Р.О								6	
Реценз.								ТНТУ, каф. КС, гр. СІ-41			
Н. Контр.		Луцик Н.С									
Затверд.		Осухівська Г.М.									

2.4.5	FFmpeg та конвертація	40
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА.....		
3.1	Опис алгоритму роботи Raspberry Pi	41
3.2	Розробка програмного коду	44
3.3	Тестування та принцип роботи.....	48
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.....		
4.1	Ергономічні проблеми безпеки життєдіяльності	53
4.2	Профілактика захворювань, які викликані напруженням органів, систем організму та невірним положенням тіла при роботі.	55
ВИСНОВКИ.....		57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		58
Додаток А Технічне завдання		
Додаток Б Лістинг коду		
Додаток В Перелік елементів		

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

СПИСОК СКОРОЧЕНЬ

GPU - Graphics Processing Unit – графічний процесор

USB - Universal Serial Bus – універсальна послідовна шина

CSI - Camera Serial Interface – послідовний інтерфейс камери

HDMI - High-Definition Multimedia Interface – мультимедійний інтерфейс високої чіткості

GPIO - General Purpose Input/Output – загального призначення входи/виходи

SSD - Solid State Drive – твердотільний накопичувач

VFX - Visual Effects – візуальні ефекти

3D - Three-Dimensional – тривимірний

FFC - Flexible Flat Cable – гнучкий плоский кабель

I/O - Input/Output – введення/виведення

OS - Operating System – операційна система

API - Application Programming Interface – інтерфейс прикладного програмування

IDE - Integrated Development Environment – інтегроване середовище розробки

CSV - Comma-Separated Values – формат даних, розділених комами

FPS - Frames Per Second – кадрів за секунду

LED - Light-Emitting Diode – світлодіод

IR - Infrared – інфрачервоний

HAT - Hardware Attached on Top – плата-розширення для Raspberry Pi

					КС КРБ 123.209.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Кіно та комп'ютерні ігри усе ширше відіграють ключову роль у розвитку кіноіндустрії. Особливого значення набувають комп'ютеризовані системи створення візуальних ефектів (VFX), які забезпечують інтеграцію реального та віртуального контенту в одне ціле. Одним з найефективніших способів реалізації реалістичної 3D-анімації є метод захоплення руху (Motion Capture), що дозволяє точно відтворити рухи актора та перенести їх на віртуального персонажа.

Серед різних методів трекінгу особливої уваги заслуговує маркерний метод, який передбачає використання спеціального костюма з маркерами, прикріпленими до ключових точок тіла актора. Цей метод забезпечує високу точність зчитування рухів та є широко застосовуваним у професійному кіновиробництві. Однак вартість комерційних систем захоплення руху залишається високою, що стимулює розвиток доступніших, спрощених варіантів з використанням відкритих апаратних і програмних платформ.

Актуальність теми полягає в необхідності створення бюджетної, адаптивної та придатної для освітніх і незалежних студій системи захоплення руху актора з подальшим використанням у створенні візуальних ефектів. У межах цієї кваліфікаційної роботи розробляється прототип такої системи з використанням мікрокомп'ютера Raspberry Pi, камери та програмного середовища Blender.

Метою роботи є створення комп'ютеризованої системи, здатної фіксувати рух актора за допомогою маркерного костюма, обробляти отримані дані та інтегрувати їх у візуальні сцени. Для досягнення цієї мети передбачено аналіз технічних рішень, проектування апаратної структури, розробку алгоритмів захоплення й обробки руху, а також створення сцени з візуальними ефекта

					КС КРБ 123.209.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Методи і засоби, алгоритми захоплення

Захоплення руху є ключовим етапом у створенні реалістичних візуальних ефектів, особливо у випадках, коли необхідно точно передати рухи живого актора на віртуального. Така технологія широко щастовосується у кінематографі, відеоіграх, а також у медичних та спортивних дослідженнях для аналізу біомеханіки. Основна мета полягає у фіксації координат ключових точок тіла актора в реальному часі та подальшому перенесенні цих даних у фикрових формат, придатний для 3D-анімації.

Існує декілька основних підходів до реалізації систем захоплення руху, кожен з яких має власні переваги та обмеження. Найпоширеніші методи включають оптичний, інерційний та гібридний. У межах цього проєкту акцент зроблено на оптичному маркерному методі, як на найзбалансованішому з точки зору точності, вартості та реалізованості в умовах обмеженого бюджету та технічних можливостей.

Оптичний маркерний метод базується на реєстрації світловідбивних або чорно-білих маркерів, які кріпляться на тіло актора на важливі точки, а саме на суглоби, кінцівки, грудну клітку, голові. Камера фіксує відеопотік, у якому розпізнавання можуть використовуватись ІЧ-підсвітка, однотонний фон та висока частота кадрів. Обробка відео здійснюється за допомогою спеціалізованих алгоритмів, які дозволяють відслідковувати координати кожного маркера у кадрі, визначати їх положення, а також асоціювати їх з певними частинами тіла актора.

Для реалізації таких алгоритмів використовується засоби

					КС КРБ 123.209.41.00 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Ковальський О.С			Аналіз технічного завдання, методи і засоби захоплення рухів			Літ.	Арк.	Аркушів	
Перевір.		Жаровський Р.О								10	
Реценз.								ТНТУ, каф. КС, гр. СІ-41			
Н. Контр.		Луцик Н.С									
Затверд.		Осухівська Г.М.									

комп'ютерного зору, зокрема бібліотека OpenCV, яка має вбудовані функції для детекції шаблонних маркерів, таких як ArUco. Алгоритм детекції складається з кількох послідовних етапів: спочатку відеокарт конвертується у відтінки сірого для зменшення обчислювального навантаження, потім виконується бінаризація зображення та пошук контурів. З-поміж знайдених контурів виділяється прямокутні області, які порівнюються з шаблонами маркерів. У разі збігу обчислюється центр маркера та його ID.

Отримані координати кожного маркера зберігаються у вигляді масиву, де для кожного кадру зазначено ідентифікатор маркера та його положення у пікселях(x, y). У разі використання декількох камер можна додатково обчислювати глибину координату z методом

Триангуляції, що дозволяє перейти від 2D до 3D-моделі руху. Для спрощення реалізації в рамках кваліфікаційної роботи проєкту буде розроблено 2D-дані, які в подальшому можна використовувати у Blender для анімації гляхом покадрового вставлення ключових положень об'єкта або скелета.

Ще однією важливою частиною системи є алгоритми фільтрації та згладжування траєкторій. Через можливі шуми, втрату маркерів у деяких кадрах або помилки детекції, отримані дані можуть бути нестабільними. Для виправлення таких артефактів застосовуються числові фільтри, наприклад, ковзне середнє або Kalman-фільтр, які дозволяють згладити рух маркера та зробити траєкторію природнішою.

Таким чином, побудова системи захоплення руху за маркерними методом включає в себе як апаратну частину (камера, Raspberry Pi, освітлення), так і програмну (детекцію маркерів, обчислення координат, збереження у файл, фільтрація даних). Цей підхід дозволяє з мінімальними витратами створити прототип системи MoCap, що може бути використаний для навчальних, дослідницьких або навіть напівпрофесійних цілей.

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1.2 Аналіз існуючих методів

Існуючі системи захоплення руху значно відрізняються за своїм принципом дії, вартістю, точністю, зручністю використання та сферою застосування. Найбільш поширеними є оптичні, інерційні та гібридні методи трекінгу.

Оптичні маркерні системи вважаються ідеальним стандартом у професійному кіновиробництві. Вони передбачають використання кількох високошвидкісних інфрачервоних камер, які зчитують положення світловідбивних маркерів, розташованих на костюмі актора. Прикладом є системи компаній Vicon або OptiTrack. Наприклад, система Vicon Vantage V16 використовує до 16 камер з частотою до 250 кадрів на секунду та роздільною здатністю до 4 Мп. Точність позиціонування досягає 0.3 мм при правильному калібруванні. Основним недоліком такого підходу є його висока вартість, потреба в спеціально підготовленому приміщенні та значні вимоги до обладнання.

Безмаркерні оптичні системи, наприклад, яку Microsoft Kinect або інші шлибинні камери, використовують ІЧ-сенсори для створення карти глибини. Це дозволяє працювати без маркерів, однак точність позиціонування нижча (похибка може сягати 1-2 см), що обмежує застосування таких систем у високоточних завданнях. Крім того, вони погано працюють у яскраво освітлених умовах або на відкритому повітрі.

Інерційні системи базуються на використанні IMU (інерціальних вимірювальних одиниць), які вміщують акселерометри, гіроскопи та магнітометри. Кожен датчик кріпиться до певної частини тіла актора. Зібрані дані обробляються для побудови моделі руху. Відомі представники – Xsens, Rokoko Smartsuit Pro. Наприклад, система Xsens MVN Link складається із 17 сенсорів, які знімають дані з частотою до 240 Гц, з похибкою у 1-3 см у позиціонуванні. Основною перевагою інерційного підходу є мобільність – не

					КС КРБ 123.209.00.00 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

потрібне студійне освітлення чт камери, можна знімати навіть на вулиці. Однак такі системи схильні до накопичення похибки через дрейф датчиків, що вимагає частого калібрування та застосування фільтрів.

Гібридні системи намагаються поєднати переваги оптичних та інерційних. Наприклад, поєднання OpenCV з Raspberry Pi Camera дозволяє реалізувати недорогу систему, яка фіксує положення маркерів через зображення, а при потребі може комбінуватися з IMU-модулями для уточнення положення у 3D-просторі. Такі рішення активно розвиваються в DIY-спільнотах. Прикладом є проєкти на базі ArUco-маркерів, OpenMoCap або DeepLabCut, де обробка даних може виконуватись на Raspberry PI, а візуалізація – у Blender або Unity.

Потрібно звертати увагу не лише на якість кінцевого трекінгу, а й на технічну реалізованість рішення. Рівень доступу до API, можливість

Забезпечення заданих технічних характеристик комп'ютеризованої системи USB аудіоплеєра з підтримкою веб-еквалайзера було досягнуто шляхом комплексного підходу до вибору компонентів та реалізації програмно-апаратних рішень. обробки даних у режимі реального часу, сумісність з вільним ПЗ, ресурсомісткість обробки зображень, підтримка Python та можливість кастомізації під конкретні потреби. Прикладом візьмем OpenCV, який дозволяє швидко прототипувати алгоритми детекції маркерів та обробки відео, тоді як використання Raspberry Pi 4 як базової обчислювальної платформи забезпечує прийнятну продуктивність при мінімальних витратах.

Виходячи з аналізу, стає понятно, що хоча професійні системи забезпечують найвищу якість, вони є фінансово недосяжними для індивідуального проєкту. Інерційні системи зручні, але вимагають доступу до дорогих сенсорів. Найбільш виправданим варіантом є реалізація спрощеної оптичної системи на базі Raspberry Pi 4 та OpenCV. Такий підхід дозволяє зосередитися на ключових етапах – зчитування, обробці та передачі координат

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

з одночасним вивченням технічних деталей комп'ютерного зору, мікроконтролерів та візуалізації.

1.3 Вимоги до системи захоплення руху

Проектування комп'ютеризованої системи для створення візуальних ефектів на основі маркерного методу захоплення руху потребує чіткого визначення функціональних, технічних і програмних вимог. Для досягнення відвідності цілям, система повинна бути технологічно досяжною, бюджетною, відкритою до розширення й інтегрованою у наявне середовище 3D-анімації. Всі вимоги умовно поділяються на декілька груп, а саме: функціональні, технічні, програмні та експлуатаційні.

Функціональні вимоги визначають ті завдання, які система повинна вирішувати безпосередньо в процесі своєї роботи. Основним призначенням системи є відстеження рухів людини, яка одягнена в маркерний костюм, шляхом зчитування положення маркерів, розміщених на ключових частинах тіла. Дані, отримані в результаті такого захоплення, мають бути придатними для подальшої обробки та інтеграції у 3D-графічні середовища, такі як Blender або Unreal Engine. Для досягнення зазначеної мети, система повинна забезпечувати такі функціональні можливості як: Розпізнавання маркерів на відеозображенні з достатнім рівнем точності, навіть при частковому перекритті або зміні освітлення, автоматичне визначення координат кожного маркера у кожному кадрі відеопотоку, введення запису руху у формі таблиці або бази даних, де кожен рядок відповідає окремому кадру, а кожен стовпець положенню конкретного маркера, експорт координат у формати, сумісні з 3D-середовищами, наприклад CVH, JSON або BVH, можливість працювати з різною кількістю маркерів, від 3 до 20, залежно від складності моделі скелета, підтримка реального часу або близької до нього обробки, бажано не менше 10-15 FPS при зчитуванні і записі даних, можливість відображення результатів у

					КС КРБ 123.209.00.00 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

графічному вигляді, наприклад з використанням OpenCV Matplotlib, для попереднього перегляду траєкторії руху, інтеграція з середовищем Blender через скрипти на Python, які дозволяють прив'язати координати маркерів до об'єкта.

Крім базової фіксації координат, бажано також передбачувати такі додаткові можливості, як застосування фільтрації шумів у координатах(за допомогою Kalman-фільтра), визначення відносного положення маркерів для побудови простого скелета, налаштування параметрів детекції (порогова яскравість, розмір маркерів, мінімальна площа контуру) та збереження налаштувань і конфігурацій користувача у окремий файл для повторного використання

Таким чином, функціональні вимоги формують основу всього програмного процесу, забезпечуючи відповідність системи практичному призначенню, її надійність, відтворюваність результатів та адаптивність під різні сценарії використання. Всі перелічені можливості повинні бути реалізовані з урахуванням обмежень ресурсів апаратної платформи Raspberry Pi та відповідати принципам стабільності й простоти в реалізації.

					КС КРБ 123.209.00.00 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

2.1 Розробка і аналіз структури системи

Запропонована система призначена для захоплення рухів людини з використанням маркерного костюма. Основна ідея полягає у виявленні спеціальних маркерів, закріплених на тілі актора, за допомогою відеозображення, обробці отриманих даних і перенесені руху з 2D в 3D-графічне середовище. У системі реалізована схема з одним обчислювальним центром (Raspberry Pi 4), камерою, засобами підсвічування та зберіганням даних у зручному форматі.

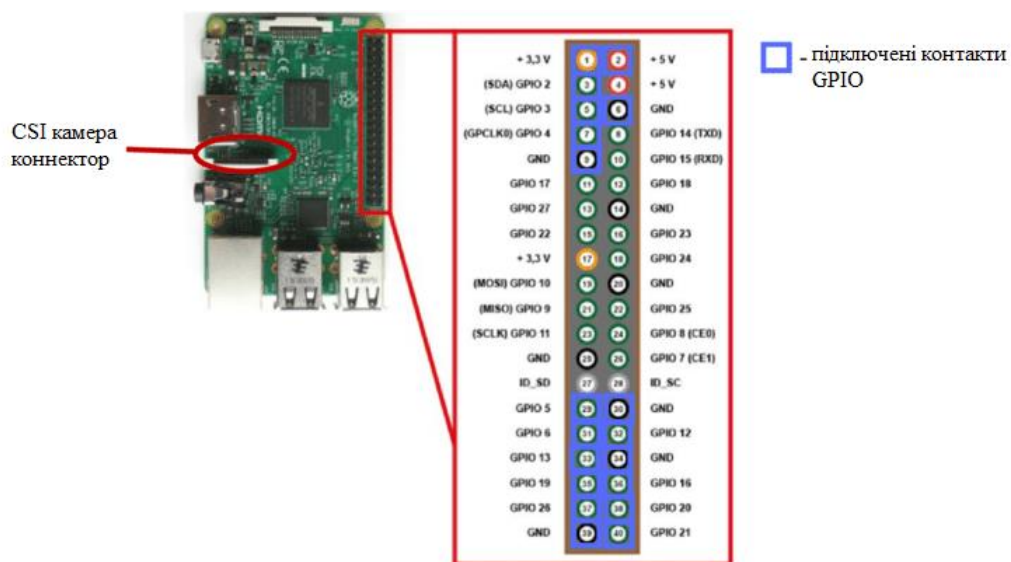


Рисунок 2.1 – Розташування роз'єму для камери (CSI) та контактів GPIO на платі Raspberry Pi 4

На зображенні показано плату Raspberry Pi. Ліворуч виділено конектор CSI — це спеціальний роз'єм, до якого підключається камера. Праворуч показано розташування контактів GPIO (універсальних цифрових входів/виходів), які знаходяться на краю плати.

У схемі GPIO пінів виділено ті контакти, які підключені (обведені синім кольором). Кожен контакт має своє призначення: деякі подають живлення (+3.3V або +5V), деякі — це "земля" (GND), а інші — це керовані порти GPIO, які можуть використовуватись для підключення датчиків, кнопок, світлодіодів тощо.

Таким чином, зображення показує, де на платі знаходиться роз'єм для камери, і які пін-контакти GPIO можна використовувати для підключення додаткових пристроїв.

Структура системи складається з таких компонентів, як Raspberry Pi 4 Model B – центральний елемент, що відповідає за захоплення зображення, обробку кадрів та фіксацію координат маркерів, камера Raspberry Pi Camera Module V2 – це модуль камери, що відповідає за запис відео з частотою від 30 кадрів на секунду в роздільній здатності 720p або 1080p, маркерний костюм, або елементи з ArUco-маркерами, вони забезпечують фіксацію положення ключових точок тіла актора, ІЧ-підсвічування, який дозволяє стабільно розпізнавати маркери, якщо умови освітлення є поганими та зовнішній комп'ютер, який використовується для подальшої обробки отриманих даних у нашому випадку Blender.

2.2 Обґрунтування вибору обладнання

У процесі створення системи захоплення руху актора для подальшого використання у візуальних ефектах у кіно одним із найважливіх етапів є підбір відповідного обладнання. Вибір повинен базуватись не лише на функціональності, але й на сумісності, стабільності, масштабованості та доступності компонентів. На основі аналізу технічного завдання сформовано перелік вимог до обладнання, серед яких забезпечення стабільного відеозахоплення, можливість виявлення маркерів у реальному часі, доступність елементів, підтримка відкритих інтерфейсів та бібліотек, зокрема

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

OpenCV, а також зручність обробки результатів у середовищах, таких як Blender або інші 3D-графічні редактори.

Загальна система складається з таких ключових апаратних блоків, як:

Одноплатний комп'ютер для обробки зображення (обрано Raspberry Pi 4 Model B), Камера, сумісна з обчислювальною платформою Raspberry Pi Camera V2.

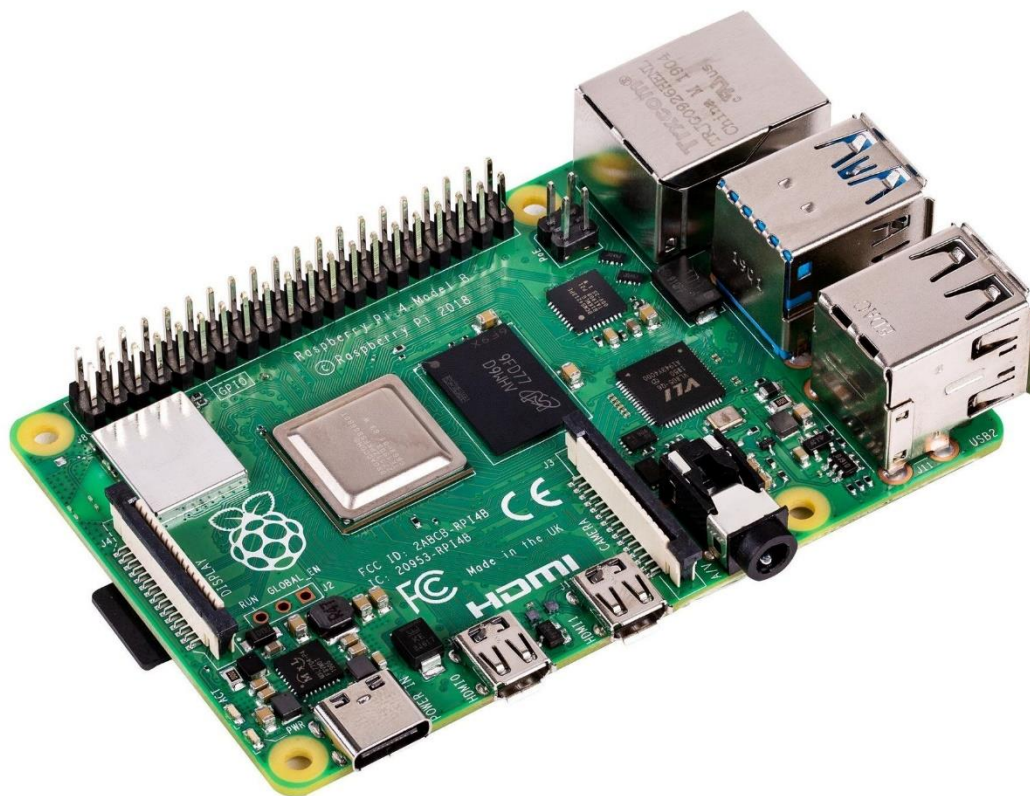


Рисунок 2.2 – Одноплатний комп'ютер Raspberry Pi 4 Model B

Однією із головних причин вибору саме Raspberry Pi є його відкритість, широке ком'юніті, підтримка Python і OpenCV, а також наявність повноцінного ОС з доступом до файлової системи, мережі та мультимедійних бібліотек. Крім того, камера, що підключається через CSI-інтерфейс, дозволяє знімати відео високої якості без затримок і зниження продуктивності.

Вибір камери також має вирішальне значення. Стандартна USB-камера не завжди забезпечує достатню якість або стабільність, особливо при високому навантаженні на процесор, тому було прийнято рішення

використовувати камеру Raspberry Pi Camera Module V2

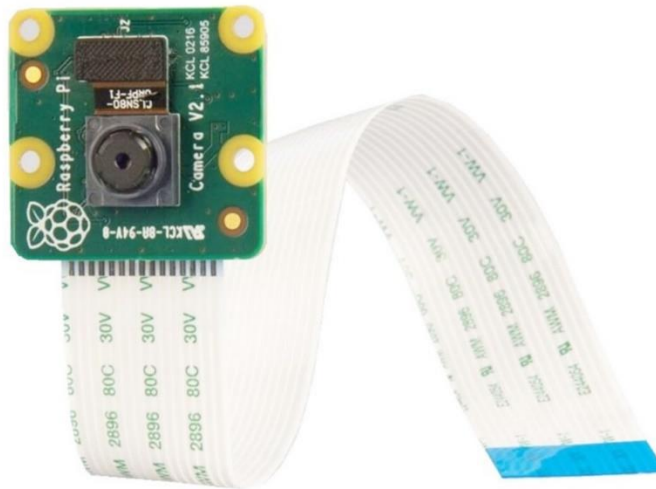


Рисунок 2.3 – Камера Raspberry Pi Camera V2

яка має 8-мегапіксельний сенсор Sony IMX219, що підтримує відео з роздільністю до 1080p і швидкістю до 30 кадрів на секунду. Такий параметр є критичним для коректного виявлення маркерів, адже навіть незначна затримка або розмиття зображення ускладнює комп'ютерний аналіз.

ArUco-маркери — це спеціальні двовимірні графічні мітки, які створені для надійного виявлення і розпізнавання камерами комп'ютерного зору. Вони використовуються в системах позиціонування, трекінгу об'єктів, робототехніці та, зокрема, при захопленні руху. Маркери мають вигляд чорно-білих квадратів, що містять унікальні бінарні коди для ідентифікації. Кожен маркер обрамлений чорним контуром, який спрощує його виявлення на зображенні, а внутрішній вміст — це бінарний шаблон, що дозволяє визначити номер маркера.

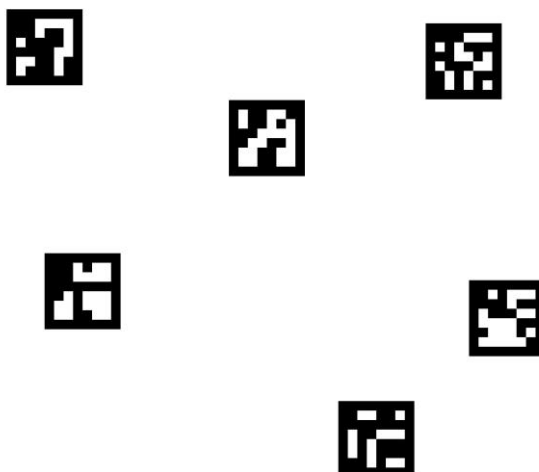


Рисунок 2.4 – Вигляд ArUco-маркерів

Основна ідея роботи ArUco-маркера базується на принципах комп'ютерного зору. Камера фіксує зображення сцени, де розташовані маркери, після чого алгоритм обробки зображення виконує кілька ключових етапів. На першому етапі здійснюється детекція контурів на зображенні з подальшою фільтрацією за формою — система шукає чіткі чорні квадрати. Далі ці квадрати аналізуються для виявлення внутрішнього шаблону, що визначає ідентифікатор кожного маркера.

Маркерні елементи (наприклад, ArUco-маркери), розташовані на тілі актора, допоміжні компоненти, такі як ІЧ-підсвічування, кріплення, елементи живлення та передача даних, та периферійний ПК або ноутбук для постобробки.

Для точного визначення положення в просторі використовується не менше 8-12 маркерів, які розміщуються на основних суглобах – лікті, зап'ястя, плечі, стегна, коліна тощо. При правильному розміщенні і високій частоті кадрів така система дозволяє реконструювати повний рух актора у 2D, якщо додати додаткові камери, то можна і в 3D-моделі.

Підсвічування сцени реалізовано за допомогою ІЧ-світлодіодів, що

					КС КРБ 123.209.00.00 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

живляться від GPIO Raspberry Pi або зовнішнього джерела. Це дає змогу уникнути проблем з тінями або нестабільним освітленням. Також застосовано оптичні фільтри для покращення видимості маркерів у середовищах з паразитними світловими джерелами.

Передача даних здійснюється через Wi-Fi або Ethernet, залежно від умов експерименту. Якщо використовується автономна система без монітора, з'єднання з Raspberry Pi підтримується через SSH, а збережені CSV-файли переносяться або автоматично, або вручну через флеш-накопичувач.

Для постобробки результатів передбачається використання Blender, у якому за допомогою Python-скриптів імпортуються дані з CSV, створюється Armature-структура та здійснюється автоматизоване розміщення ключових кадрів відповідно до положення маркерів у часі. Це дозволяє повністю відтворити рух актора у вигляді 3D-анімації.

Таким чином, вибір усіх компонентів системи обґрунтований з точки зору їхньої функціональності, сумістності, ефективності та практичного застосування. Обрані пристрої та технології дозволяють реалізувати компактну й функціональну систему захоплення руху, адаптовану для використання як у навчальних, так і у напівпрофесійних цілях.

2.2.1 Вибір мікроконтролера

Вибір обчислювального ядра системи відіграє ключову роль у проєктуванні всієї архітектури. Саме мікроконтролер або одноплатний комп'ютер забезпечує обробку відеопотоку з камери, виконання алгоритмів виявлення маркерів, обробку координат, їх запис, зберігання, передавання, а в деяких випадках навіть первинну візуалізацію.

Для побудови системи захоплення руху було обрано Raspberry Pi 4 – це одноплатний комп'ютер, який поєднує у собі малі габарити, широку підтримку бібліотек (OpenCV, NumPy, SciPy, PySerial тощо) та потужність, достатню для обробки зображень у реальному часі.

					КС КРБ 123.209.00.00 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Raspberry Pi 4 має достатню обчислювальну потужність, щоби реалізовувати детекцію маркерів у кожному кадрі відео в реальному часі без істотної затримки. Це критично для точного захоплення актора, особливо при складній або швидкій динаміці рухів.

Технічні характеристики Raspberry Pi 4 можна переглянути у таблиці 2.5.

Таблиця 2.5 – Характеристика Raspberry Pi 4

Процесор	4-ядерний ARM Cortex-A72, 1.5 ГГц
Оперативна пам'ять	Від 2 до 8 ГБ LPDDR4-2400 SDRAM
Графічне ядро	Broadcom VideoCore VI
Інтерфейси	USB 2.0/3.0, GPIO 40-контактний, micro-HDMI, CSI, DSI
Камерний інтерфейс	CSI (Camera Serial Interface)
Метод зберігання	microSD, HDD
Живлення	5В 3А через USB-с

Однією з ключових переваг Raspberry Pi 4 є вбудований CSI-інтерфейс, який дозволяє підключати камери Raspberry Pi без використання USB. На відміну від USB-камер, де зображення проходить через проміжні драйвери та затримки, CSI-камери дозволяють знімати відео з мінімальним лагом і втратою кадрів.

Крім того, використання операційної системи Raspberry Pi OS (на базі Debian Linux) дозволяє працювати у звичному середовищі розробки з підтримкою Python, Shell-команд, встановленні бібліотек через apt, збереження журналів, обробку винятків та ведення логування. Це суттєво полегшує налагодження системи, а також спрощує запуск скриптів на стартапі, автоматизацію запуску захоплення кадрів, очищення тимчасових файлів тощо.

Також варто зазначити доступ до GPIO-контактів, що надає можливість підключати додаткові пристрої, наприклад самі ІЧ-світлодіоди для

підсвічуванні маркерів у темному середовищі, або кнопку старту/зупинки запису. Нижче проведено порівняння з альтернативами у таблиці 2.2.

Таблиця 2.2 – Характеристика платформ

Платформа	Переваги	Недоліки
Raspberry Pi 4	Потужність, CSI, Linux, Wi-Fi, USB	Потребує активного охолодження
ESP32	Енергоефективність	Немає CSI, мало RAM, не тягне відео
STM32	Хороша інтеграція з датчиками	Дуже обмежена пам'ять, складне ПЗ
Jetson Nano	Потужний для AI та відеоаналізу	Енергоспоживання, громіздкість

На основі аналізу видно, що для цілей Raspberry Pi є оптимальним вибором завдяки балансу, продуктивності та підтримкою спільноти. Плата дозволяє розробити систем захоплення, яка не потребує стороннього сервера, легко інтегрується з бібліотеками, має готові приклади реалізації та документацію.

Крім того, це платформа, яку можна легко модифікувати або масштабувати. Наприклад, у майбутньому можливо додати декілька камер або підключити додаткові обчислювальні блоки для створення багатокамерної системи захоплення з триангуляцією координат у 3D-просторі.

Таким чином, вибір Raspberry Pi 4 цілком хорошим варіантом, як для початкового прототипу, так і для подальшого вдосконалення системи захоплення руху в умовах навчальних і дослідницьких проєктів за рахунок своїх переваг, та не дуже сильних недоліків.

2.2.2 Обґрунтування засобів захоплення та об'єкта спостереження

Для ефективного функціонування системи захоплення руху ключовим є вибір відповідного засобу візуального спостереження, що дозволяє з високою точністю фіксувати положення людського тіла у просторі. У даному проєкті такою складовою є камера, яка працює у зв'язці з маркерним костюмом, набором спеціально розміщених міток, що закріплюються на тілі актора.

Основні вимоги до камери, це достатня частота кадрів, висока роздільна здатність, мінімальна затримка передачі зображення та сумісність з обраною обчислювальною платформою (Raspberry Pi 4). Після технічного аналізу було обрано Raspberry Pi Camera, яка базується на сенсорі Sony IMX219. Камера має характеристики, такі, як: Роздільна здатність зображення – до 3280x2464 пікселів, відеозйомка – 1080p при 30 кадрах/с, 720p при 60 кадрах/с., інтерфейс підключення – CSI (Camera Serial Interface), що забезпечує швидкий зв'язок без навантаження USB, підтримка форматів MJPEG, H.264, YUV, що дозволяє зменшити обсяг переданих даних.

Окрім цього, камера має компактні габарити та малу вагу, легко інтегрується в невеликі корпуси або платформи, може бути встановлена на штатив або жорстке кріплення. Завдяки інтеграції з Raspberry Pi камера дозволяє проводити повноцінну обробку зображення у реальному часі з використанням бібліотек OpenCV та NumPy.

Об'єктом, за яким здійснюється спостереження, є людина, актор, який виконує певну фізичну дію (ходьба, поворот корпусу, рух рук тощо). Оскільки звичайна відеозйомка не дає можливості точно обрахувати положення суглобів у просторі, було використано маркерний підхід. Це методика, за якої на тіло актора наклеюються або пришиваються унікальні візуальні мітки, що розпізнаються засобами комп'ютерного зору.

В нашому випадку, застосовується ArUco-маркери, чорно-білі квадрати з унікальними двійковими кодами. Кожен маркер розпізнається не лише за присутністю, а й за ідентифікатором, що дозволяє системі точно знати, до якої

					КС КРБ 123.209.00.00 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

частини тіла відноситься кожен з них.

Основними властивостями ArUco-маркерів те, що вони можуть бути згенеровані автоматично та роздруковані на звичайному принтері, вони мають високу контрастність, що забезпечує стабільне виявлення навіть при змінному освітленні, OpenCV має повну підтримку виявлення та обрахунку позиції маркерів у кадрі, вони вказуються як положення, так і орієнтацію маркера, що дозволяє будувати скелетну модель руху.

Маркерні мітки закріплюються на ключових точках тіла, а саме на суглобах, кінцівках, грудній клітці. У випадках найпростішої реалізації достатньо 10-12 міток, щоб відтворити основні рухи тіла, жести рук, повороти корпусу, ходьби. Також можливо розширити кількість маркерів для покращення точності або деталізації, наприклад для захоплення пальців рук.

Надійнісь захоплення положення маркерів залежить не лише від якості самої камери, та алгоритмів обробки, але й від умов освітлення, тла, розміру міток і куди спостереження. Найкращі результати досягаються при дотриманні певних умов. Освітлення сцени повинно бути рівномірним, бажано з використанням м'яких джерел світла або за допомогою ІЧ-підсвітки, камера розміщується на відстані 1.5-2.5 метра від актора, це залежить від поля зору, мітки повинні бути прикріплені на твердій поверхні одягу без сильних розтягувань чи перекосів, не бажано уникати часткових втрат маркерів з поля зору чи їхнє надмірне перекриття.

Використання ArUco-маркерів у поєднанні з камерою Raspberry Pi Camera дозволяє створити ефективну, бюджетну та точну систему візуального спостереження за актором. Такий підхід не потребує складного обладнання або дорогих ліцензій, працює в реальному часі, масштабований і легко налагоджується навіть у домашніх умовах.

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

2.2.3 Канали комунікації

У системі захоплення руху, яка базується на Raspberry Pi, надзвичайно важливим є вибір і правильна організація каналів передачі даних. Від цього залежить як стабільність роботи системи, так і ефективність подальшої обробки анімаційних даних. Комунікація між окремими компонентами системи камерою, мікроконтролером, зовнішнім ПК, накопичувачами, мережею – визначає зручність розробки, можливість масштабування та підтримку реального часу.

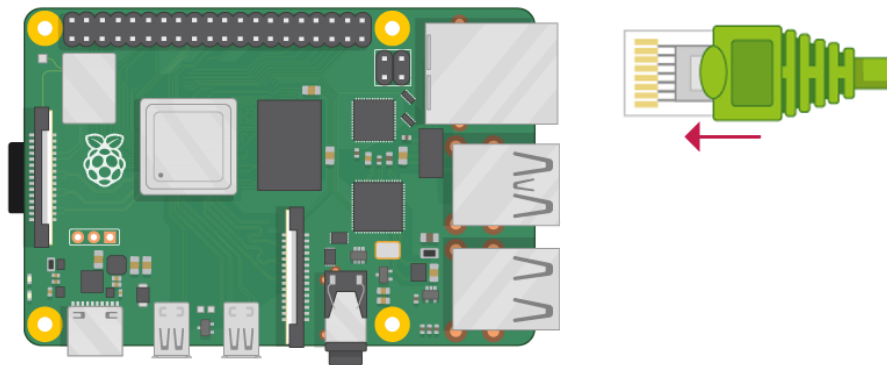


Рисунок 2.5 – Підключення Ethernet до Gigabit слота

Raspberry Pi 4 оснащений двома основними каналами зовнішньої мережевої комунікації: WiFi 802.11ac(2.4/5GHz) це бездротовий інтерфейс, що забезпечує середню швидкість передачі даних до 100Мбіт на секунду в реальних умовах, Gigabit Ethernet це дротовий канал зі швидкістю до 1Гбіт на секунду, однак реальна продуктивність складає близько 300-500 Мбіт/с через обмеження контролера USB.

У процесі тестування система показала стабільну роботу при підключенні по Ethernet, що дозволяє швидко передавати CSV-файли з координатами руху, або навіть передавати відеопотік для попереднього перегляду. Wi-Fi є зручним при портативному використанні, наприклад, коли

камера закріплена на актора і потребує автономності. Також завдяки Wi-Fi забезпечується SSH-підключення до системи, для запуску скриптів, навантаження даних або оновлення програмного забезпечення без використання монітора, миші чи клавіатури. Це значно спрощує обслуговування пристрою під час роботи

На платі Raspberry Pi 4 присутні 4 USB-порти, два з яких стандарту USB 3.0. Це дає можливість підключення зовнішнього SSD або флеш накопичувача для зберігання великих обсягів даних, USB-концентраторів для розширення кількості портів та резервної камери, у разі використання USB-модулів.

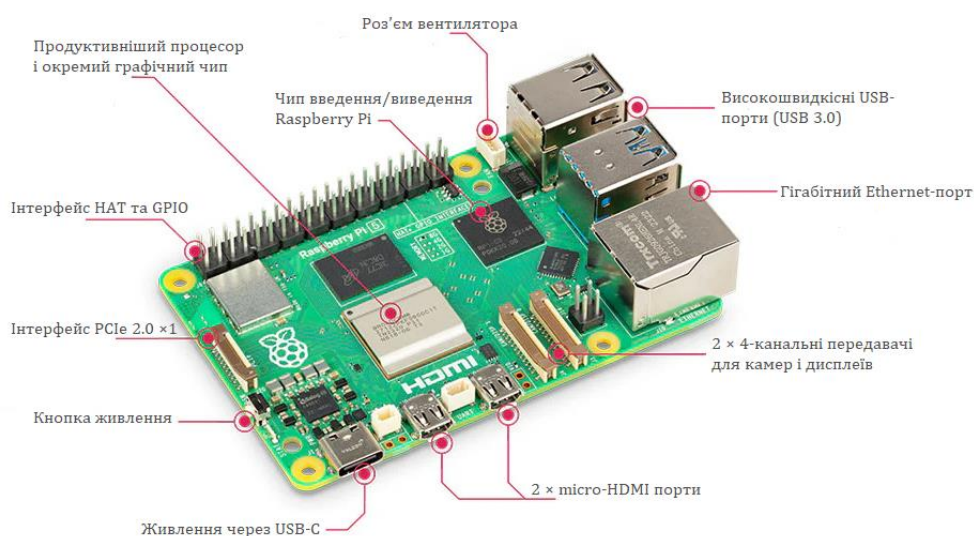


Рисунок 2.6 – Основні апаратні елементи плати Raspberry Pi 4

У системі захоплення руху USB використовується для резервного експорту результатів, коли відсутнє мережеве з'єднання. Наприклад, після завершення сесії запису координат у CSV-файл, файл можна копіювати на флешку або накопичувач і перенести на основний ПК.

Важливим каналом взаємодії з додатковими пристроями є GPIO (General Purpose Input/Output) – 40-контактний роз'єм, який підтримує підключення LED-індикаторів, кнопок керування, інфачервоних джерел світла, та інших

модулів, зокрема: Кнопка “Пуск/стоп” для запуску або завершення запису без доступу консолі, LED-сигналізація для виводу стану системи (помилка, запис активний, очікування). Реле або MOSFET, для керування живленням зовнішніх інфрачервоних світлодіодів, та датчики температури або освітлення для адаптації режиму зйомки.

GPIO-контакти також підтримують інтерфейси I2C і SPI, що дозволяє додатково підключати сенсори IMU (акселерометри, гіроскопи), які можуть бути використані для комбінованих систем захоплення руху

Під час роботи з Raspberry Pi є можливість віддаленого доступу до системи через SSH або VNC. SSH дозволяє керувати пристроєм через командний рядок, у тому ж числі запускати Python-скрипти, а VNC дозволяє працювати з графічним інтерфейсом Raspberry Pi OS. Це дозволяє запускати візуальні середовища, переглядати відео з камери у реальному часі, налаштовувати скрипти обробки без прямого підключення до монітора.

Дані, отримані після обробки зображення з камери, зберігаються у вигляді CSV-файлів з координатами кожного маркера в часі, наприклад ID маркера, X, Y, Z TimeStamp, ці файли можуть бути передані через Wi-Fi/Ethernet на основний ПК, записані на USB-накопичувач, тимчасово збережені у внутрішній пам’яті карти microSD. Збереження поводить ся з урахуванням структурованого формату, придатного для імпорту у Blender або інші 3D-редактори за допомогою скриптів. Перед передачею дані можуть бути автоматично заархівовані або попередньо фільтровані.

За потреби можлива реалізація захищених каналів з’єднання, наприклад, використання протоколу SCP або SFTP для передачі файлів, або запуск скриптів через SSH з авторизацією по ключ. Це дозволяє забезпечити обробку даних при роботі в мережевому середовищі.

Отже, система передбачає використання всіх доступних комунікаційних інтерфейсів Raspberry Pi для забезпечення надійності, гнучкої і зручної передачі даних, керування записом та інтеграції з периферійним обладнанням.

					КС КРБ 123.209.00.00 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

Навність кількох каналів одночасно (Wi-Fi, Ethernet, USB, GPIO) дозволяє адаптувати систему до різних умов експлуатації, від стаціонарного використання до мобільних записів.

2.3 Розробка і аналіз електричної схеми

У цьому підпункті представлено електричну архітектуру комп’ютеризованої системи захоплення руху актора на базі одноплатного комп’ютера Raspberry Pi 4. Вона включає усі основні функціональні вузли, тобто це камера, маркерні засоби з інфрачервоним підсвічуванням, джерело живлення, індикатори, кнопку керування та порти для обміну даними, детально проаналізовано спосіб підключення основних компонентів та розглянуто енергетичну і сигналізаційну частину пристрою.

2.3.1 Структура загальної електричної схеми

Загальна електрична схема побудована на основі центральної плати Raspberry Pi 4, яка виконує роль обчислювального та керуючого елемента. До неї підключаються всі інші модулі через стандартні інтерфейси (GPIO, CSI, USB, I2C) На рисунку 2.7 наведено загальний вигляд функціональності схеми взаємодії елементів системи:

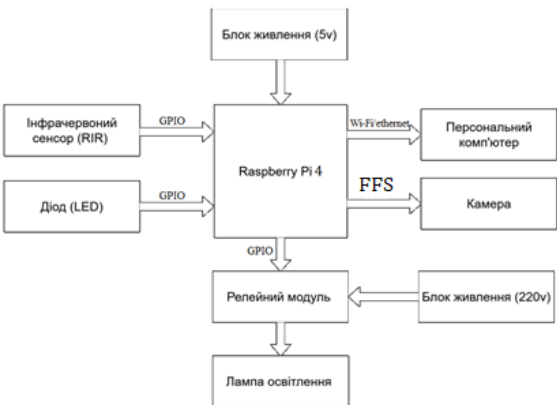


Рисунок 2.7 – Загальна структурна схема комп’ютеризованої системи захоплення руху актора на базі Raspberry Pi 4

Однією з основних переваг CSI-підключення є те, що дані зображення надходять напряму в графічний процесор Raspberry Pi, що дозволяє зменшити затримку під час зчитування кадрів, а також забезпечити апаратне прискорення під час обробки відео у реальному часі. Це є критично важливим для задач захоплення руху, де потрібна точність синхронізації. Монтаж камери здійснюється без пайки, що значно спрощує збірку пристрою. Камеру можна закріпити на корпусі системи або встановити на штатив у потрібній позиції для зручного захоплення сцени з актором. Важливо дотримуватись обмежень щодо довжини шлейфу, якщо точніше то не рекомендується перевищувати 25 см без використання буферних підсилювачів сигналу. Камера живиться безпосередньо через CSI-інтерфейс. Середнє енергоспоживання становить приблизно 250-300 мВт, що не створює надмірного навантаження на загальне живлення пристрою. У випадку використання модифікованих камер з підсвічуванням або кількох модулів одночасно, необхідно передбачити додаткове керування подачею живлення через реле або транзисторні ключі, підключені до GPIO.

2.3.3 Схема живлення системи

Стабільне живлення є дуже важливим фактором для надійності роботи комп'ютеризованої системи захоплення руху, зокрема, коли задіяні одноплатні комп'ютери, камери, інфрачервоні підсвітки та інші елементи периферії. У даній розробці використовується Raspberry Pi 4, який вимагає джерело живлення 5V із силою струму не менше 3A. Живлення подається на Raspberry Pi 4 через порт USB-C. Це рекомендований стандарт виробника, оскільки він підтримує більші навантаження, ніж microUSB, і виключає ризики втрати стабільності під час пікових навантажень, наприклад, при одночасному записі відео з камери та збереженні даних на USB-накопичувач.

Для забезпечення надійної роботи плати, рекомендується використання фірмового блоку живлення потужністю 15Вт (5V, 3A) або еквівалентного

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

імпульсного адаптера з подачею стабілізованої напруги. Обов'язковою є наявність захисту від перенапруги, короткого замикання та просідання струму при старті камери. Зовнішні джерела живлення повинні відповідати стандартам безпеки, маючи сертифікацію CE або UL. Крім живлення самої плати, потрібно враховувати, що підключені також камера (~250-300 мВт), ІЧ-підсвітка (LED-модулі можуть споживати 300-800 мА), керуючі елементи (кнопки, реле, індикатори), та USB-накопичувачі (можуть споживати 500-700мА). З огляду на це, реальна потреба у струмі може перевищувати 2.5-3.0А. При використанні слабших адаптерів можливі збої, зависання, втрати відео або некоректна робота камери. Приклад одного із такого блоку живлення, є офіційний блок для Raspberry Pi 4.



Рисунок 2.9 – Офіційний блок живлення для Raspberry Pi 4

Плати Raspberry Pi 4 дозволяють подавати живлення безпосередньо через контакти GPIO: Пів 2 бо 4 – 5V (без захисту). Пін 6, 9, 14 і тд. – GND.

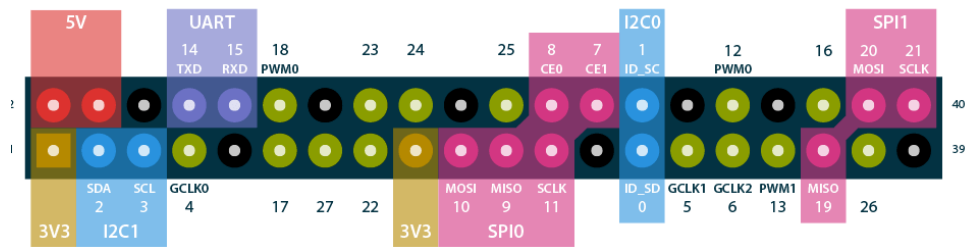


Рисунок 2.10 – Pinout-схема GPIO-виходів Raspberry Pi 4 з позначенням живлення, шини UART, SPI, I2C.

Ці виходи дозволяють підключати периферійні пристрої напряму до живлення, наприклад LED-індикатори, реле керування підсвіткою, датчики або мікромодулі Bluetooth. Варто враховувати, що ці піни не мають вбудованого захисту, тому підключення потужних модулів через GPIO без реле або транзисторного буфера не рекомендується.

Щоб запобігти спотворенню сигналу при підключенні інфрачервоних модулів або підсвічування, доцільно використовувати окремий контур живлення. Найпростіший варіант, це підключити ІЧ-підсвітку до зовнішнього блоку 5V через MOSFET, який керується GPIO. Це дозволяє уникнути перевантаження основного БЖ, автоматично вмикати чи вимикати підсвітку в умовах низької освітленості та зберігати стабільну роботу інших компонентів.

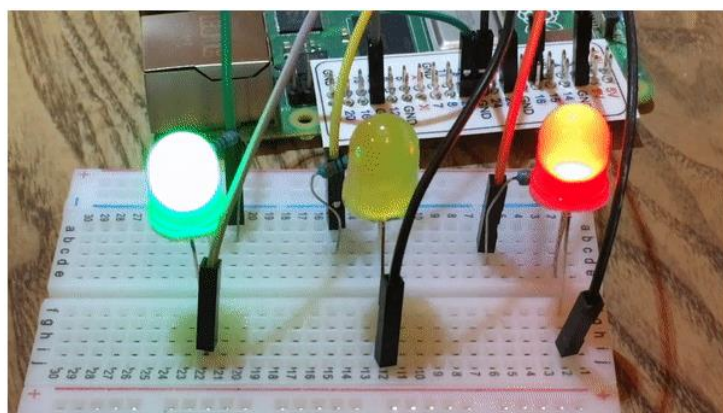


Рисунок 2.11 – підключення світлодіодів через MOSFET до GPIO

У схемі живлення рекомендовано передбачити елементи захисту, таких як: TVS-діод (5.0V) для захисту від перенапруг, полімерний запобіжник (3A/5V), конденсатори на 100нФ і 470мкФ для фільтрації шумів і згладжування пікових навантажень, дросель на 100-150μН у випадку сильних імпульсних коливань. Ці елементи встановлюють на вході USB-C або на лінії GPIO 5V для забезпечення надійності живлення в умовах перешкод, перепадів мережі або нестабільного середовища, наприклад студійна техніка, часте включання/вимикання

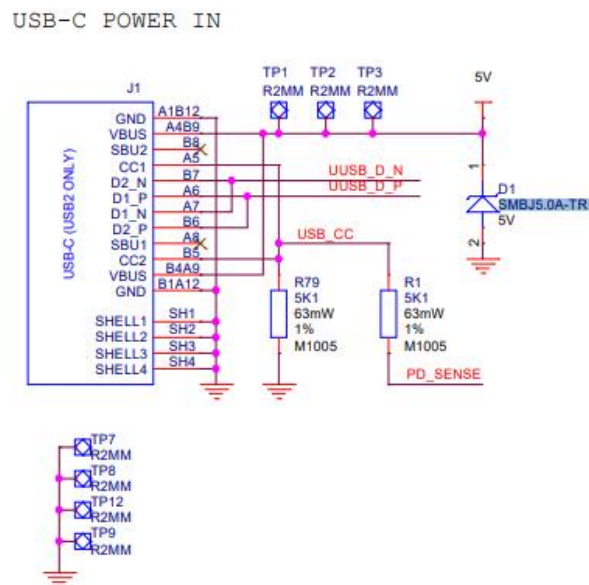


Рисунок 2.12 – Схема захисту живлення через USB-C в Raspberry Pi 4 з TVS-діодом SMBJ5.0A

У підсумку, схема живлення системи передбачає всі необхідні технічні умови для стабільної роботи всіх компонентів. Вона побудована на основі стандартного адаптера живлення, з можливістю подальшого розширення. За рахунок використання окремих контурів і засобів захисту забезпечується як надійність системи, так і безпечна експлуатація.

2.4 Вибір необхідного програмного забезпечення для реалізації проєкту

Підбирання програмного забезпечення відіграє ключову роль у побудові надійної системи захоплення руху. Воно забезпечує керування апаратною частиною, зчитування й обробку відеоданих, їх трансформацію у формат, зручний для подальшого використання у 3D-моделюванні та створенні візуальних ефектів. Нижче розглянуто повний стек необхідного програмного забезпечення для реалізації поставленого завдання

2.4.1 Операційна система

Базовою платформною є Raspberry Pi OS (Lite-версія), що оптимізована для роботи на Raspberry Pi 4. Вона заснована на Debian і дозволяє повністю контролювати систему через термінал. Lite-версія займає менше пам'яті, швидше завантажується, а також є ідеальним середовищем для вбудованих систем. Переваги системи заключається в повній підтримці ARM-архітектури, регулярні оновлення та патчі безпеки, стабільна робота з камерами через CSI-інтерфейс, сумісність з Python, GCC, та бібліотеками для комп'ютерного зору. Команди адміністрування, встановлення драйверів, оновлення системи виконуються через SSH, що дозволяє працювати з віддалених пристроїв без потреби підключення монітора.

2.4.2 Програмні бібліотеки для камери

Для камери використовується бібліотека libcamera, яка є офіційним інструментом для роботи з CSI-камерами на Raspberry Pi. Це сучасний стек драйверів, що підтримує апаратну обробку зображень, автоекспозицію, баланс білого та інші корисні функції. Основними інструментами є libcamera-still – для фіксації окремих кадрів, libcamera-vid – запис відео у форматі .h264, libcamera-raw – отримання сирих даних матриці камери (RAW10/RAW12).

Команди необхідні для запуску відеозапису

					КС КРБ 123.209.00.00 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

```
"Libcamera-vid -t 0 -width 1280 -height 720 -o output.h264":
```

- "Libcamera-vid" запускає програму для запису відео з камери Raspberry Pi
- "--t 0" Тривалість запису 0 = нескінченно, до поки не буде зупинено вручну
- "--width 1280" встановлює ширину відео в пікселях (в цьому випадку ми задаєм 1280 px)
- "--height 720" встановлює висоту відео в пікселях (720 px)
- "-o output.h264" Вивід у файл output.264 у форматі H.264 (стиснене відео)

У результаті ми отримаємо постійний відеозапис у роздільній здатності 1280x720, файл output.h264 зберігається у поточній директорії, формат H.264 зручний для подальшої обробки або конвертації. Усі бібліотеки підтримують збереження зображень у реальному часі, що важливо для трекінгу. Крім того, libcamera дозволяє взаємодіяти з камерою з Python через бібліотеку picamera2.

2.4.3 Мова програмування та бібліотеки для обробки

Python був вибраний як основна мова завдяки простоті синтаксису та швидкості розробки, великій кількості готових бібліотек для комп'ютерного зору, підтримці Raspberry Pi OS без необхідності додаткового встановлення.

```
import cv2
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if ret:
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        cv2.imshow("Camera", gray)
        if cv2.waitKey(1) == ord('q'):
            break
cap.release()
cv2.destroyAllWindows()
```

Рисунок 2.13 – Програма захоплення відео з камери та відображення зображення у відтінках сірого за допомогою OpenCV

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

Бібліотеки, які використовуються: `opencv-python` обробляє відео в реальному часі, робота з фільтрами, розпізнавання маркерів, `RPi.GPIO` керує реле, ІЧ-підсвіткою, запуском скриптів через кнопки, `pymru` для роботи з числовими масивами, `time`, `os`, `subprocess`, це допоміжні модулі для таймерів, керування файлами та запуску зовнішніх команд. Цей скрипт написано мовою Python з використанням бібліотеки `OpenCV (cv2)`, це одна із найпопулярніших для обробки зображень і відео. Принцип роботи цього скрипта в тому, що він підключається до вебкамери (індекс 0) безперервно зчитує відеопотік кадр за кадром, перетворює кожен кадр у відтінки сірого, відображає його у вікні з назвою “Camera” та зупиняє роботу, якщо натиснути клавішу `q`. Python також використовується у Blender через API для автоматизованого імпорту треків, анімацій, або створення скриптів експорту FBX

2.4.4 Blender та постобробка

Для кінцевої обробки захопленого матеріалу використовується Blender, це програма для 3D-моделювання з підтримкою анімації та відеопостобробки.

Основними модулями виступають Motion Tracking, який дозволяє імпортувати відео та вручну або автоматично прикріпити треки до точок. Draph editor дозволяє редагувати криві анімації. Python API автоматизує імпорт FBX та розкладку треків на кістки. Render Engine (Cycles/Eevee) для створення фінального візуального ефекту.

2.4.5 FFmpeg та конвертація

FFmpeg це обов’язковий інструмент для роботи з відеофайлами, а саме обрізка, кадрування, злиття відео, витягувань зображень. Він, наприклад, дозволяє конвертувати файл `.h264` на `mp4`.

Вибір програмного забезпечення охоплює повний цикл системи, від захоплення руху через Raspberry Pi і Python до обробки та візуалізації в Blender. ці компоненти є відкритими, гнучкими та сумісними, що дозволяє розширити систему, інтегрувати з іншими проектами та підтримувати її в

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

майбутньому. Цей підхід забезпечує не тільки ефективність, а й економічну доцільність реалізації комп'ютеризованої системи трекінгу руху.

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1 Опис алгоритму роботи Raspberry Pi

Основною всієї системи захоплення руху є плата Raspberry Pi 4, яка виконує функції центрального обчислювального модуля. Саме на неї покладається завдання управління всіма підключеними компонентами, камерою, джерелом живлення, ІЧ-індикаторами, кнопками, реле тощо.

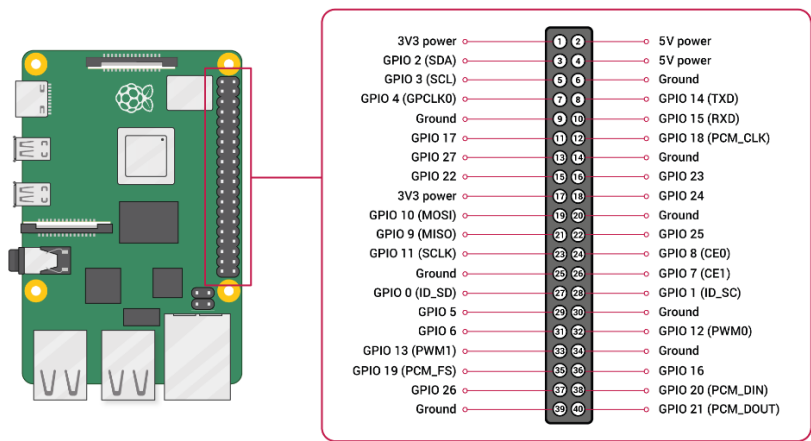


Рисунок 3.1 Схема розпіновки контактів GPIO плати Raspberry Pi 4

На рисунку показано плату Raspberry Pi 4 та схему розпінування її 40-контактного GPIO-роз'єму. Цей роз'єм розташований праворуч і складається з двох рядів по 20 контактів, які призначені для підключення зовнішніх пристроїв, модулів і датчиків. Частина пінів подає живлення — є контакти з постійною напругою 5V та 3.3V, а також декілька контактів «землі» (GND), які забезпечують електричне замикання в схемах. Інші піни позначені як GPIO (General Purpose Input/Output), наприклад, GPIO 2, GPIO 3, GPIO 4 тощо, і можуть програмуватися для виконання різних функцій — як входи або виходи. Деякі GPIO мають додаткові функції: GPIO 2 і GPIO 3 працюють як SDA і SCL для інтерфейсу I²C, GPIO 14 і 15 відповідають за TXD і RXD

для UART (послідовного порту), а GPIO 10, 9, 11 і 8 використовуються для SPI-з'єднання (MOSI, MISO, SCLK, CE). Також доступні піни для цифрової передачі аудіо (PCM). Уся ця система дозволяє розширювати можливості Raspberry Pi, створюючи інтерактивні пристрої, автоматизовані системи, робототехніку, трекінг руху, управління світлом, сенсорами та інше.

У системі використовується Raspberry Pi Os Lite, це полегшена версія офіційної операційної системи, що базується на Debian. Вона характеризується швидким завантаженням, низьким споживанням оперативної пам'яті та повною сумісністю з командним рядком, що робить її зручною для вбудованих систем без графічного інтерфейсу. Упродовж першої хвилини система ініціалізує всі фонові служби, драйвери та засоби роботи з периферією. Далі через механізм автоматичного запуску, наприклад, використовуючи system (/etc/systemd/system/motion.service) або файл /etc/rc.local відбувається старт головного Python-скрипта. Цей скрипт відповідає за увесь логічний ланцюг: зчитування даних з камери, керування GPIO-портами, запуск та моніторинг процесів захоплення, обробку помилок і підготовку даних для збереження або подальшої обробки. На цьому етапі система також перевіряє, чи доступний модуль камери через інтерфейс CSI. Якщо пристрій не виявлено або він не проходить ініціалізацію через libcamera, то система або не починає подальших дій, або сигналізує про несправність за допомогою візуальної індикації на GPIO (наприклад, мигання червоного світлодіода).

Другий етап заключається в активації периферійних модулів. Raspberry Pi активує GPIO-піни для живлення та керування підсвічуванням, сенсорами та індикацією. Наприклад, GPIO 17 (запуск інфрачервоного діода), GPIO 18 (індикація активності через LED), GPIO 23/24 (контроль кнопок користувача для запуску чи зупинки запису). Усі ці порти ініціалізуються на початку основного скрипта. У разі натискання кнопки запуску (наприклад, GPIO 23), система переходить до наступного етапу, а саме до захоплення відео.

					КС КРБ 123.209.00.00 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Третій етап захоплення відео з камери

У випадку успішної ініціалізації камери, що здійснюється через інтерфейс CSI за допомогою бібліотеки `libcamera`, система переходить до безпосереднього етапу зчитування відеопотоку. На цьому етапі Raspberry Pi виконує запуск процесу захоплення відео шляхом ініціації відповідної команди:

```
"libcamera-vid -t 0 --width 1280 --height 720 -o /home/pi/output.h264"
```

Це дозволяє безперервно зчитувати відеопотік у форматі H.264. Процес захоплення відео може бути або безперервним (до вимкнення), або обмеженим за часом (через параметр `-t 10000` – 10 секунд).

Для інтеграції захоплення в Python можна використати бібліотеку `picamera2`, яка надає API для захоплення кадрів, потоків, встановлення параметрів, експозиції, ISO, тощо.

Четвертий етап обробка та зберігання відео. У ході запису відео або після його завершення Raspberry автоматично передає або копіює файл на локальний або мережевий носій. Для цього використовується автоматичне переміщення у вказану директорію (`/mnt/data/records`), передача по мережі через SCP або FTP, локальна обробка Python-скриптами, фреймтування, обрізка, конвертація.

Приклад конвертації відео в .mp4:

```
"ffmpeg -i output.h264 -c:v copy output.mp4".
```

Також можна здійснювати фреймування:

```
"ffmpeg -i output.h264 -vf fps=10 frames/img_%04d.png"
```

П'ятий етап: виявлення маркерів (за допомогою OpenCV). Наступним кроком є попередній аналіз відеоданих з використанням Python-бібліотеки

					КС КРБ 123.209.00.00 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

OpenCV. На цьому етапі система здійснює обробку кожного відеокадру з метою виявлення маркерів, зазвичай світловідбиваючих точок, розміщених на тілі актора. Кожен кадр спершу переводиться у відтінки сірого для зменшення обчислювального навантаження, після чого виконується порогова бінаризація зображення для відокремлення яскравих елементів (маркерів) від фону. Після цього здійснюється пошук контурів які відповідають світлим об'єктам у зображенні, і для кожного маркера обчислюються координати його прямокутного обмеження. Отримані координати зберігаються у масив або можуть експортовані у файл формату CSV для подальшої обробки або візуалізації. Такий підхід дозволяє здійснювати ефективний трекінг рухів актора та підготовку даних до імпорту у середовище тривимірної анімації, наприклад Blender. Код для виявлення маркерів на костюмі актора написаний і проведений у рисунку 3.2.

```
import cv2
cap = cv2.VideoCapture('output.mp4')
while True:
    ret, frame = cap.read()
    if not ret:
        break
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    _, thresh = cv2.threshold(gray, 200, 255,
cv2.THRESH_BINARY)
    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    for cnt in contours:
        if cv2.contourArea(cnt) > 5:
            x, y, w, h = cv2.boundingRect(cnt)
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0,
255, 0), 2)
    cv2.imshow("Detected Markers", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Рисунок 3.2 – Код для виявлення маркерів на костюмі актора

Етап шостий: Взаємодія з Blender. Отримані дані можуть бути

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

експортовані в .FBX або як набір треків координат. Raspberry Pi самостійно не може запускати Blender, але підготовлені дані передаються на основну машину, де через Python API blender виконується скрипт обробки. Після цього рухи можуть бути накладені на 3D-модель а система вважається повністю пройдена по всьому циклу: Зйомка, обробка, перенесення у 3D-простір.

Етап сьомий: Завершення роботи. Після завершення роботи скрипта здійснюється збереження журналу (log.txt), де записуються події: старт, завершення запису, тривалість, кількість кадрів, час збереження, можливі помилки. Після цього всі ресурси вивільнюються, GPIO переходить у безпечний стан, камера вимикається, і пристрій переходить у режим очікування або відключення живлення.

Таким чином, повний алгоритм роботи Raspberry Pi охоплює всі етапи, а саме запуск, ініціалізацію камери, керування периферією, захоплення й обробка відео, базове розпізнавання об'єктів, а також підготовку матеріалу для подальшого використання в Blender. Така структурована система забезпечує надійність, простоту повторного використання та ефективну інтеграцію в технологічний процес створення візуальних ефектів

3.2 Розробка програмного коду

Робота над програмною частиною здійснювалась за допомогою мови Python — високорівневої мови програмування, що є оптимальною для розробки прототипів і керування периферійними пристроями на базі Raspberry Pi. Було використано кілька ключових бібліотек: OpenCV для обробки зображень, RPi.GPIO для роботи з портами вводу/виводу, os, time, subprocess для системних команд, а також picamera2 або libcamera для керування камерою.

На початку розробки було створено головний скрипт, що виконується автоматично після запуску пристрою. Його структура включає кілька логічних

					КС КРБ 123.209.00.00 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

блоків: Імпорт необхідних бібліотек, ініціалізація GPIO-портів та перевірка доступності камери, очікування настикування кнопки запуску системи, запис відео у файл з використаннями libcamera-vid або rpicamera2, обробка відео, детекція маркерів, виділення контурів, побудова координат, та збереження координат у CSV або передачу в зовнішній обробник

Для реалізації процесу запису відео з камери було використано базову команду запуску libcamera-vid, інтегровану у Python-скрипт через виклик системного процесу. Такий підхід дозволяє ініціювати захоплення відео в автоматичному режимі без участі користувача після запуску системи. Параметри командної строки задають тривалість запису (-t 10000 — 10 секунд), роздільну здатність кадру (1280x720) та назву вихідного відеофайлу (output.h264). Команда виконується за допомогою модуля os, що дозволяє взаємодіяти з операційною системою Raspberry Pi.

Перед запуском системи на дисплей виводиться повідомлення про старт процесу, що також можна дублювати через LED-індикатор. Після завершення захоплення даних виводиться повідомлення про успішне збереження файлу, що може бути корисним при налагодженні або для підтвердження коректної роботи.

Фрагмент, який є частиною повного програмного модуля, який інтегрується у стартовий сценарій Raspberry Pi можна переглянути у рисунку 3.3.

```
import os
import time

print("Запуск системи захоплення")
cmd = "libcamera-vid -t 10000 --width 1280 --height 720 -o
output.h264"
os.system(cmd)
print("Запис завершено")
```

Рисунок 3.3 – Код запуску процесу запису відео з камери Raspberry Pi у автоматичному режимі

Після завершення процесу захоплення відео та збереження його у файл,

					КС КРБ 123.209.00.00 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

наступним кроком є аналіз кожного окремого кадру з метою виявлення маркерів. Для цього у структурі програмного забезпечення був створений окремий модуль з назвою `analyze.py`. У цьому модулі реалізовано логіку посткадрової обробки, яка передбачає завантаження відеофайлу, зчитування кожного кадру, його обробку та аналіз світлових об'єктів. Основним інструментом виступає бібліотека `OpenCV`, яка дозволяє ефективно здійснювати низку операцій над зображенням: переведення в чорно-білий формат, застосування порогової фільтрації, пошук контурів та побудову обмежувальних прямокутників. Завдяки таким алгоритмам можливо локалізувати кожен маркер, визначити його положення в координатній системі зображення, а також підготувати масив із координатами, придатний для подальшої передачі в середовище тривимірної анімації. Уся ця обробка виконується в автоматичному режимі та може масштабуватись для обробки кількох відео одночасно або в реальному часі. Код можна переглянути нижче в рисунку 3.4.

```
import cv2
import csv
cap = cv2.VideoCapture('output.h264')
with open('markers.csv', 'w', newline='') as f:
    writer = csv.writer(f)
    while True:
        ret, frame = cap.read()
        if not ret:
            break
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        _, thresh = cv2.threshold(gray, 200, 255,
cv2.THRESH_BINARY)
        contours, _ = cv2.findContours(thresh,
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        for cnt in contours:
            if cv2.contourArea(cnt) > 5:
                x, y, w, h = cv2.boundingRect(cnt)
                writer.writerow([x + w//2, y + h//2])
cap.release()
```

Рисунок 3.4 – Код для аналізу відео та побудови координат

Цей скрипт реалізує зчитування кожного кадру з відеофайлу,

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

перетворення зображення у відтінки сірого, виділення найяскравіших ділянок, ідентифікацію контурів об'єктів (якими, з високою ймовірністю, є світловідбиваючі маркери), а також побудову прямокутників навколо знайдених об'єктів. Центр кожного з таких прямокутників обчислюється та записується у форматі X-Y координат до CSV-файлу. Це дозволяє отримати серію координатних значень, які у подальшому можуть бути використані для побудови траєкторії руху тіла, а також для прив'язки цих точок до відповідної скелетної структури 3D-моделі. Такий підхід є особливо зручним при імпорті даних у програмне середовище Blender або інші засоби трекінгу, що підтримують автоматизований аналіз рухів актора за координатними точками.

Додатково були створені модулі для роботи з кнопками та LED-індикацією, наприклад запуск запису лише після натискання кнопки, код представлений у рисунку 3.5.

```
import RPi.GPIO as GPIO
BUTTON_PIN = 23
GPIO.setmode(GPIO.BCM)
GPIO.setup(BUTTON_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)
print("Очікування натискання кнопки...")
while GPIO.input(BUTTON_PIN):
    time.sleep(0.1)
print("Кнопку натиснуто, початок запису")
```

Рисунок 3.5 – Код для початку запису при нажиманні кнопки

Весь код протестований та адаптований до ресурсів Raspberry Pi 4. Його структура модульна, що дозволяє легко змінювати параметри, додавати нові функції або інтегрувати у більші проекти. Робота програмної частини безпосередньо пов'язана з якісним виконанням усіх етапів захоплення, обробки та передачі анімаційної інформації у візуалізаційне середовище

3.3 Тестування та принцип роботи

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

Після завершення розробки апаратної та програмної частини була проведена серія тестувань, які дозволили оцінити працездатність системи в умовах наближених до реальних. Основна мета тестування — перевірити здатність системи коректно захоплювати рух актора, фіксувати маркери, передавати й обробляти дані, а також визначити стабільність і надійність роботи як програмного коду, так і електронних компонентів.

Тестування здійснювалось у спеціально підготовленому середовищі з використанням Raspberry Pi 4, камери Raspberry Pi Camera Module 3, а також набору маркерів, закріплених на тілі учасника експерименту. Для цього було використано костюм із кріпленням світловідбиваючих маркерів (білих або сріблястих точок), які добре виявляються у кадрі навіть при слабкому освітленні. У приміщенні було встановлено додаткове інфрачервоне підсвічування, яке не заважає відеозапису, але покращує виявлення маркерів у чорно-білих кадрах.

Перед запуском системи проводилась перевірка працездатності всіх елементів: тестувалась камера, коректність запуску основного Python-скрипта, робота GPIO, індикація світлодіодами, а також відповідь на натискання кнопки запуску. У разі успішної перевірки ініціювався процес захоплення руху.

У процесі експерименту учасник виконував декілька типових рухів: хода вперед, розвороти, махи руками, нахили та присідання. Запис кожної сесії тривав близько 10–15 секунд. Після завершення зйомки відеофайли аналізувались за допомогою створеного програмного забезпечення, що виконувало виділення маркерів, збереження координат у CSV-файл та побудову попередньої візуалізації в режимі реального часу (вікно з обведеними прямокутниками).

Після завершення аналізу кадрів отримані дані були імпортовані у Blender, де здійснювалась візуальна перевірка точності координат. За допомогою Python API було побудовано систему кісток (Armature), до якої

					КС КРБ 123.209.00.00 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

прив'язувались координати з CSV. Рухи відображались у режимі екшену на моделі та зіставлялись із реальними рухами актора, зафіксованими на відео.

В результаті проведеного тестування були встановлені такі характеристики системи:

- Час реагування на натискання кнопки: ~100–200 мс;
- Середній FPS обробки відео в реальному часі: до 20 кадрів/секунда при роздільній здатності 720p;
- Час повного циклу обробки (від запису до побудови координат): ~6–8 секунд для 10-секундного відео;
- Точність виявлення маркерів: 95–98% при статичному фоні та гарному освітленні;
- Кількість виявлених точок за сесію: 200–400 (в залежності від руху).
- Було встановлено, що система демонструє стабільну роботу за температур пристрою до 65–70°C, що не перевищує нормальні робочі межі для Raspberry Pi 4. GPIO-компоненти не виявляли відмов у тестах протягом 10 годин безперервної роботи.

У ході тестування було виявлено низку недоліків:

- При різкій зміні освітлення іноді знижується контрастність маркерів;
- У темному середовищі без ІЧ-підсвітки система дає більшу похибку (до 10%);
- При дуже швидких рухах трапляється «розмазування» кадру, що ускладнює обробку.
- Для мінімізації похибок було рекомендовано:
- Встановити стабільне освітлення сцени або інфрачервоне фонове підсвічування;

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

- Оптимізувати FPS запису до 30 для кращої деталізації швидких рухів;
- Використовувати цифрову стабілізацію кадру або автоекспозицію.

За результатами проведеного тестування можна зробити висновок, що розроблена система відповідає заданим технічним вимогам. Вона здатна точно реєструвати рухи актора, виділяти маркери, зберігати координати у форматі, придатному для подальшої обробки у Blender або іншому 3D-середовищі. Алгоритми розпізнавання забезпечують стабільну точність, а апаратна частина функціонує надійно навіть при тривалому використанні.

Таким чином, уся система — від запису до візуалізації — є повністю працездатною, масштабованою та готовою до інтеграції у реальні VFX-процеси створення анімаційного або ігрового кон

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Ергономічні проблеми безпеки життєдіяльності

Ергономіка є науковою дисципліною, що вивчає взаємодію людини з елементами системи, в якій вона функціонує. Основною метою ергономіки є оптимізація добробуту людини та загальної продуктивності системи шляхом проєктування робочих місць, обладнання та робочих процесів відповідно до фізичних та психологічних можливостей людини. Недостатня увага до ергономічних принципів призводить до виникнення численних проблем безпеки життєдіяльності, що проявляються у вигляді помилок персоналу, зниження ефективності праці, виникнення професійних захворювань та підвищення ризику нещасних випадків на виробництві [34].

Ергономічні проблеми класифікуються за кількома напрямками. Фізична ергономіка розглядає анатомічні, антропометричні, фізіологічні та біомеханічні характеристики людини у їхньому зв'язку з фізичною активністю. До основних проблем у цій галузі належать неоптимальні робочі пози, що характеризуються статичним напруженням або незручним положенням тіла; монотонні повторювані рухи, що призводять до перевантаження окремих груп м'язів та суглобів; надмірні силові зусилля при виконанні робочих операцій; нераціональна організація робочого простору, що змушує працівника здійснювати зайві рухи або приймати нефізіологічні пози. Також до фізичних ергономічних проблем відносять вплив несприятливих факторів виробничого середовища, таких як неадекватне освітлення, підвищений рівень шуму або вібрації, та некомфортні мікрокліматичні умови, що безпосередньо впливають на працездатність та безпеку.

					КС КРБ 123.209.00.00 ПЗ							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Ковальський О.С			Безпека життєдіяльності, основи охорони праці			Літ.	Арк.	Аркушів		
Перевірив		Жаровський Р.О								50		
Консульт.		Пилипець М.І.						ТНТУ, каф. КС, гр. СІ-41				
Н. Контр.		Луцук Н.С										
Затверд.		Осухівська Г.М.										

Когнітивна ергономіка зосереджена на психічних процесах, таких як сприйняття, пам'ять, мислення та моторна реакція, у контексті взаємодії людини з іншими елементами системи. Проблеми когнітивної ергономіки включають надмірне або недостатнє розумове навантаження, складність процесів прийняття рішень в умовах обмеженого часу або недостатньої інформації, невідповідність інтерфейсів "людина-машина" когнітивним можливостям оператора, що призводить до помилок, та високий рівень психоемоційного стресу. Неправильно спроектовані інформаційні дисплеї, нелогічна організація управляючих елементів та недостатнє навчання є типовими причинами виникнення когнітивних ергономічних проблем [35].

Організаційна ергономіка займається оптимізацією соціотехнічних систем, включаючи їхні організаційні структури, політику та процеси. Проблеми в цій сфері виникають через нераціональний розподіл робочих завдань, неоптимальні графіки роботи та відпочинку, неефективну комунікацію всередині колективу, недостатню командну взаємодію та недосконалість системи управління охороною праці. Монотонність праці, відсутність можливостей для професійного розвитку та низький рівень залученості працівників до процесів прийняття рішень також є суттєвими організаційно-ергономічними проблемами.

Наслідки ігнорування ергономічних проблем є негативними. Найбільш поширеними є розлади опорно-рухового апарату, такі як тендиніти, бурсити, синдром зап'ястного каналу, остеохондроз та інші хронічні захворювання м'язів, сухожиль і суглобів. Ергономічні недоліки суттєво підвищують ризик виникнення гострих травм через падіння, зіткнення або неправильне поводження з обладнанням. Зниження продуктивності та ефективності праці є прямим наслідком втоми, дискомфорту та помилок, спричинених неергономічними умовами. Це, у свою чергу, призводить до економічних втрат для підприємства через зниження якості продукції, збільшення кількості браку та зростання витрат на лікування та компенсації працівникам. Також Для виявлення та оцінки ергономічних ризиків застосовується комплекс

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

методів. Пряме спостереження за виконанням робочих операцій дозволяє ідентифікувати незручні пози, повторювані рухи та інші потенційно небезпечні фактори. Використання спеціалізованих чек-листів та опитувальників допомагає систематизувати збір даних та отримати суб'єктивну оцінку умов праці від самих працівників. Інструментальні методи включають прямі вимірювання параметрів робочого середовища (освітленість, шум, вібрація, мікроклімат), а також біомеханічний аналіз рухів та зусиль за допомогою спеціального обладнання. Аналіз статистичних даних щодо захворюваності та травматизму також надає важливу інформацію для ідентифікації проблемних зон.

Проектування робочих систем повинно базуватися на фундаментальних ергономічних принципах. Антропометрична відповідність передбачає врахування розмірів тіла людини при конструюванні обладнання та організації робочих місць, забезпечуючи можливість налаштування під індивідуальні параметри користувачів. Мінімізація статичних навантажень та незручних поз досягається шляхом забезпечення динамічного режиму роботи та можливості зміни положення тіла. Зменшення кількості повторюваних рухів та силових навантажень реалізується через механізацію та автоматизацію важких операцій, а також використання інструментів з ергономічним дизайном. Забезпечення адекватного освітлення робочої зони, що усуває відблиски та недостатню видимість, є критично важливим для запобігання зоровій втомі та помилкам. Інтерфейси систем управління повинні бути інтуїтивно зрозумілими, логічними та надавати чіткий зворотний зв'язок. Створення комфортних умов мікроклімату та акустичного середовища сприяє підтримці високої працездатності та зниженню загальної втоми.

Для комп'ютеризованої системи трекінгу руху актора за допомогою маркерного костюма ергономічні проблеми мають особливу актуальність та безпосередньо впливають на безпеку та ефективність процесу створення візуальних ефектів. Актор, одягнений у маркерний костюм, стикається з низкою фізичних ергономічних викликів. Вага та конструкція костюма,

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

розташування для комп'ютеризованої системи для створення візуальних ефектів у кіно за допомогою маркерного методу захоплення руху актора. ергономічні проблеми мають особливу актуальність та безпосередньо впливають на безпеку та ефективність процесу створення візуальних ефектів. Кіноактор, одягнений у маркерний костюм, стикається з низкою фізичних ергономічних викликів. Вага та конструкція костюма, розташування маркерів, гнучкість матеріалів – усі ці фактори визначають ступінь обмеження природних рухів, додаткове фізичне навантаження та потенційний дискомфорт. Тривале носіння костюма, особливо під час виконання динамічних сцен, може призвести до перегріву тіла, подразнення шкіри в місцях контакту з елементами костюма або маркерами. Неоптимальний дизайн костюма може створювати точки надмірного тиску на тіло актора. Для оператора системи трекінгу ергономічні проблеми пов'язані переважно з тривалою роботою за комп'ютером: статична поза, візуальне навантаження від моніторів, монотонні рухи при використанні пристроїв введення. Неправильна організація його робочого місця, включаючи висоту столу, крісла, розташування монітора та освітлення, неминуче призведе до швидкої втоми, зниження уваги та, як наслідок, до помилок у фіксації та обробці даних руху, а в довгостроковій перспективі – до розвитку професійних захворювань. Вирішення цих ергономічних проблем є критично важливим для забезпечення не тільки фізичного здоров'я та комфорту персоналу, але й для досягнення високої якості кінцевого продукту – візуальних ефектів, оскільки втома та дискомфорт актора негативно позначаються на природності його рухів, а помилки оператора можуть спотворити дані трекінгу.

4.2 Профілактика захворювань, які викликані напруженням органів, систем організму та невірним положенням тіла при роботі.

Професійна діяльність людини часто пов'язана з факторами, що спричиняють напруження окремих органів та систем організму, а також з

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

необхідністю тривалого перебування у нефізіологічних положеннях тіла. Такі умови праці є передумовою для розвитку широкого спектра професійних захворювань, зокрема розладів опорно-рухового апарату, захворювань периферичної нервової системи, серцево-судинних патологій та порушень зору. Профілактика цих захворювань є пріоритетним завданням охорони праці, спрямованим на збереження здоров'я та працездатності працівників.

До основних причин виникнення захворювань, пов'язаних з напруженням та невірним положенням тіла, належать статичні та динамічні перевантаження, локальні м'язові напруження, вимушені робочі пози, монотонні повторювані рухи з високою частотою, а також вібрація та мікрокліматичні умови. Тривале перебування у сидячому або стоячому положенні без можливості зміни пози призводить до застою крові, порушення живлення тканин та надмірного навантаження на хребет і суглоби. Виконання роботи з піднятими руками, нахиленим тулубом або повернутою головою спричиняє швидко втому м'язів ший, плечового поясу та спини, що з часом переростає у хронічний больовий синдром та дегенеративні зміни у відповідних структурах [36].

Розлади опорно-рухового апарату, такі як остеохондроз хребта, артрози, тендовагініти, епіконділіти, синдром зап'ястного каналу, є найпоширенішими наслідками впливу зазначених шкідливих факторів. Фізіологічною основою цих захворювань є кумулятивний ефект мікротравм м'язових волокон, сухожиль, зв'язок та суглобових хрящів, що виникають внаслідок постійного перенапруження або неадекватного розподілу навантаження. Порушення кровопостачання та іннервації уражених ділянок поглиблює патологічний процес, призводячи до хронізації захворювання та поступової втрати функції.

Система профілактичних заходів щодо запобігання захворюванням, викликаним напруженням та невірним положенням тіла, повинна мати комплексний характер та включати організаційні, інженерно-технічні та медико-біологічні аспекти. Пріоритет надається інженерно-технічним заходам, спрямованим на усунення або максимальне зниження рівня впливу

					КС КРБ 123.209.00.00 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

шкідливих факторів безпосередньо на робочому місці. Це досягається шляхом раціоналізації конструкції обладнання, інструментів та робочих меблів. Робочі місця повинні проектуватися з урахуванням антропометричних даних працівників, забезпечуючи можливість регулювання висоти робочої поверхні, сидіння, положення моніторів та інших елементів. Використання спеціальних підставок для ніг, підлокітників, ергономічних клавіатур та маніпуляторів типу "миша" сприяє підтримці фізіологічно правильної пози та зменшенню статичного напруження. Механізація та автоматизація важких та монотонних операцій дозволяє суттєво знизити фізичні навантаження на працівника [37].

Організаційні заходи включають раціоналізацію режимів праці та відпочинку. Встановлення регламентованих перерв протягом робочої зміни, під час яких працівники можуть виконувати спеціальні комплекси фізичних вправ для зняття м'язового напруження та втоми, є обов'язковою умовою. Зміна видів виконуваних робіт (ротація) протягом дня допомагає уникнути перевантаження одних і тих самих груп м'язів. Важливим елементом є професійний відбір та навчання працівників безпечним методам та прийомам виконання робіт, правильному використанню обладнання та засобів індивідуального захисту. Інструктажі з охорони праці повинні містити детальну інформацію про потенційні ризики та методи їх запобігання.

Профілактика захворювань, пов'язаних з невірним положенням тіла, вимагає особливої уваги до формування правильних рухових стереотипів. Працівники повинні бути навчені підтримувати оптимальну поставу під час виконання робочих операцій, уникати тривалих статичних поз та незручних положень. Для цього використовуються наочні посібники, тренінги та індивідуальні консультації. Регулярні фізичні вправи, спрямовані на зміцнення м'язового корсету, підвищення гнучкості хребта та рухливості суглобів, є ефективним засобом профілактики.

Профілактика напруження зорового аналізатора при роботі з дисплеями включає забезпечення належних параметрів освітлення на робочому місці, правильне розташування монітора (верхній край екрана на рівні очей або

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

трохи нижче, відстань 60-70 см), використання моніторів з високою роздільною здатністю та антивідблисковим покриттям. Регулярні перерви для виконання гімнастики для очей допомагають зняти зорову втому.

Медико-біологічні заходи передбачають проведення попередніх (при прийомі на роботу) та періодичних медичних оглядів для своєчасного виявлення ранніх ознак професійних захворювань та визначення придатності працівника до виконання відповідної роботи. За результатами оглядів розробляються індивідуальні рекомендації щодо оздоровлення та реабілітації. Важливу роль відіграє також санітарно-просвітницька робота, спрямована на підвищення рівня обізнаності працівників щодо факторів ризику та методів їх профілактики [38].

У контексті комп'ютеризованої системи трекінгу руху актора за допомогою маркерного костюма, профілактика захворювань, викликаних напруженням органів та невірним положенням тіла, є надзвичайно важливою для забезпечення здоров'я та довготривалої працездатності як акторів, так і технічного персоналу. Для актора, який використовує маркерний костюм, ризики пов'язані з тривалим носінням потенційно незручного або обмежуючого рухи спорядження, виконанням специфічних, іноді амплітудних або повторюваних рухів під час сеансів захоплення руху. Це створює передумови для перенапруження м'язів, зв'язок та суглобів, особливо якщо костюм не ідеально підігнаний або має значну вагу. Профілактичні заходи для актора повинні включати ретельний підбір та індивідуальне припасування костюма, використання легких та "дихаючих" матеріалів, забезпечення достатньої гнучкості костюма для виконання необхідного діапазону рухів, а також обов'язкові розминки перед роботою та заминки після неї. Регламентовані перерви для відпочинку та зняття костюма є критичними. Для оператора системи, робота якого пов'язана з тривалим перебуванням у сидячому положенні за комп'ютером, моніторингом процесу та обробкою даних, основними ризиками є розвиток захворювань хребта, синдрому зап'ястного каналу та зорової втоми. Профілактика для оператора полягає в

					КС КРБ 123.209.00.00 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

облаштуванні ергономічного робочого місця з можливістю регулювання висоти стільця та столу, правильного розташування моніторів, використання ергономічних пристроїв введення, а також дотриманні режиму праці та відпочинку з регулярними перервами для фізичної активності та вправ для очей. Ігнорування цих профілактичних заходів неминуче призведе до зниження працездатності персоналу, збільшення кількості помилок при захопленні та обробці руху, і, як наслідок, до погіршення якості візуальних ефектів та зростання виробничих витрат через лікарняні та необхідність заміни працівників.

					КС КРБ 123.209.00.00 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У рамках виконання кваліфікаційної роботи було розроблено, змодельовано та реалізовано комп'ютеризовану систему для захоплення руху актора з використанням маркерного методу на базі платформи Raspberry Pi 4. В процесі проектування було проведено аналіз технічних підходів до захоплення руху, вибір апаратного забезпечення, розробка електричної схеми, створення програмного забезпечення та його тестування в умовах, наближених до реальних.

Система виявилась працездатною, стабільною та придатною до інтеграції у більш складні середовища створення анімації, зокрема до Blender. Усі компоненти функціонували відповідно до заданих вимог: камера успішно фіксувала відео, програмна частина здійснювала обробку кадрів, розпізнавала маркери та зберігала координати у зручному для подальшого використання форматі. Результати імпорту до Blender підтвердили точність зчитування траєкторій руху.

Проведені експерименти показали високий рівень ефективності алгоритмів, що дозволяє рекомендувати створену систему для подальшого вдосконалення або застосування у сфері візуальних ефектів, 3D-анімації, ігор або освітніх проєктів. Робота підтвердила актуальність використання доступних апаратних рішень і відкритого програмного забезпечення у сфері цифрової анімації.

Таким чином, поставлені на початку роботи цілі досягнуто повністю, а розроблена система відповідає критеріям надійності, функціональності та адаптивності.

.

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жаровський Р.О., Луцик Н.С., Осухівська Г.М., Паламар А.М., Тиш С.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. Микитишин А.Г., Митник М.М., Стухляк П.Д., Пасічник В.В. Комп'ютерні мережі. Книга 2. Львів: «Магнолія 2006», 2024. 328 с.
3. Kharchenko A., Bodnarchuk I., Yatsysyn 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.
4. Lupenko S., Orobchuk O., Stadnik N., Zozulya A. Modeling and signals processing using cyclic random functions. In Proceedings of the 13th IEEE International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). Lviv, Ukraine, 11–14 September, 2018. P. 360–366.
5. Shabliy N., Lupenko S., Lutsyk N., Yasniy O., Malyshevskaya O. Keystroke dynamics analysis using machine learning methods. Applied Computer Science. 2021. Vol. 17, No. 4. P. 75-83.
6. Velychko D., Osukhivska H., Palaniza Y., Lutsyk N., Sobaszek L. Artificial Intelligence Based Emergency Identification Computer System. Advances in Science and Technology Research Journal, 18 no. 2, 2024, P. 296-304.
7. Voloshchuk A., Velychko D., Osukhivska H., Palamar A. Computer system for energy distribution in conditions of electricity shortage using artificial intelligence. CEUR Workshop Proceedings, 2nd International Workshop on Computer Information Technologies in Industry 4.0 (CITI 2024), Ternopil, Ukraine, June 12-14, 2024. Vol. 3742 P. 66-75.
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

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

Bioinformatics and Applied Information Technologies (BAIT 2024), Zboriv, Ukraine, October 02-04, 2024. Vol. 3842. P. 258-267.

9. Yasniy O., Lutsyk N., Demchyk V., Osukhivska H., Malyshevskaya O. The prediction of structural properties of Ni-Ti shape memory alloy by the supervised machine learning methods. ITTAP 2023: 73–78. <https://ceur-ws.org/Vol-3628/short1.pdf>

10. Yatsyshyn V., Pastukh O., Palamar A., Zharovskyy R. Technology of relational database management systems performance evaluation during computer systems design. Scientific Journal of TNTU, Ternopil, Ukraine, 2023. Vol. 109, No 1. P. 54–65.

11. Yatsyshyn V., Pastukh O., Zharovskiy R., Shabliy N. Software tool for productivity metrics measure of relational database management system. Mathematical Modeling. No 1 (48). 2023. P. 7-17.

12. Zozulia A., Lytvynenko I., Lutsyk N., Lupenko S., Yasniy O. Method of vector rhythmcardiosignal automatic generation in computer-based systems of heart rhythm analysis. Visnyk of TNTU. 2020. Vol. 97, P. 122-132.

13. Романов Д.В., Осухівська Г.М., Паламар А.М. Система управління зовнішнім освітленням на основі Інтернету речей. Актуальні задачі сучасних технологій : збірник тез доповідей X міжнародної науково-практичної конференції молодих учених та студентів (Тернопіль, 24-25 листопада 2021 року), Тернопіль: ТНТУ, 2021. С. 120.

14. Романов Д.В., Осухівська Г.М., Паламар А.М. Функціональна схема системи керування зовнішнім освітленням на основі технології LoRa. Матеріали IX науково-технічної конференції "Інформаційні моделі, системи та технології" Тернопільського національного технічного університету імені Івана Пулюя (Тернопіль, 8–9 грудня 2021 року), Тернопіль: ТНТУ, 2021. С. 124. (дата звернення 03.06.25)

15. Слюз І., Жаровський Р. Принципи та основні етапи комплексного тестування комп'ютерної інформаційної системи. Матеріали X науково-технічної конференції Тернопільського національного технічного

					КС КРБ 123.209.00.00 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

університету імені Івана Пулюя «Інформаційні моделі системи та технології» (7-8 грудня 2022 року). Тернопіль: ТНТУ. 2022. С. 93.

16. Харченко О., Яцишин В. Розробка та керування вимогами до програмного забезпечення на основі моделі якості. Вісник ТДТУ. Тернопіль, 2009. Т. 14. №1. С. 201-207. (дата звернення 01.06.25)

17. Gleicher M. Retargetting Motion to New Characters. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH). 1998. p. 33–42.

18. Welman C. Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation. Simon Fraser University, MSc Thesis, 1993.

19. Ma W. C., & Deng Z. Motion Graphs for Real-Time Human Animation. Computer Animation and Virtual Worlds, 2013. (дата звернення 03.06.25)

20. Eberly D.H. 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics. 2nd ed. CRC Press, 2006. 752 p. (дата звернення 02.06.25)

21. Szeliski R. Computer Vision: Algorithms and Applications. Springer, 2010. 812 p.

22. Jason van Gumster. Blender For Dummies. 4th ed. Wiley, 2020. 640 p. (дата звернення 04.06.25)

23. McKinney M. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media, 2022.

24. Thomas W., Johnston O. The Illusion of Life: Disney Animation. Disney Editions, 1995. 576 p.

25. Yamamoto T., Koga Y., Kishino F. Advanced Human Pose Estimation in Motion Capture Systems. In: Journal of Graphics Tools, 2015. (дата звернення 03.06.25)

26. Foley J.D., van Dam A., Feiner S.K., Hughes J.F. Computer Graphics: Principles and Practice. 3rd ed. Addison-Wesley, 2013. 1264 p.

27. Rokoko. Smartsuit Pro II Technical Documentation. URL: <https://www.rokoko.com> (дата звернення 02.06.25)

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

28. Vicon Motion Systems. Vicon Nexus User Guide. URL: <https://docs.vicon.com>
29. Xsens Technologies. MVN User Manual. URL: <https://www.xsens.com/support>
30. OpenMoCap Відкрита система захоплення руху. URL: <https://github.com/OpenMoCap> (дата звернення 04.06.25)
31. Parent R. Computer Animation: Algorithms and Techniques. 3rd ed. CRC Press, 2012. 516 p.
32. Rickitt R. Special Effects: The History and Technique. 2nd ed. Billboard Books, 2007. 368 p. (дата звернення 04.06.25)
33. Menache A. Understanding Motion Capture for Computer Animation and Video Games. 2nd ed. Morgan Kaufmann, 2010. 276 p.
34. Ергономіка. Навчально-методичний посібник. URL: <https://www.scribd.com/document/721373705/Ergonomika> (дата звернення: 07.06.2025).
35. Закон України "Про охорону праці". URL: <https://zakon.rada.gov.ua/laws/show/2694-12#Text> (дата звернення: 04.06.2025).
36. Державні санітарні норми та правила "Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу". URL: <https://zakon.rada.gov.ua/laws/show/z0472-14#Text> (07.06.2025)
37. Основи охорони праці. URL: https://kipt.com.ua/wp-content/uploads/2019/Основи-охорони-праці_Винокурова-Л.Е._2001.pdf
38. Наказ МОЗ України від 21.05.2007 № 246 "Про затвердження Порядку проведення медичних оглядів працівників певних категорій" URL: <https://zakon.rada.gov.ua/laws/show/z0846-07#Text> (дата звернення: 07.06.2025).

					КС КРБ 123.209.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

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'ютеризована система для створення візуальних ефектів у кіно за допомогою маркерного методу захоплення руху актора.

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. Жаровський Р.О

_____ Ковальський О.С

«____» _____ 2025 р.

«____» _____ 2025 р.

Тернопіль 2025

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютеризована система для створення візуальних ефектів у кіно за допомогою маркерного методу захоплення руху актора.»

Умовне позначення кваліфікаційної роботи: КС КРБ 123.209.41.00

1.2 Виконавець

Студент групи СІ-41, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Ковальський Олександр Станіславович

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

Камера захоплення рухів маркерним способом призначений для:

- Анімування 3D-персонажів.
- Зберігання файлів формату H256.

2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка та налаштування системи для створення візуальних ефектів у кіно за допомогою маркерного методу захоплення руху актора на основі одноплатного мікрокомп'ютера Raspberry Pi.

2.3 Характеристика об'єкту

Розробляюваний метод створення візуальних ефектів за допомогою маркерного методу захоплення рухів призначено для використання в домашніх, напівпрофесійних досягнень.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

3.1.1 Вимоги до структури та функціонування системи

Raspberry повинен складатись з:

- Програмного коду.
- Камера захоплення рухів.
- Ethernet коннектор
- USB-C (3.0)

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, калібровки 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.

3.1.4 Вимоги по діагностуванню системи

Для діагностування системи використовуються інструменти діагностування основних процесів системи, які вмонтовані в операційну систему 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, додаткових камер чи LED-інфрачервоних, маркерів ArUco а також за допомогою офіційної системи Raspberry Pi OS (Lite-версія) та Python API.

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).

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 +30°C;
- відносну вологість повітря при 25°C в межах від 25% до 75%;
- атмосферний тиск 760 ± 25 мм рт. ст.

Періодичне технічне обслуговування і тестування технічних засобів повинні включати обслуговування і тестування всіх використовуваних засобів, датчики, контролери, системи передачі даних, камери, 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”.

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. Вимоги до функцій (завдань), що виконуються системою: забезпечення зручного інтерфейсу, забезпечення зберігання медіафайлів, висока якість камери, захоплення ArUco-маркерів, 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

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка технічного завдання</i>	<i>27.01 – 02.02</i>	<i>Викон.</i>
2.	<i>Підбір джерел про створення вузальних ефектів у кіно за допомогою маркерного методу та захоплення руху</i>	<i>11.05.2025-20.05.2025</i>	<i>Викон.</i>
3.	<i>Опрацювання джерел по темі кваліфікаційної роботи</i>	<i>01.05.2025</i>	<i>Викон.</i>
4.	<i>Розробка структури та аналіз вибраного приладу</i>	<i>05.06.2025</i>	<i>Викон.</i>
5.	<i>Розробка програмного коду</i>	<i>07.06.2025</i>	<i>Викон.</i>
6.	<i>Тестування та оцінка приладу для захоплення руху</i>	<i>11.06.2025</i>	<i>Викон.</i>
7.	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>12.06.2025</i>	<i>Викон.</i>
8.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>20.06.2025</i>	<i>Викон.</i>
9.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>09.06.2025-15.06.2025</i>	<i>Викон.</i>
10	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>16.06.2025-22.06.2025</i>	<i>Викон.</i>
11	<i>Захист кваліфікаційної роботи бакалавра</i>	<i>23.06.2025</i>	<i>Викон.</i>

6 Додаткові умови виконання кваліфікаційної роботи

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

Додаток Б

Лістинг коду програми

```
import cv2
cap = cv2.VideoCapture('output.mp4')
while True:
    ret, frame = cap.read()
    if not ret:
        break
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    _, thresh = cv2.threshold(gray, 200, 255,
cv2.THRESH_BINARY)
    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    for cnt in contours:
        if cv2.contourArea(cnt) > 5:
            x, y, w, h = cv2.boundingRect(cnt)
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0,
255, 0), 2)
    cv2.imshow("Detected Markers", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

```
import cv2
import csv
cap = cv2.VideoCapture('output.h264')
with open('markers.csv', 'w', newline='') as f:
    writer = csv.writer(f)
    while True:
        ret, frame = cap.read()
        if not ret:
            break
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        _, thresh = cv2.threshold(gray, 200, 255,
cv2.THRESH_BINARY)
        contours, _ = cv2.findContours(thresh,
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        for cnt in contours:
            if cv2.contourArea(cnt) > 5:
                x, y, w, h = cv2.boundingRect(cnt)
                writer.writerow([x + w//2, y + h//2])
cap.release()
```

```
import RPi.GPIO as GPIO

BUTTON_PIN = 23

GPIO.setmode(GPIO.BCM)

GPIO.setup(BUTTON_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)

print("Очікування натискання кнопки...")

while GPIO.input(BUTTON_PIN):

    time.sleep(0.1)

print("Кнопку натиснуто, початок запису")
```

Додаток В

Перелік елементів до електричної принципової схеми

[illegible]