

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

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

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

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: *Комп'ютерна система керування світлофорами*

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

(шифр і назва спеціальності)

	<u>Вовчак Р.А.</u> (підпис) (прізвище та ініціали)
Керівник	<u>Шингера Н.Я.</u> (підпис) (прізвище та ініціали)
Нормоконтроль	<u>Луцик Н.С.</u> (підпис) (прізвище та ініціали)
Завідувач кафедри	<u>Осухівська Г.М.</u> (підпис) (прізвище та ініціали)
Рецензент	<u>Бревус В.М.</u> (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 Вовчака Романа Андрійовича
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система керування світлофорами

Керівник роботи Шингера Наталія Ярославівна
(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 18 червня 2025р.

3. Вихідні дані до роботи Технічне завдання

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

Вступ. 1. Аналіз технічного завдання

2. Проєктна частина

3. Практична частина

4. Безпека життєдіяльності, основи охорони праці.

Висновки

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

1. Структурна схема

2. Схема електрична принципова

3. Блок-схема алгоритму роботи програми виявлення руху (A2)

4. Блок-схема алгоритму роботи програми центрального контролера (A2)

5. Результат роботи

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>27.01.2025р. – 02.02.2025р.</i>	<i>Викон.</i>
2.	<i>Робота над основними розділами – Аналіз технічного завдання</i>	<i>28.02.2025р. – 15.03.2025р.</i>	<i>Викон.</i>
3.	<i>Робота над основними розділами – Проектна частина</i>	<i>16.03.2025р. – 07.04.2025р.</i>	<i>Викон.</i>
4.	<i>Робота над основними розділами – Програмна частина</i>	<i>8.04.2025р. – 10.05.2025р.</i>	<i>Викон.</i>
5.	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>11.05.2025р. – 20.05.2025р.</i>	<i>Викон.</i>
6.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>25.05.2025р. – 9.06.2025р.</i>	<i>Викон.</i>
7.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>9.06.2025р. – 15.06.2025р.</i>	<i>Викон.</i>
8.	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>16.06.2025р. – 22.06.2025р.</i>	<i>Викон.</i>
9.	<i>Захист кваліфікаційної роботи бакалавра</i>	<i>25.06.2025р.</i>	

Студент

(підпис)

Вовчак Р.А.

(прізвище та ініціали)

Керівник роботи

(підпис)

Шингера Н.Я.

(прізвище та ініціали)

АНОТАЦІЯ

Вовчак Р.А. Комп'ютерна система керування світлофорами: робота на здобуття кваліфікаційного ступеня бакалавра: спец. 123 — комп'ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2025.

Ключові слова: комп'ютерна система, мікрокомп'ютер, комп'ютерний зір, адаптивна система керування світлофором, рух транспорту, камера спостереження.

У кваліфікаційній роботі розроблено комп'ютерну систему керування світлофорами на базі мікрокомп'ютерів Raspberry Pi з використанням технологій комп'ютерного зору. Система забезпечує автоматичне виявлення транспортних засобів на перехресті з двома дорогами (головною та другорядною) за допомогою камер, встановлених над смугами руху.

Збір інформації про наявність автомобілів виконується за рахунок програми яка порівнює поточний знімок з попереднім, і якщо зміни суттєві то відсилається команда через протокол UDP, після чого центральний контролер аналізує ситуацію і формує команди перемикання світлофорів, що передаються по ТСР. Реалізовано алгоритм пріоритету головної дороги, логіку перемикання в залежності яка ситуація на дорозі в поточний момент

Робота демонструє ефективність застосування доступних компонентів та мікрокомп'ютерів у задачах інтелектуального регулювання дорожнього руху, що значно покращує якість роботи дорожньої інфраструктури.

ANNOTATION

Vovchak R.A. Computer traffic light control system: Bachelor's Graduation Thesis: speciality 123 — computer engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2025.

Keywords: computer system, microcomputer, computer vision, adaptive traffic light control system, traffic, surveillance camera.

In the qualification work, a computer traffic light control system based on Raspberry Pi microcomputers using computer vision technologies has been developed. The system provides automatic detection of vehicles at the intersection of two roads (main and secondary) using cameras installed above the traffic lanes.

Information about the presence of cars is collected by a program that compares the current image with the previous one, and if the changes are significant, a command is sent via the UDP protocol, after which the central controller analyzes the situation and generates traffic light switching commands transmitted via TCP. The algorithm for the priority of the main road, the switching logic depending on the current road situation, has been implemented

The work demonstrates the effectiveness of using available components and microcomputers in the tasks of intelligent traffic control, which significantly improves the quality of road infrastructure.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ	10
1.1 Аналіз вимог до комп'ютерної системи керування світлофором	10
1.1.1 Призначення комп'ютерної системи керування світлофором	10
1.1.2 Вимоги до комп'ютерної системи керування світлофором.	11
1.1.3 Вимоги до апаратного забезпечення комп'ютерної системи керування світлофором	13
1.2 Аналіз можливих рішень поставленого завдання до комп'ютерної системи керування світлофором.....	14
1.2.1 Огляд існуючих систем комп'ютерного керування світлофорами	14
1.2.2 Приклади існуючих систем.....	16
1.3 Особливості, призначення і застосування комп'ютерного зору на регульованих перехрестях.....	18
РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА.....	20
2.1 Проектування архітектури комп'ютерної системи керування світлофорами	20
2.2 Характеристики і структура Raspberry Pi.....	22
2.3 Проектування схеми та обґрунтування вибору апаратної частини комп'ютерної системи.....	27
2.3.1 Модуль камери Raspberry Pi	32
2.3.2 Raspberry Pi Expansion Board Power Relay	38

					КС КРБ 123.198.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Вовчак Р.А.			Комп'ютерна система керування світлофорами		
Перевір.		Шингера Н.Я.					
Реценз.		Бревус В.М.					
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркуші
						6	
					ТНТУ, каф. КС, гр. СІ-41		

2.3.3	LED Traffic Light	40
2.3.4	Перетворювач напруги	41
РОЗДІЛ 3 ПРОГРАМНА ЧАСТИНА		42
3.1	Розробка алгоритму	42
3.2	Розробка програмного забезпечення для виявлення руху по смугах.....	46
3.3	Розробка програмного забезпечення для центрального керування.....	49
3.4	Розробка програмного забезпечення для керування реле	52
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ..		55
4.1	Долікарська допомога при переломах	55
4.2	Вплив світла на людину	58
ВИСНОВКИ.....		61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		62
Додаток А Технічне завдання		
Додаток Б Перелік елементів		
Додаток В Лістинг коду		

					КС КРБ 123.198.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

СПИСОК СКОРОЧЕНЬ

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

HDMI – High Definition Multimedia Interface

RAM – Random Access Memory

GPIO – General Purpose Input/Output

SoC – System-on-a-chip

USB – Universal Serial Bus

BCM – Broadcom Chip Model

PoE – Power over Ethernet

CSI – Camera Serial Interface

RPi – Raspberry Pi

					<i>КС КРБ 123.198.00.00 ПЗ</i>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У сучасному світі питання безпеки дорожнього руху набуває особливої актуальності. Зі зростанням кількості транспортних засобів та інтенсивністю руху виникає необхідність у впровадженні інтелектуальних систем керування, здатних забезпечити ефективну організацію дорожнього трафіку та зниження аварійності на перехрестях. Саме тому активно розвиваються автоматизовані комп'ютерні системи керування світлофорами, які поєднують в собі апаратну та програмну складові.

Особливої уваги потребують міські перехрестя, де традиційні алгоритми перемикавання сигналів світлофора часто виявляються недостатньо гнучкими в умовах змінного транспортного навантаження. Впровадження комп'ютерного керування дозволяє адаптувати роботу світлофорів у реальному часі, враховуючи наявність або відсутність транспорту на окремих смугах руху. Це сприяє зменшенню заторів, скороченню часу простою та підвищенню загальної пропускної здатності перехресть.

У межах даної кваліфікаційної роботи розглядається проєкт створення системи керування світлофорами з використанням мікрокомп'ютера Raspberry Pi 4 Model B та камери. Пропонується рішення, при якому кожна смуга обладнується окремим вузлом виявлення транспортного засобу, а центральний контролер приймає рішення щодо перемикавання сигналів. Такий підхід дозволяє досягти гнучкого, масштабованого й економічно ефективного способу організації руху на регульованих перехрестях.

					<i>КС КРБ 123.198.00.00 ПЗ</i>	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вимог до комп'ютерної системи керування світлофором

1.1.1 Призначення комп'ютерної системи керування світлофором

Комп'ютерна система керування світлофорами призначена для автоматичного регулювання роботи світлофорів регульованих перехресть, з метою забезпечення ефективного та безпечного руху транспорту. Основне завдання цієї системи полягає в оптимізації потоку транспорту в залежності від поточної ситуації на дорогах. Це досягається завдяки використанню камер спрямованих на конкретні смуги дороги а також обробці даних у реальному часі.

Основна мета системи полягає в оптимізації руху транспортних засобів шляхом регулювання тривалості фаз зеленого та червоного сигналу світлофора в залежності від потоку транспорту і включення світлофорів тільки в тих напрямках руху, в яких це потрібно. Завдяки адаптивному керуванню сигналами світлофорів мінімізується кількість заторів на регульованих перехрестях.

Не менш важливим є покращення безпеки на дорогах. Система призначена для зменшення випадків, коли через зупинки і довгі простої на кожному світлофорі, через людський фактор і можливу економію часу, водії намагаються “проскочити” на жовтий або червоний сигнал світлофора, створюючи цим велику загрозу як для себе так і для інших учасників дорожнього руху.

					КС КРБ 123.198.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Вовчак Р.А.			Аналіз технічного завдання		
Перевір.		Шингера Н.Я.					
Реценз.		Бревус В.М.					
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркушів
						10	
					ТНТУ, каф. КС, гр. СІ-41		

Транспортні засоби оснащені двигунами внутрішнього згорання виробляють найбільше викидів на початку руху, тому зменшення кількості вимушених зупинок на червоний сигнал світлофора і скорочення часу перебування автомобілів в заторах суттєво зменшує рівень викидів шкідливих речовин. Також в умовах населеного пункту важливим чинником є зменшення рівня шуму від транспортних засобів.

Підвищення пропускної здатності перехресть сприяє більш ефективному використанню дорожньої інфраструктури, що впливає на суттєве покращення функціонування громадського транспорту та служб екстреної допомоги.

Система повинна бути підключена до центру управління, з можливістю дистанційного ручного керування світлофором. Це потрібно для ситуацій, таких як: дорожньо-транспортна пригода, перекриття руху на конкретній ділянці дороги або інші ситуації які неможливо передбачити. Також це дає змогу примусово перемикає сигнали світлофора на користь служб екстреної допомоги, створюючи для них маршрут без перешкод, для швидкого і безпечного прибуття на місце виклику.

1.1.2 Вимоги до комп'ютерної системи керування світлофором.

Для ефективної та безпечної роботи комп'ютерної системи керування світлофорами необхідно врахувати низку технічних, функціональних та експлуатаційних вимог. Ці вимоги забезпечують правильну роботу системи в умовах постійно змінюваного дорожнього руху та мінливих зовнішніх факторів.

Функціональні вимоги:

– автоматичне регулювання світлофорів. Система повинна автоматично регулювати тривалість фаз світлофорів залежно від потоку транспорту. Також потрібна бути можливість встановлення алгоритмів для

					КС КРБ 123.198.00.00 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

- адаптивного регулювання світлофорів, з урахуванням часу доби та погодних умов;
- інтерфейс для ручного управління. Наявність централізованої панелі управління, через яку оператор може вносити зміни до алгоритмів роботи світлофорів в разі потреби (наприклад, при аварійних ситуаціях або проведенні дорожніх робіт);
- підключення до камер відеоспостереження. Система повинна бути підключена до модулів камер спостереження, збирати дані, аналізувати їх і передавати конкретні команди;
- моніторинг стану світлофорів. Система повинна надавати в реальному часі інформацію про поточний стан світлофорів (червоне, зелене, жовте світло) для операторів в центрі управління [2].

Технічні вимоги:

- надійність і безпека. Система повинна бути стійкою до збоїв, з можливістю автоматичного відновлення роботи після виникнення проблем (наприклад, при відключенні електроживлення або збоєм в мережі);
- достатня пропускна здатність. Система повинна мати можливість обробляти великі обсяги даних у реальному часі, щоб гарантувати швидке реагування на зміни ситуації на дорозі;
- масштабованість. Можливість додавання нових світлофорів до системи без значних змін у її інфраструктурі;
- сумісність з різними платформами, а саме підтримка роботи з різними типами апаратних засобів, таких як мікроконтролери, камери та плати розширення.

Експлуатаційні вимоги:

- простота обслуговування. Система повинна легко налаштовуватися та підтримуватися. Для цього необхідна наявність інтуїтивно зрозумілого інтерфейсу для налаштування параметрів роботи світлофорів;

					КС КРБ 123.198.00.00 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

- можливість оновлення програмного забезпечення. Підтримка оновлення програмного забезпечення для покращення алгоритмів керування або виправлення помилок;
- моніторинг і звітність. Наявність функцій для збору статистики та аналізу роботи світлофорів (Наприклад, частота зміни світла, середній час затримки, ефективність управління потоком);
- адаптація до умов експлуатації. Система повинна витримувати вплив різних погодних умов (підвищена вологість, температура, пил, дощ) та працювати без перебоїв в різних кліматичних умовах.

1.1.3 Вимоги до апаратного забезпечення комп'ютерної системи керування світлофором

Апаратне забезпечення є критичним елементом комп'ютерної системи керування світлофорами, оскільки саме воно забезпечує безперервне функціонування системи, збір даних з навколишнього середовища та фізичне керування світловими сигналами. Вимоги до апаратного забезпечення охоплюють характеристики обчислювальних пристроїв, сенсорів, виконавчих механізмів і комунікаційних модулів.

Основним апаратним компонентом є контролер, на якому базується вся система [1]. Для цього проекту було обрано мікрокомп'ютер Raspberry Pi 4 (model B), якого достатньо для керування роботою світлофорів, обробки даних від комплектних камер і передачі команд. Він має достатню кількість цифрових і аналогових входів/виходів і GPIO-пінів для керування світлодіодами/реле.

Щодо світлофорних модулів, то найкраще підходять світлодіодні лампи червоного, жовтого та зеленого кольору. В них низьке енергоспоживання, висока яскравість і довговічність. Також вони мають мати захист від вологи та пилу (ступінь захисту IP65 або вище). Перемикання електроживлення до

					<i>КС КРБ 123.198.00.00 ПЗ</i>	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

світлофорних ламп має відбуватися за допомогою релейних модулів, які повинні бути надійно ізольованими. Основною вимогою до них є швидка реакція на сигнали контролера.

Крім підключення до основної електромережі (220v), має бути встановлене резервне живлення, на випадок знеструмлення.

Додаткові вимоги до апаратного забезпечення:

- всі компоненти мають бути стійкими до кліматичних умов і бути розрахованими на роботу в умовах дощу, снігу, пилу, вітру, різких перепадів температур;
- компоненти повинні легко замінюватися або оновлюватися без повного демонтажу системи, наприклад завдяки використанню плат розширення для мікрокомп'ютера Raspberry Pi, які підключаються простим стикуванням GPIO пінів і фіксуванням чотирма стійками;
- вузли, що знаходяться на вулиці (контролери, реле, блок живлення), повинні бути розміщені у герметичних корпусах з рівнем захисту не нижче IP65;
- конструкція світлофорних стійок і коробів має дозволяти швидке обслуговування персоналом.

1.2 Аналіз можливих рішень поставленого завдання до комп'ютерної системи керування світлофором

1.2.1 Огляд існуючих систем комп'ютерного керування світлофорами

На сьогодні існує велика кількість систем комп'ютерного керування світлофорами, які застосовуються у містах по всьому світу. Вони відрізняються за рівнем інтелектуальності, масштабом, алгоритмами керування та способами взаємодії з іншими інфраструктурними компонентами. У цьому розділі подано короткий огляд найбільш відомих

					КС КРБ 123.198.00.00 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

типів і прикладів таких систем.

Світлофори можна класифікувати за принципом роботи на фіксовані, адаптивні та інтелектуальні.

Фіксована (статичні) система керування працює за попередньо заданими таймерами, які не змінюються в реальному часі. Це найпоширеніший тип світлофора в Україні. До його переваг можна віднести найнижчу вартість з усіх інших можливих типів, простоту обслуговування і відсутність складних обчислювальних операцій. Серед недоліків, основним є статичні таймери перемикачів сигналів світлофора, що не враховують поточний трафік. Через це світлофори, через зміну потоку машин, часто створюють затори на дорозі. Тому щоб покращити ситуацію, робочому персоналу доводиться проводити заміри трафіку вручну, і змінювати таймери в системі світлофору. Але через швидкі зміни кількості автомобілів на дорозі, ці дані швидко застарівають.

Адаптивна система керування світлофорами реагує на зміну дорожньої обстановки за допомогою датчиків та алгоритмів адаптації. Ця система дорожча за фіксовану, але набагато продуктивніша. Вона фіксує поточну ситуацію на дорозі, і після обчислень змінює сигнали світлофора. До переваг можна віднести збільшення ефективності функціональності перехрестя завдяки системі яка сама адаптується до умов на дорозі, не потрібно вручну вносити зміни для покращення роботи перехрестя.

Інтелектуальна транспортна система інтегрує дані з багатьох джерел (камери, GPS, сенсори, громадський транспорт) і використовує штучний інтелект і машинне навчання для прогнозування заторів і динамічного керування. Ця система є одна з найдорожчих, і вимагає масштабної роботи з містом. Вона має бути підключена до великої кількості баз даних, і проводити багато обчислень. Перевага даної системи є можливість різних інтеграцій, таких як перевага руху для громадського транспорту, відслідковування потоку машин до під'їзду до перехрестя а також об'єднання різних перехресть в одну систему.

					КС КРБ 123.198.00.00 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

Переваги та недоліки існуючих систем керування світлофорами представлені в таблиці 1.1.

Таблиця 1.1 – Типи систем керування світлофорами

Тип Системи	Переваги	Недоліки
Фіксовані	Простота, низька вартість, надійність	Не реагують на реальний трафік, збільшують час очікування
Адаптивні	Ефективність, зменшення заторів, масштабованість	Висока вартість впровадження, залежність від сенсорів
Інтелектуальні	Реакція в реальному часі, автономність, гнучкість	Складність налаштування, висока обчислювальна потреба

1.2.2 Приклади існуючих систем

Однією з найвідоміших адаптивних систем керування дорожнім рухом є SCOOT (Split Cycle Offset Optimization Technique), розроблена у Великій Британії лабораторією Transport Research Laboratory (TRL). Система працює в режимі реального часу, змінюючи тривалість світлофорних фаз відповідно до інтенсивності руху, яку вона визначає за допомогою сенсорів, розміщених на дорогах. Впровадження SCOOT дозволяє зменшити затори на 15–30% та помітно покращити середню швидкість пересування транспорту. Система ефективно застосовується в таких великих містах, як Лондон, Манчестер та інших мегаполісах Європи.

Ще однією потужною системою адаптивного керування світлофорами є SCATS (Sydney Coordinated Adaptive Traffic System), створена у Австралії транспортною службою Transport for New South Wales. Її робота базується на зборі даних з індуктивних петель, розміщених на проїзній частині, що дозволяє в реальному часі оцінювати потік транспорту. На основі отриманої

інформації система централізовано змінює сигнали світлофорів. SCATS відома своєю високою масштабованістю та здатністю керувати до 1000 перехресть в межах єдиної мережі. Вона широко впроваджена у Сіднеї, Гонконгу, Торонто, а також у низці міст США.

У США розроблено іншу інноваційну систему — Surtrac, створену в університеті Carnegie Mellon. На відміну від централізованих рішень, ця система є децентралізованою та використовує алгоритми машинного навчання. Surtrac дозволяє кожному перехрестю самостійно аналізувати ситуацію та приймати рішення, що забезпечує оперативну реакцію без необхідності постійного зв'язку з центральним сервером. У тестуванні в місті Піттсбург було досягнуто зниження затримок транспорту до 40%, що свідчить про ефективність підходу.

Ще одне сучасне рішення — система InSync, яку розробила американська компанія Rhythm Engineering. Ця технологія ґрунтується на відеоаналітиці, тобто визначає транспортну ситуацію за допомогою камер та алгоритмів обробки зображення. InSync може прогнозувати потік транспорту й адаптувати світлофорні сигнали відповідно до очікуваного навантаження на перехрестя, що дозволяє забезпечити динамічну синхронізацію руху.

В українських містах системи керування світлофорами часто ще базуються на фіксованих таймерах або простих адаптивних алгоритмах. У Києві реалізовано проект "Інтелектуальна транспортна система" для головних магістралей. У Львові встановлюють адаптивні світлофори з датчиками руху. В Харкові ведуться пілотні проекти щодо впровадження інтелектуальних транспортних систем.

Сучасні комп'ютерні системи керування світлофорами демонструють високий рівень адаптивності та інтелекту, але потребують значних ресурсів для впровадження. В Україні ці технології поступово впроваджуються, що відкриває перспективи для створення власних рішень, адаптованих до місцевих умов та обмежень.

					КС КРБ 123.198.00.00 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3 Особливості, призначення і застосування комп'ютерного зору на регульованих перехрестях

Комп'ютерний зір (Computer Vision) – це технологія, що дозволяє комп'ютерним системам аналізувати зображення або відео в режимі реального часу, ідентифікуючи об'єкти та приймаючи відповідні рішення без участі людини. На регульованих перехрестях комп'ютерний зір використовується для інтелектуального керування дорожнім рухом, що дозволяє значно підвищити ефективність роботи світлофорів, знизити затори та підвищити безпеку учасників руху.

Основне призначення комп'ютерного зору у сфері керування перехрестями полягає у виявленні транспортних засобів, аналізі інтенсивності руху та передаванні інформації на блок керування світлофором. Це дозволяє реалізувати адаптивне керування світлофорними фазами. Система автоматично переключає сигнали світлофора в залежності від кількості автомобілів на кожній смузі. Такий підхід суттєво зменшує час простою транспорту та покращує пропускну здатність перехрестя.

Крім цього, комп'ютерний зір може бути використаний для:

- фіксації порушень правил дорожнього руху;
- визначення напрямку руху автомобілів;
- розпізнавання номерних знаків;
- моніторингу пішоходів та велосипедистів;
- збору статистичних даних для міського планування.

З технічного погляду, система комп'ютерного зору складається з камер, встановлених над або збоку від смуг руху, комп'ютерного модуля та програмного забезпечення для аналізу зображень. Найпоширенішим способом фіксації транспортних засобів є порівняння поточних кадрів з попередніми. Якщо зафіксовано значні зміни, це свідчить про появу автомобіля на смузі. Такий підхід є достатньо простим для реалізації на недорогому обладнанні, але

					КС КРБ 123.198.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

вимагає якісного налаштування чутливості до змін.

У більш складних системах використовуються алгоритми штучного інтелекту, які здатні не лише виявляти об'єкти, а й класифікувати їх за типом (легкові, вантажні автомобілі, автобуси), визначати швидкість руху та прогнозувати подальші дії. До цих систем можна віднести такі нейронні мережі як YOLO та MobileNet. Проте такі системи вимагають більшої обчислювальної потужності та набагато складніших апаратно-програмних рішень.

Системи комп'ютерного зору вже активно впроваджуються у містах для розбудови інтелектуальної транспортної інфраструктури. Наприклад, адаптивні світлофорні системи з виявленням транспорту на базі відеокамер застосовуються у рамках концепції Smart City в багатьох європейських країнах, США, а також в Україні. У простіших системах, таких як ті, що реалізуються на базі Raspberry Pi, машинний зір дозволяє створювати доступні локальні рішення, які не потребують дорогих сенсорів або складної серверної інфраструктури.

Одним із прикладів є використання декількох Raspberry Pi з камерами, кожна з яких слідкує за окремою смугою руху. Кожна камера фіксує зміну зображення та повідомляє центральний блок керування про появу автомобіля. У відповідь система визначає оптимальну комбінацію світлофорних сигналів, забезпечуючи ефективне та безпечне регулювання руху.

Комп'ютерний зір є ключовою складовою сучасних систем автоматизованого керування дорожнім рухом. Його впровадження дозволяє значно покращити ефективність функціонування перехресть, підвищити безпеку та створити основу для побудови інтелектуальних транспортних систем.

					<i>КС КРБ 123.198.00.00 ПЗ</i>	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

2.1 Проектування архітектури комп'ютерної системи керування світлофорами

У процесі створення комп'ютерної системи керування світлофорами, ключовим етапом є побудова її архітектури – логічної схеми, що відображає структуру, призначення елементів і способи взаємодії між ними. Архітектура дозволяє сформулювати уявлення про функціональні блоки та шляхи обміну даними відповідно до технічних вимог.

Апаратна частина системи включає наступні компоненти:

- мікрокомп'ютер Raspberry Pi 4 Model B;
- камера Raspberry Pi Camera Module 3;
- локальну мережу для обміну повідомленнями між пристроями;
- карту пам'яті microSD для кожного вузла;
- плата розширення Raspberry Pi Expansion Board Power Relay.

Комп'ютерна система керування світлофорами побудована на модульному принципі, що дозволяє забезпечити гнучкість, масштабованість і надійність у роботі. У системі передбачено кілька окремих, але взаємопов'язаних компонентів, кожен із яких виконує свою функцію, що в сукупності дозволяє здійснювати адаптивне керування дорожнім рухом на перехресті.

Ключову роль у системі відіграють локальні вузли збору даних. У даній конфігурації кожна смуга руху обладнана окремим мікрокомп'ютером Raspberry Pi 4 Model B, до якого під'єднана камера Raspberry Pi Camera Module 3. Ці вузли встановлюються на стовпах таким чином, щоб камера мала чіткий огляд лише однієї смуги. Основним завданням кожного такого вузла є

					КС КРБ 123.198.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Вовчак Р.А.			Проектна частина		
Перевір.		Шингера Н.Я.					
Реценз.		Бревус В.М.					
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркуші
						20	
					ТНТУ, каф. КС, гр. CI-41		

постійне створення знімків, аналіз зображення, виявлення змін і в разі виявлення, передачі конкретної команди. Для цього реалізується алгоритм комп'ютерного зору, який порівнює поточне зображення з попереднім і визначає наявність руху шляхом оцінки зміни пікселів [3].

Усі локальні вузли підключені до центрального контролера – ще одного Raspberry Pi, який виконує функції головного керівника системи. Саме він приймає сигнали від усіх детекторів, обробляє отриману інформацію та приймає рішення щодо перемикання сигналів світлофорів. Центральний блок виконує аналіз ситуації на перехресті та визначає, для якого напрямку слід увімкнути зелений сигнал, а де – червоний. Основним критерієм прийняття рішення є факт наявності чи відсутності автомобілів на конкретних смугах. В пріоритеті завжди має бути головна дорога, але в разі зatoryного потоку машин по головній дорозі і наявності транспортних засобів на другорядній, то після закінчення таймера, незалежно від того чи є автомобілі на головній дорозі, сигнал світлофора переключиться на зелений для другорядної дороги. Архітектура комп'ютерної системи продемонстрована на рисунку 2.1.

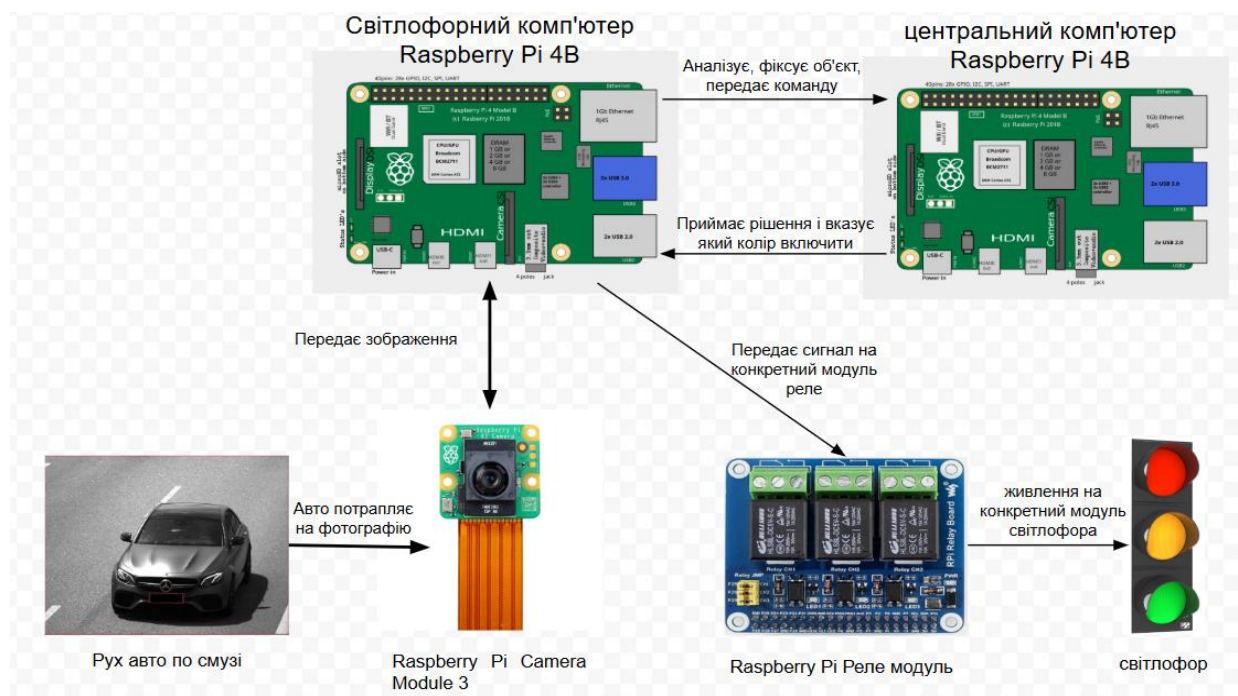


Рисунок 2.1 – Архітектура комп'ютерної системи керування світлофорами

Фізичне перемикання сигналів світлофора реалізується через плату розширення Raspberry Pi Expansion Board Power Relay, яка дозволяє переключати живлення світлофорних секцій. Кожен релейний вихід відповідає певному кольору світлофора, а перемикання виконується згідно з алгоритмом, що реалізований у центральному контролері.

Для обміну даними між вузлами використовується локальна комп'ютерна мережа Ethernet. Кожен локальний вузол надсилає повідомлення про виявлення транспорту до центрального контролера, де ця інформація накопичується, аналізується, а потім використовується для формування команди на перемикання світлофора і розсилання цієї команди на світлофорні контролери.

Функціональна логіка системи побудована на адаптивному підході. Якщо на певній смузі фіксується транспортний засіб, система перемикає сигнал у цьому напрямку, надаючи зелене світло. Якщо виявлено декілька активних смуг, система продовжує аналізувати смуги і паралельно запускає таймер. Якщо ситуація на смугах не зміниться і таймер закінчиться, тоді система примусово перемкне сигнал світлофора, щоб пропустити автомобілі які очікували.

2.2 Характеристики і структура Raspberry Pi

Одноплатні комп'ютери Raspberry Pi є популярними інструментами для створення прототипів, досліджень і навчання у сфері комп'ютерної інженерії. Не зважаючи на компактний розмір, Raspberry Pi 4 Model B забезпечує широкий спектр функцій, необхідних для розробки сучасних інтелектуальних систем [4]. Його легко підключити до телевізора, монітора, клавіатури чи інших периферійних пристроїв через стандартні порти HDMI та USB. Структуру і зовнішній вигляд продемонстровано на рисунку 2.2.

					КС КРБ 123.198.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

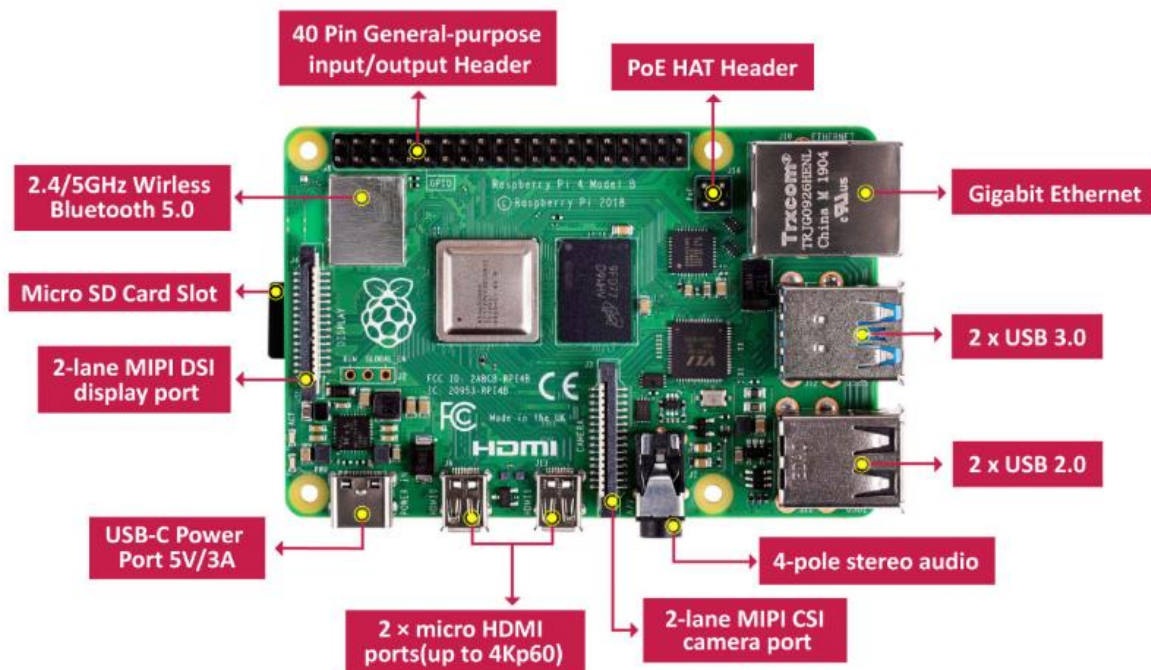


Рисунок 2.2 – Структура Raspberry Pi

Raspberry Pi 4 Model B відрізняється від попередніх версій не лише покращеними характеристиками, а й розширеним функціоналом. Він активно застосовується як у сфері освіти, так і при розробці вбудованих рішень. Зокрема, цей мінікомп'ютер широко використовується для обробки текстових та графічних даних, програмування і тестування різних алгоритмів, створення інтелектуальних систем обробки відео та зображень, підключення до мережі Інтернет і розгортання мережесервісів, розробки інженерних проєктів із сенсорами, камерами та іншими модулями. Було обрано саме цю модель, оскільки її потужності цілком вистачає для обчислень які використовуються в даній системі, і крім цього вона має дуже широкий вибір компонентів, які були розроблені саме під неї [5].

Технічні характеристики Raspberry Pi 4 Model B значно перевершують попередні моделі. У таблиці 2.1 наведено ключові параметри різних варіантів моделей та версій.

Таблиця 2.1 – Порівняльні характеристики моделей Raspberry Pi

Модель	SoC	Тактова частота, МГц	RAM ГБ	Технології Ethernet	Wireless	Blue tooth
Raspberry Pi Model A+	BCM 2835	700	0.5	-	-	-
Raspberry Pi Model B+	BCM 2835	700	0.5	100Base-T	-	-
Raspberry Pi 2 Model B	BCM 2836	900	1	100Base-T	-	-
Raspberry Pi Zero W	BCM 2835	1000	0.5	-	802.11ac	4.1
Raspberry Pi 3 Model B	BCM 2837	1200	1	100Base-T	802.11ac	4.1
Raspberry Pi 3 Model A+	BCM 2837	1400	0.5	-	802.11ac	4.2
Raspberry Pi 3 Model B+	BCM 2837	1400	1	1000Base-T	802.11ac	4.2
Raspberry Pi 4 Model B	BCM2711	1500	2/4/8	1000Base-T	802.11ac	5.0
Raspberry Pi 5	BCM2712	2400	2/4/8/16	1000Base-T	802.11ac	5.0

Особливістю плати є наявність інтерфейсу GPIO (General Purpose Input/Output), що дозволяє здійснювати фізичне з'єднання із зовнішніми пристроями, сенсорами або виконавчими механізмами. Контакти GPIO здатні виконувати різні функції, які визначаються програмно. Зокрема, до основних можливостей GPIO належать:

- подача живлення на зовнішні компоненти;
- з'єднання із загальним проводом (GND);
- передача або прийом цифрових сигналів керування.

Інтерфейс GPIO зображено на рисунку 2.3.

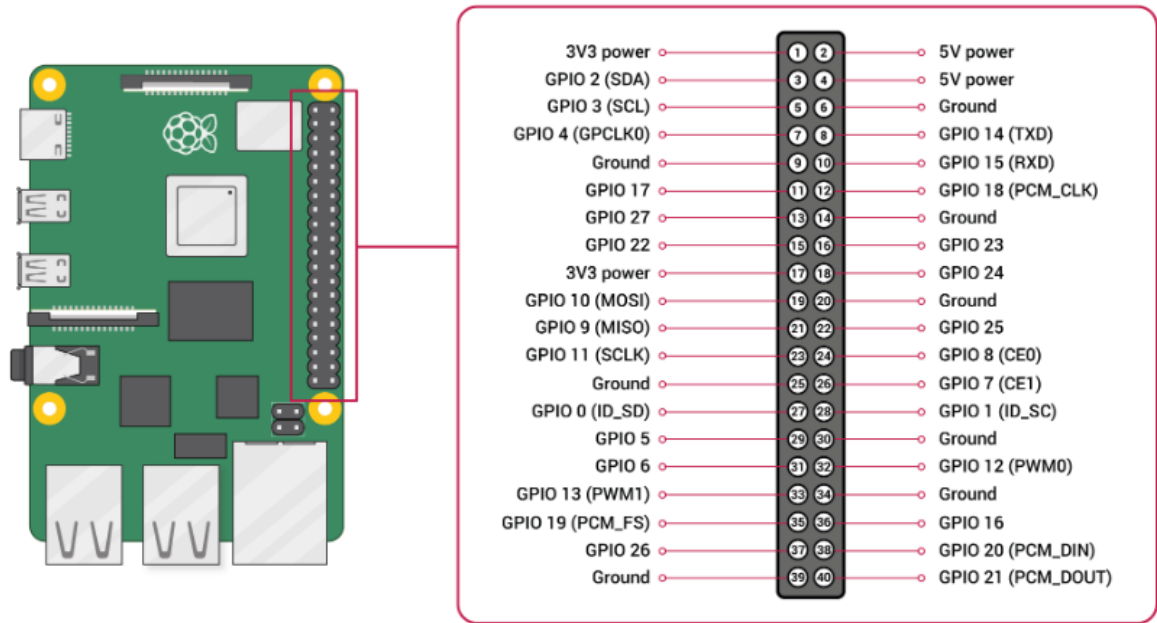
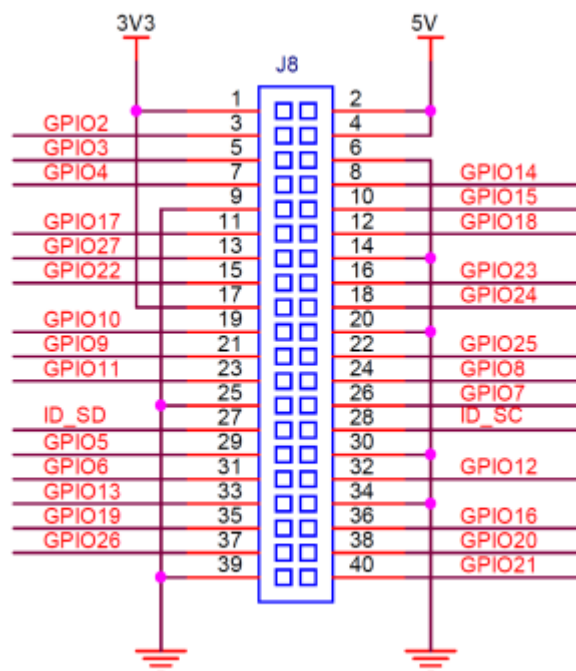


Рисунок 2.3 – GPIO в Raspberry Pi 4 Model B

Конфігурація кожного з контактів визначається програмно, що дозволяє використовувати один і той самий пін як для входу, так і для виходу. Проте, під час проєктування необхідно уникати перевантаження, оскільки допустимий струм для GPIO не повинен перевищувати 50 мА. На рисунку 2.4 зображено схему електричну принципову пінів GPIO.



ID_SD and ID_SC PINS:

These pins are reserved for HAT ID EEPROM.

At boot time this I2C interface will be interrogated to look for an EEPROM that identifies the attached board and allows automatic setup of the GPIOs (and optionally, Linux drivers).

DO NOT USE these pins for anything other than attaching an I2C ID EEPROM. Leave unconnected if ID EEPROM not required.

Рисунок 2.4 – Схема електрична принципова GPIO

Особливу увагу слід звернути на пари контактів GPIO-0 (ID_SD) та GPIO-1 (ID_SC), які зазвичай зарезервовані для підключення додаткових модулів розширення і мають протокол передачі даних I2C. Їх можна побачити під номерами контактів 27 і 28 на рисунку.

Керування GPIO реалізується на програмному рівні. Найбільш поширеною мовою програмування є Python, завдяки великій кількості бібліотек, зрозумілому синтаксису та активній спільноті. Водночас, можлива підтримка й інших мов, таких як C/C++, JavaScript, або PHP, що дозволяє гнучко адаптувати систему під конкретні завдання.

До основних апаратних характеристик Raspberry Pi 4 Model B належать:

- процесор: 64-бітний 4-ядерний Broadcom BCM2711 (Cortex-A72) з тактовою частотою 1.5 ГГц;
- оперативна пам'ять: 2/4/8 ГБ LPDDR4;
- графічне ядро: VideoCore VI;
- бездротові модулі: Wi-Fi 802.11ac, Bluetooth 5.0;
- мережа: Gigabit Ethernet;

- порти USB: 2 х USB 3.0, 2 х USB 2.0;
- 2 micro HDMI порти з підтримкою 4K відео;
- GPIO: 40-контактний роз'єм;
- підтримка камер та дисплеїв через порти CSI та DSI;
- слот для microSD – використовується як системний накопичувач;
- живлення: через USB-C з підтримкою 5В/3А.

Враховуючи ці характеристики, Raspberry Pi 4 Model B цілком підходить для побудови комп'ютерної системи керування світлофорами. Потужність даного мікрокомп'ютера достатня для ефективного обчислення.

2.3 Проєктування схеми та обґрунтування вибору апаратної частини комп'ютерної системи

Для розробки схеми комп'ютерної системи керування світлофорами з використанням модульних камер Raspberry Pi, потрібне таке апаратне забезпечення:

- Raspberry Pi 4 Model B;
- Raspberry Pi Camera Module 3;
- Raspberry Pi Expansion Board, Power Relay;
- світлодіодний модуль світлофора;
- Ethernet комутатор TP-Link LS1005G;
- перетворювач струму AC-DC з змінного 220В в постійний 24В;
- перетворювач струму AC-DC з змінного 220В в постійний 5В;
- перетворювач струму DC-DC з постійного 24В в постійний 5В.

В процесі розробки було спроектовано три варіанти функціональної схеми комп'ютерної системи керування світлофорами. Перший варіант передбачав центральний блок реле-перемикачів, який мав розміщуватися безпосередньо біля центрального контроллера (рисунок 2.5).

					<i>КС КРБ 123.198.00.00 ПЗ</i>	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

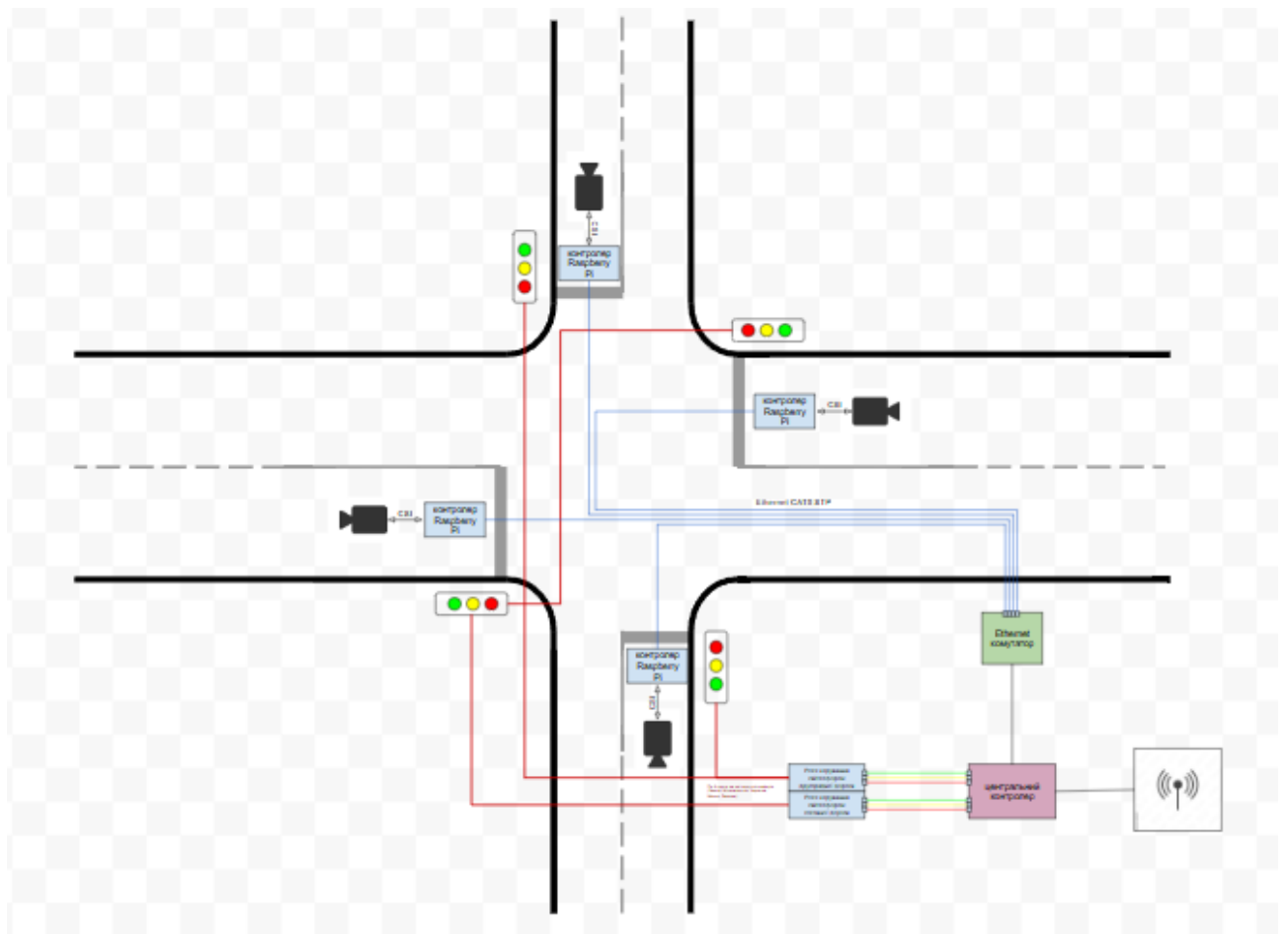
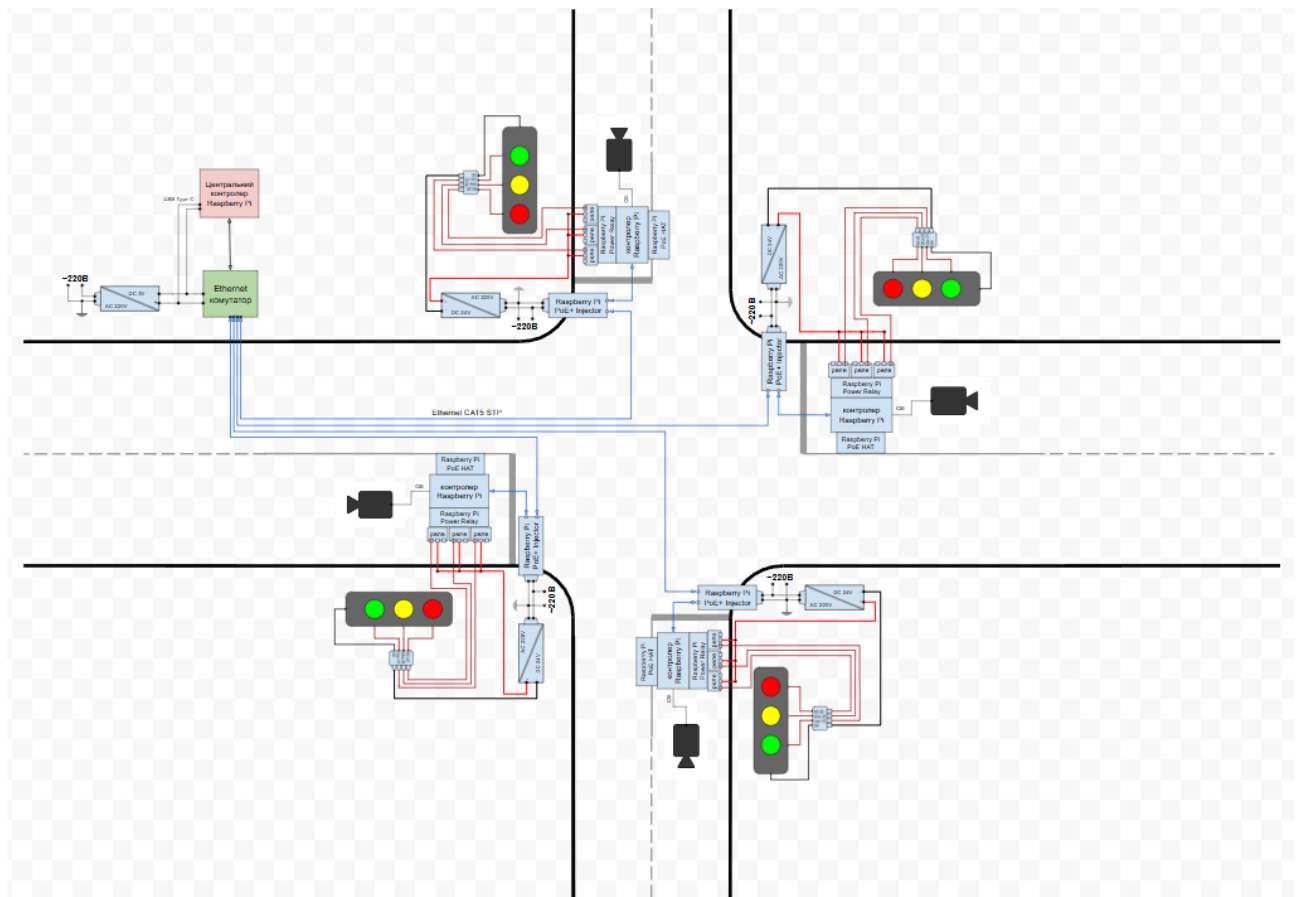


Рисунок 2.5 – Перший варіант функціональної схеми комп'ютерної системи керування світлофором

Цей варіант дає можливість використовувати меншу кількість реле, шляхом підключення світлофорів по діагоналі. Але через завелику довжину силових провідників, що спричиняє збільшення опору, розсіювання енергії у вигляді тепла і можливе падіння напруги, що призведе до зниження ефективності роботи приладів. Тому було прийнято рішення змінити схему, щоб зменшити довжину силових кабелів.

Оскільки в даній системі використовується Ethernet кабель для з'єднання між собою контролерів, було розроблено другий варіант, в якому інформація після обробки в центральному контролері, передає конкретну команду назад до контролерів розташованих з камерами біля світлофорів.

Ethernet кабель може передавати живлення (технологія Power over Ethernet (PoE)), тому було доставлено PoE інжектори для живлення мікрокомпютера. Raspberry Pi 4B не підтримує технологію PoE в чистому вигляді, але це можна змінити за рахунок приєднання плати розширення Raspberry Pi Power over Ethernet HAT. (рисунок 2.6)



керування світлофором

					КС КРБ 123.198.00.00 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Проаналізувавши другий варіант схеми було виявлено, що використання технології PoE в даній системі не вигідне. Потрібно встановлювати на кожен модуль PoE інжектор і плату розширення для підключення живлення до Raspberry Pi, тому що компоненти даної технології які були використані в цій схемі мають обмежену пропускну здатність сили струму, яка недостатня для повноцінної роботи системи. Тому було прийнято рішення прибрати модулі, які в даному випадку є зайвими (PoE інжектори і плату розширення PoE NAT), і переробити живлення на перетворювач напруги на 5 вольт постійного струму, і живити мікрокомп'ютери підключивши їх кабелем через роз'єм USB Type C. Третя версія схеми зображена на рисунку 2.7.

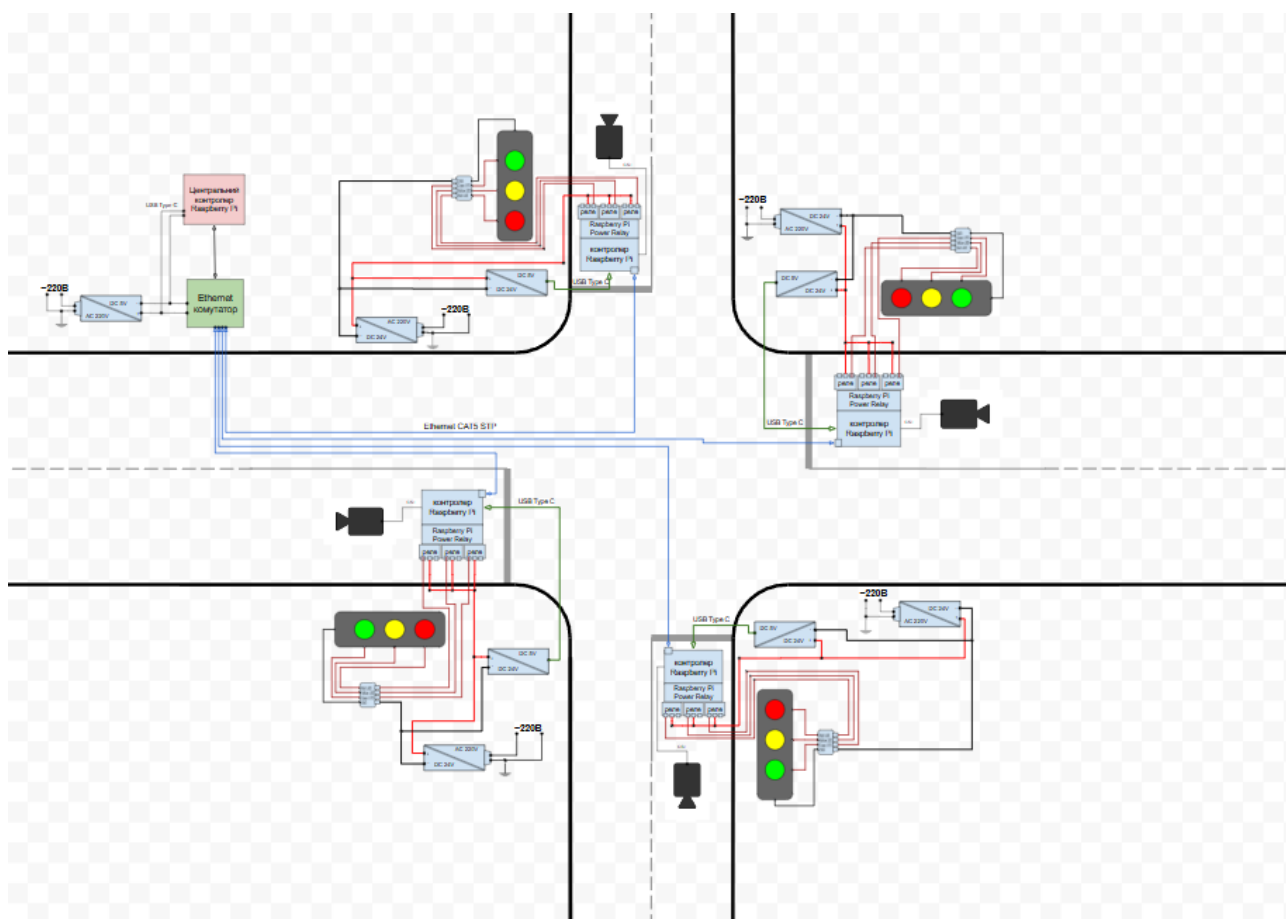


Рисунок 2.7 – Третій варіант функціональної схеми комп'ютерної системи керування світлофором

Підключення модуля камери до мікрокомп'ютера відбувається за рахунок шлейфа на 15 контактів, який йде в комплекті з камерою і підключається в інтерфейс CSI. Довжина шини складає 200мм, а оскільки камера розміщується поряд з мікрокомп'ютером, цього цілком достатньо.

Для живлення плат які розміщені біля камер було обрано варіант живлення через перетворювач напруги. Підключається він до перетворювача напруги на 24В постійного струму, що живить світлофор. До мікрокомп'ютера підключається кабелем USB Type C.

Ще однією платою розширення Raspberry Pi в цій кваліфікаційній роботі є XYGStudy Raspberry Pi Expansion Board Power Relay Module – це модуль з вбудованими реле, який в даному випадку використовується для керування подачею електроенергії на модулі світлодіодів на світлофорі. Підключення відбувається таким самим шляхом як і в інших плат розширення.

Щоб заживити модуль світлофора потрібно постійний струм з напругою 24 вольт, тому для цього потрібно перетворити змінний струм 220 вольт. Це можна зробити за допомогою перетворювача. З характеристик світлофора відомо що червоний модуль споживає 7W (ватт), жовтий – 7W, і зелений 6W. Для розрахунку потрібного струму в амперах, використовується перетворений вигляд формули закону Ома (1.1) де I - сила струму (в амперах), P - потужність (у ватах), U - напруга (у вольтах).

$$I = P / U . \quad (1.1)$$

де I – сила струму (в амперах);

P - потужність (у ватах);

U - напруга (у вольтах).

Щоб вирахувати максимально можливий струм, треба взяти до прикладу, що всі три секції світлофора включені, тобто споживання потужності 20W. Після чого підставити значення і після обчислення, обрати потрібний блок живлення.

					КС КРБ 123.198.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

Для підключення живлення до центрального контролера Raspberry Pi і Ethernet комутатора, потрібно блок живлення для перетворення 220В змінного струму в постійний струм 5В. Живлення до контролера підключається за допомогою кабеля USB Type C, а до комутатора за рахунок комплектного роз'єму.

2.3.1 Модуль камери Raspberry Pi

Для кваліфікаційної роботи було обрано Raspberry Pi Camera Module 3, оскільки це офіційний модуль камери від Raspberry Pi Foundation, який використовується для захоплення зображень та відео. Це вже третє покоління камер, що має суттєві покращення у порівнянні з попередніми версіями. Даний модуль оснащений сенсором Sony IMX708 з роздільною здатністю 11,9 мегапікселів і автофокусуванням з фазовим виявленням (PDAF) для швидкого автофокусування. Доступний в стандартному і ширококутному варіанті, обидва з яких можуть бути з інфрачервоним фільтром або без нього [6].

Цей модуль, завдяки сенсору Sony IMX708, здатний робити статичні знімки з роздільною здатністю 4608 на 2592 пікселів і видавати в форматі RAW10, а також знімати відео з максимальною роздільною здатністю 1080 p50. Крім цього можна змінити якість відео на 720 p100 або 480 p120.

Габаритні розміри камери складають 24 x 25 x 11.5мм, що дозволяє використовувати її в системах обмежених розміром. Підключення до плати відбувається за рахунок вбудованого спеціального порта CSI. Він розміщений на верхній частині плати, між портом мікро HDMI та аудіо роз'ємом. Його розташування обведено на рисунку 2.8.

					<i>КС КРБ 123.198.00.00 ПЗ</i>	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

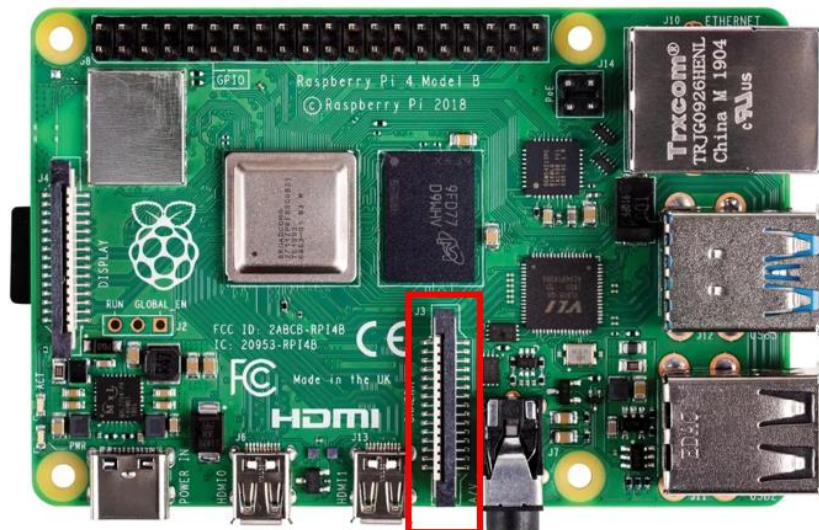


Рисунок 2.8 – Інтерфейс підключення модуля камери RPi

CSI – це апаратний інтерфейс який використовується для підключення камер до мікрокомп'ютерів, зокрема Raspberry Pi. Він забезпечує високу швидкість передачі відео та зображень без використання USB або HDMI. Опис виводів і їх призначення представлено в таблиці 2.2.

Таблиця 2.2 – Опис виводів CSI і їх призначення

№ контакту	Назва	Опис
1	GND	Ground
2	CAM_D0_N	MIPI Data Lane 0 Negative
3	CAM_D0_P	MIPI Data Lane 0 Positive
4	GND	Ground
5	CAM_D1_N	MIPI Data Lane 1 Negative
6	CAM_D1_P	MIPI Data Lane 1 Positive
7	GND	Ground
8	CAM_CLK_N	MIPI Clock Lane Negative
9	CAM_CLK_P	MIPI Clock Lane Positive

№ контакту	Назва	Опис
10	GND	Ground
11	CAM_IO0	Power Enable
12	CAM_IO1	LED Indicator
13	CAM_SCL	I2C SCL (Serial Clock Lane)
14	CAM_SDA	I2C SDA (Serial Data Lane)
15	CAM_3V3	3.3V Power Input

Даний модуль камери третьої версії представлений в чотирьох варіантах. Зовнішній вигляд цих камер показано на рисунку 2.9.



Рисунок 2.9 – Зовнішній вигляд Raspberry Pi Camera Module v3

Порівняльні характеристики чотирьох можливих варіантів вказано в таблиці 2.3.

Таблиця 2.3 – Характеристики модулів камер Raspberry Pi Camera Module v3

	Camera Module 3	Camera Module 3 NoIR	Camera Module 3 Wide	Camera Module 3 Wide NoIR
Діапазон фокусування	10см – ∞	10см – ∞	5см – ∞	5см – ∞
Фокусна відстань	4.74мм	4.74мм	2.75мм	2.75мм
Діагональне поле зору	75°	75°	120°	120°
Горизонтальне поле зору	66°	66°	102°	102°
Вертикальне поле зору	41°	41°	67°	67°
Фокусне відношення	F1.8	F1.8	F2.2	F2.2
Чутливість до інфрачервоного випромінювання	Ні	Так	Ні	Так

Для кваліфікаційної роботи було обрано Camera Module 3, оскільки для роботи комп'ютерної системи потрібно невеликий кут поля зору, щоб камера могла фіксувати зображення тільки однієї смуги руху на дорозі, і щоб в кадр не потрапляли сторонні об'єкти з узбіччя або з зустрічної смуги.

Порядок підключення модуля камери Raspberry Pi:

- вимкнути Raspberry Pi і відключити живлення;
- відкрити затискач на роземі CSI;
- підключити 15-ти піновий шлейф від камери;
- перевірити чи рівно і правильно встановлений шлейф;
- закрити затискач і переконатися чи зайшов він до кінця.

Схему електричну принципову модуля камери Raspberry Pi подано на рисунку 2.10.

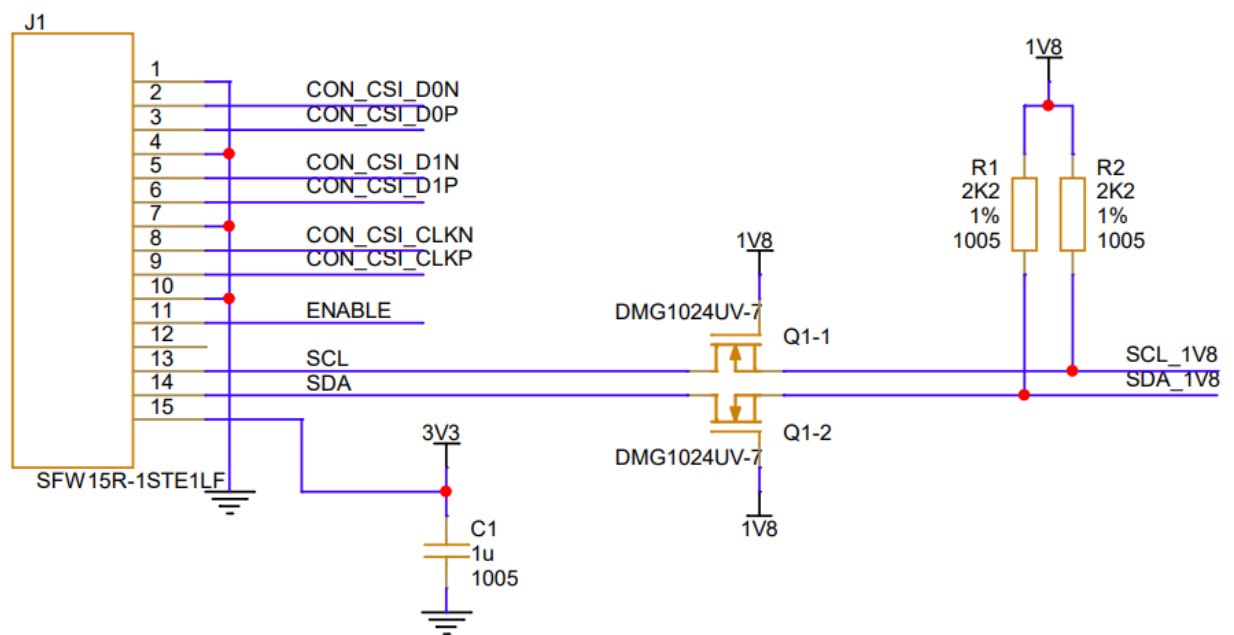


Рисунок 2.10 – Схема електрична принципова модуля камери

Після підключення модуля камери Raspberry Pi Camera Module 3 його можна задіяти в роботі системи комп'ютерного зору різними способами. Найпростіший варіант передбачає використання утиліт на основі бібліотеки libcamera, або picamera2 якщо програма написана на Python. Дані утиліти офіційно підтримуються в нових збірках операційної системи Raspberry Pi OS. Варто зазначити, що саме Camera Module 3 повною мірою орієнтований на роботу з libcamera, оскільки попередні інструменти, як-от raspistill, вже не підтримуються для цієї моделі.

Використання libcamera відкриває широкі можливості як для базової роботи з камерою, так і для складніших сценаріїв. Серед ключових переваг – регулярні оновлення від команди Raspberry Pi, активна участь сторонніх розробників у розширенні функціоналу, а також спрощений процес підключення нових камер, включно з модулями серії HQ та IMX708, які використано в Camera Module 3. Камера автоматично розпізнається системою, а користувач отримує доступ до налаштувань експозиції, фокусування, балансу білого та інших параметрів.

Завдяки вдосконаленому сенсору та автофокусній системі, Camera Module 3 чудово підходить для задач виявлення руху або об’єктів на відстані. Libcamera дозволяє програмно змінювати режим зйомки, регулювати роздільну здатність, налаштовувати частоту кадрів тощо. Оскільки модуль підтримує формат RAW10, а також передає дані через CSI-інтерфейс із високою пропускнуою здатністю, його зручно використовувати у високоточних системах виявлення транспорту.

Ще однією суттєвою перевагою є можливість інтеграції з популярними бібліотеками комп’ютерного зору, зокрема OpenCV та TensorFlow Lite, що дозволяє обробляти відеопотік у реальному часі. Це важливо для систем розпізнавання на транспортних перехрестях, де рішення про зміну сигналу світлофора має прийматись негайно після фіксації транспортного засобу.

Крім цього, libcamera забезпечує сумісність із 64-бітною операційною системою Raspberry Pi OS, що важливо для більш вимогливих застосувань, пов’язаних із великим обсягом даних та складними обчисленнями [7]. Повна підтримка автофокуса, HDR-зйомки та нічного режиму дозволяє використовувати камеру як удень, так і в умовах низького освітлення без суттєвої втрати якості зображення.

					КС КРБ 123.198.00.00 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

2.3.2 Raspberry Pi Expansion Board Power Relay

Raspberry Pi Expansion Board Power Relay – це плата розширення, з реле перемикачами, розроблена спеціально для мікрокомп'ютера Raspberry Pi (рисунок 2.11).

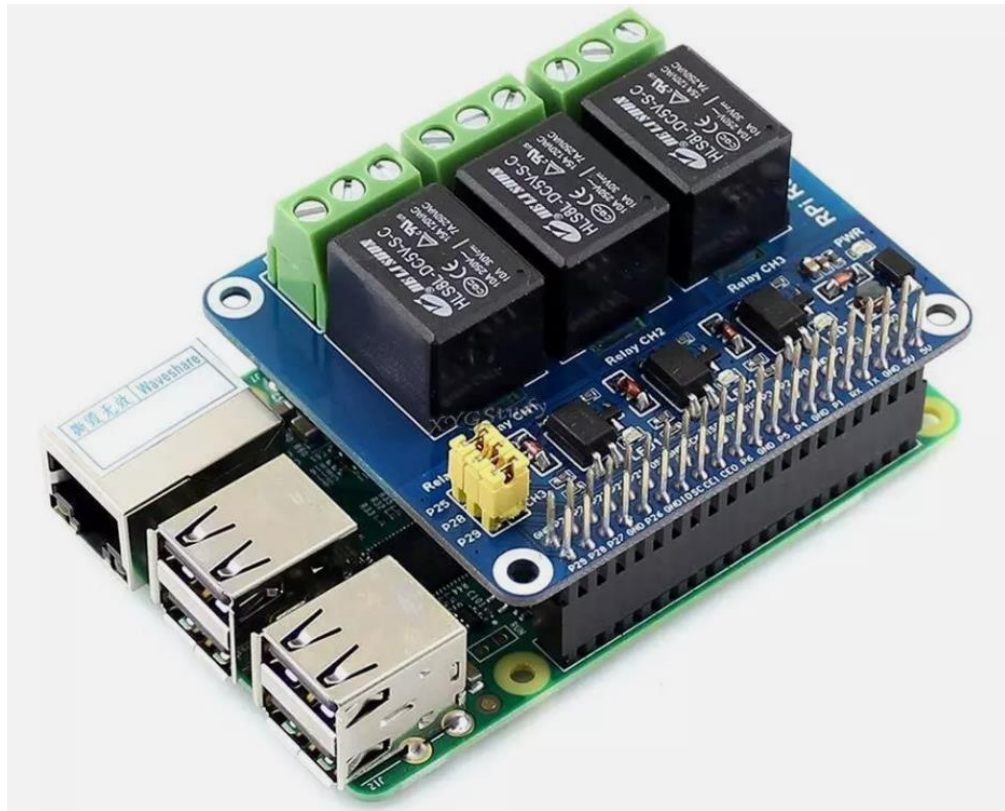


Рисунок 2.11 – Зовнішній вигляд плати Raspberry Pi Expansion Board Power Relay встановленої на мікрокомп'ютер Raspberry Pi

Як видно на рисунку 2.11, плата легко встановлюється на Raspberry Pi за рахунок наявності ідентичного інтерфейсу контактів GPIO. Також варто звернути увагу що наявні дублюючі виходи GPIO, що дозволяє приєднати інші плати розширення, які розроблені для даного мікрокомп'ютера [8].

Реле – це електро cơханічний пристрій, який дозволяє замикати або розмикати одне електричне коло, за допомогою сигналу з іншого електричного кола.

Використовується реле, коли потрібно пропускати великий струм або напругу за допомогою малопотужного сигналу. Плата підтримується версіями Raspberry Pi A+ / B+ / 2B / 3B / 3B+ та 4B.

Опис компонентів на платі пронумеровано на рисунку 2.12.

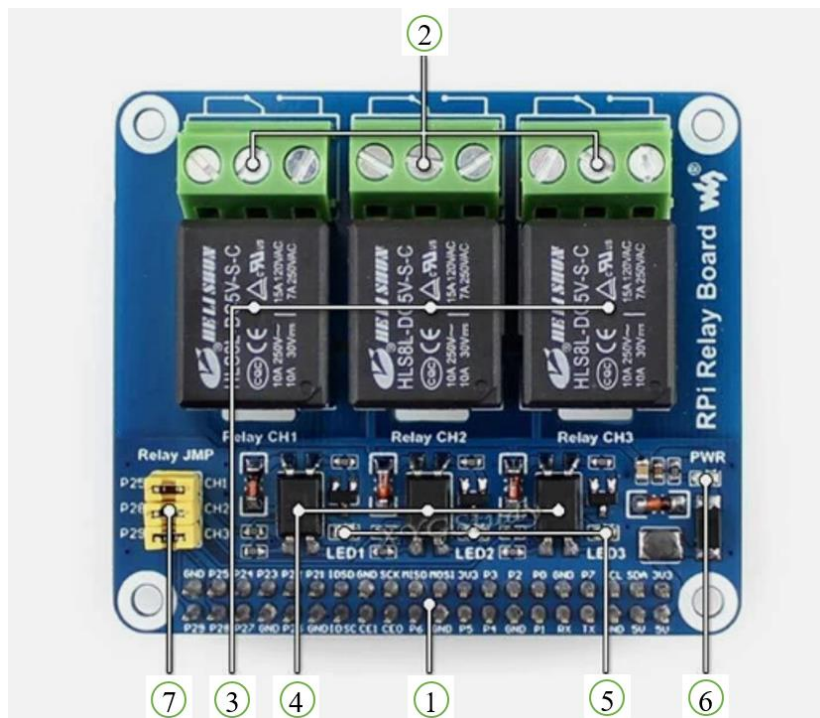


Рисунок 2.12 – Нумерація компонентів реле модуля

Опис компонентів:

- 1) Raspberry Pi GPIO інтерфейс;
- 2) клемний затискач для підключення приладів до реле;
- 3) реле HLS8L-DC5V;
- 4) оптопара PC817;
- 5) LED індикатори поточного статусу Реле;
- 6) LED індикатор живлення;
- 7) реле джампер.

Для передачі сигналу на плату розширення, за замовчуванням використовуються GPIO виходи під номерами 37, 38 і 40 (Таблиця 2.4). Але це можна змінити, витягнувши джампер і підключивши провідниками до потрібних виходів.

Таблиця 2.4 – Використання контактів платою розширення Relay Module

Chanel #	RPi #	wiringPi	BCM	Description
CH1	37	P25	26	Channel 1
CH2	38	P28	20	Channel 2
CH3	40	P29	21	Channel 3

Для реалізації керування зовнішніми пристроями було обрано саме цю плату розширення Raspberry Pi Expansion Board Power Relay, оскільки вона забезпечує повну апаратну сумісність з Raspberry Pi, має зручне підключення через GPIO і містить три незалежні релейні канали. Наявність дублюючих виходів GPIO забезпечує можливість підключення додаткових модулів без обмеження функціональності системи. Це забезпечує модульність і в подальшому дозволяє без значних змін додавати інші компоненти до системи.

2.3.3 LED Traffic Light

В якості світлофорного модуля для даної роботи було обрано готовий варіант від компанії Stack Light. Це трьохсекційний модуль світлофора оснащений світлодіодними лампами. Він простий в монтажі, має низьке споживання електроенергії і довгий термін роботи.

Підключення відбувається шляхом під'єднання провідників до клемної колодки, яка схована в корпусі світлофора. Має три окремі входи для подачі живлення на кожен сигнал світлофора, і четвертий вхід для підключення «землі», що дозволяє легко підключити до даної системи, оскільки живлення подається за допомогою трьох реле.

Може житися від 110 вольт змінного струму або від 24 вольт постійного. Схема підключення і розміри світлофора представлено на рисунку 2.13.

					КС КРБ 123.198.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

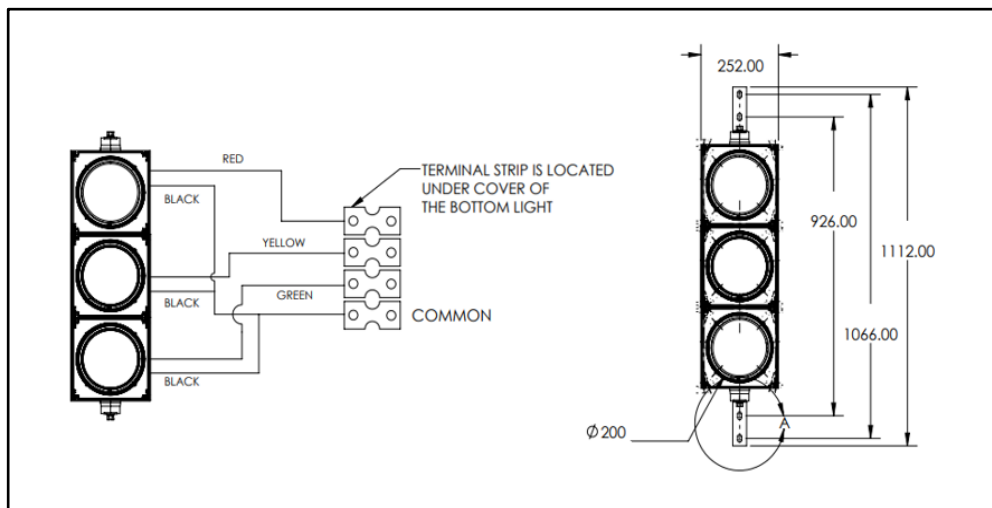


Рисунок 2.13 – Схема підключення і розміри світлофора

2.3.4 Перетворювач напруги

Перетворювачі напруги потрібні в системі керування світлофорами, щоб перетворити напругу з 220 Вольт змінного струму на постійний потрібної напруги і заживити окремі елементи системи.

Для підключення світлофора до електромережі знадобиться перетворювач напруги AC-DC, який буде видавати 24 Вольти постійного струму. Як вже було раніше обраховано, максимальне споживання світлофора не перевищує 0.83 Ампера, тому важливо щоб блок живлення видавав силу струму не менше 1 Амп. тільки для світлофора, не враховуючи іншої підключеної електроніки. Негативний контакт з перетворювача підключається до світлофора а позитивний до кожного реле, а вже звідти безпосередньо на кольори світлофора.

Живлення центральної плати Raspberry Pi і комутатора забезпечується перетворювачем напруги AC-DC, який перетворює 220 Вольт змінного струму, на 5 Вольт постійного.

Щоб заживити плати Raspberry Pi які знаходяться біля кожного світлофора, використовується перетворювач напруги DC-DC, який бере живлення від вищезгаданого перетворювача для світлофора, і видає 5 Вольт. Живлення до Raspberry Pi подається за рахунок кабеля USB Type C.

РОЗДІЛ 3 ПРОГРАМНА ЧАСТИНА

3.1 Розробка алгоритму

Перед створенням програмної частини системи керування світлофорами доцільно здійснити логічну побудову дій окремих її компонентів та їхньої взаємодії в єдиній архітектурі.

Перший етап полягає в активації всіх вузлів системи. Кожна Raspberry Pi, розміщена над окремою смугою руху, виконує перевірку доступності камери та мережевого з'єднання з центральним модулем [8]. Після запуску всі камери розпочинають цикл обробки зображення з метою виявлення наявності транспортного засобу в зоні контролю.

Коли камера фіксує суттєву зміну у зображенні, відповідна Raspberry Pi передає повідомлення центральному блоку за допомогою UDP-протоколу. Якщо змін не зафіксовано, пристрій переходить у режим очікування на заданий інтервал часу, після чого перевірка повторюється.

У центральному контролері, який також реалізовано на базі Raspberry Pi, збираються сигнали з усіх чотирьох вузлів. Відповідно до отриманої інформації, визначається пріоритетна дорога: головна має перевагу, однак у випадку її вільності та присутності транспорту на другорядній — надається дозвіл проїзду другорядній смузі через 60 секунд.

Подальші дії залежать від поточного стану світлофорів. Якщо необхідно змінити фазу, виконується триетапний процес перемикання: спочатку миготить зелений (3 с із інтервалом 0,5 с), потім вмикається жовтий (3 с), і лише після цього — червоний. У зворотному напрямку перехід від червоного до зеленого також супроводжується жовтим сигналом (3 с).

					КС КРБ 123.198.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Вовчак Р.А.			Програмна частина		
Перевір.		Шингера Н.Я.					
Реценз.		Бревус В.М.					
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркушів
						42	
					ТНТУ, каф. КС, гр. СІ-41		

Команди на включення або вимкнення відповідних реле передаються від центрального модуля через TCP-протокол до кожного Raspberry Pi, підключеного до релейного модуля [9]. Таким чином, реле 1 активує зелений сигнал, реле 2 — жовтий, реле 3 — червоний.

Функціональна структура системи побудована з урахуванням розділення логіки на три окремі програмні компоненти.

Перша програма працює на кожній з чотирьох Raspberry Pi із камерами та виконує безперервний аналіз зображення для виявлення транспортних засобів і передачі результату центральному модулю. Алгоритм програми зображено на рисунку 3.1.



Рисунок 3.1 – Алгоритм роботи програми виявлення транспорту.

Друга програма розміщена на центральному вузлі та відповідає за прийом даних із камер, обробку ситуації на перехресті та формування рішень щодо перемикавання фаз світлофорів. Блок-схема зображена на рисунку 3.2.

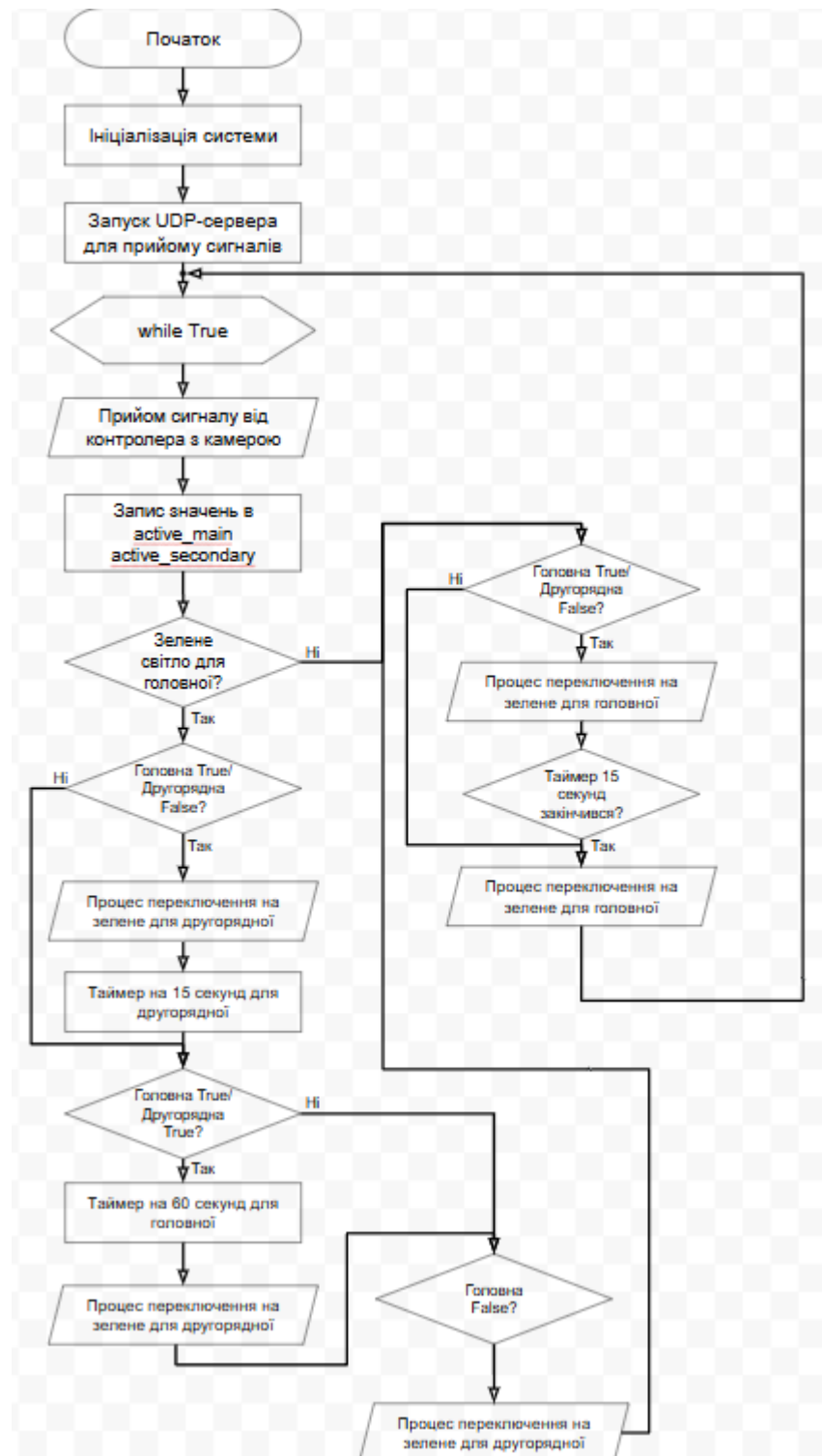


Рисунок 3.2 – Алгоритм роботи програми центрального контролера

Третя програма працює на кожній з чотирьох Raspberry Pi, які керують релейними модулями. Вона приймає команди з центрального контролера по TCP і здійснює увімкнення чи вимкнення потрібних реле відповідно до заданої логіки. Алгоритм роботи на рисунку 3.3.

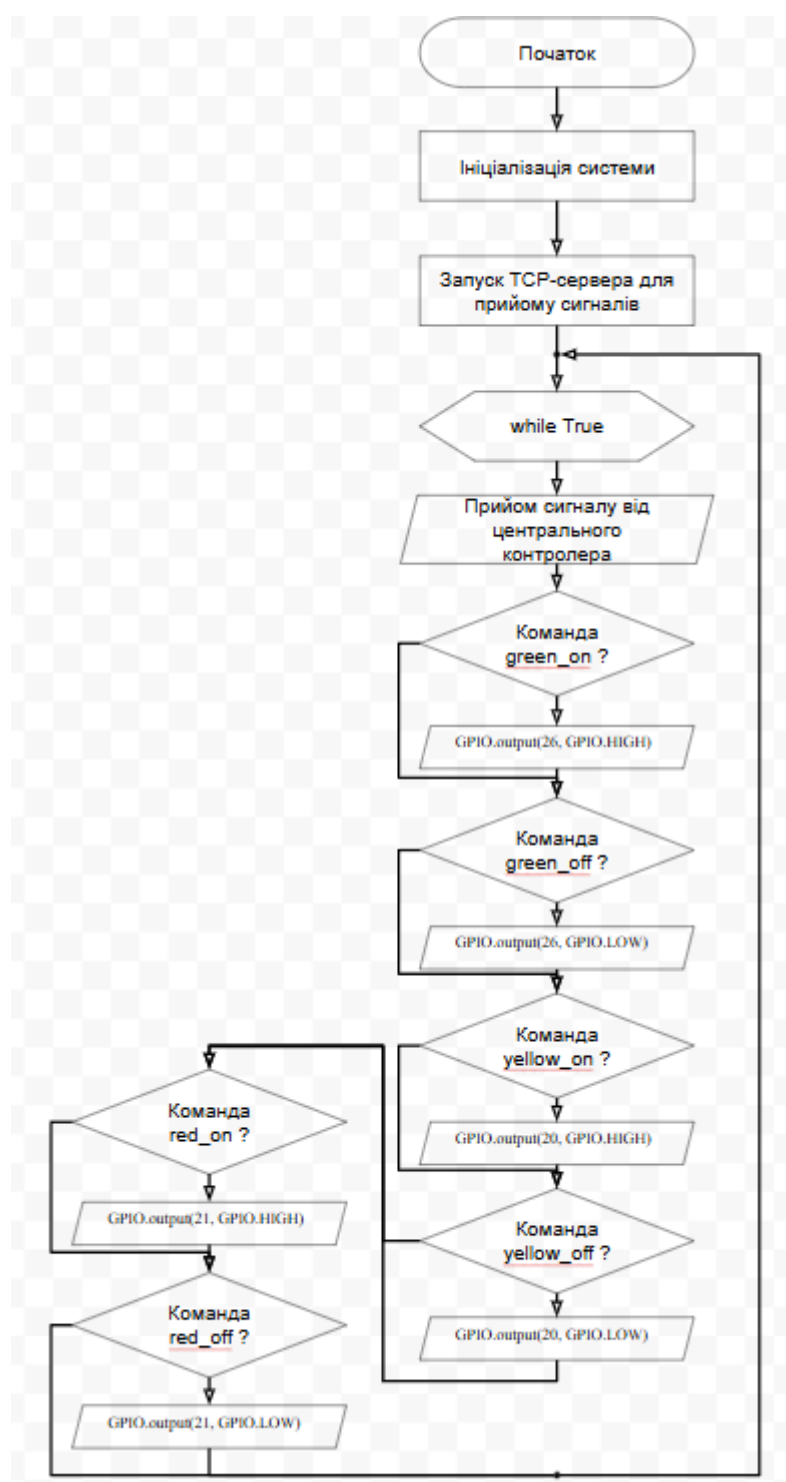


Рисунок 3.3 – Алгоритм роботи програми перемикування реле.

Після завершення одного циклу система повертається на початок, очікуючи нові вхідні дані з камер. Цей процес відбувається безперервно протягом усього часу роботи системи.

Реалізоване поєднання трьох програмних компонентів – виявлення руху, центрального контролера та керування реле світлофорів – забезпечує побудову узгодженої, ефективної та масштабованої системи керування дорожнім рухом [11]. Така модульна структура дозволяє кожному вузлу виконувати свою чітку роль: локальні Raspberry Pi з камерами самостійно фіксують появу транспорту, центральний контролер приймає рішення на основі отриманих даних, а керування світлофорами реалізується через простий набір TCP-команд.

Таке модульне розділення забезпечує гнучкість у розробці, налагодженні та масштабуванні системи, дозволяючи розподіляти навантаження між пристроями та полегшувати пошук і усунення можливих помилок у роботі.

3.2 Розробка програмного забезпечення для виявлення руху по смугах

Першим етапом розробки програмного забезпечення є створення коду для Raspberry Pi ,які забезпечують виявлення транспортних засобів за допомогою підключених камер, написаного на мові Python [10]. У системі використано чотири такі пристрої — по одному на кожну смугу перехрестя. Логіка їхньої роботи однакова, відмінність полягає лише у типі дороги: головна або другорядна.

Програмна реалізація ґрунтується на аналізі змін між кадрами. Для цього використовується обробка зображення: з камери періодично зчитується кадр, який порівнюється з попереднім. Якщо відмінності суттєві – це означає що є транспорт на смузі. У цьому випадку на центральний контролер надсилається UDP-пакет із ідентифікатором смуги, на якій зафіксовано автомобіль.

					КС КРБ 123.198.00.00 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

Для роботи з Raspberry Pi Camera Module 3 використовується бібліотека *picamera2*. Після підключення камери через CSI-порт необхідно активувати інтерфейс у конфігурації Raspberry Pi та встановити відповідні бібліотеки.

За ініціалізацію камери в програмі відповідає лістинг показаний на рисунку 3.4.

```
1 from picamera2 import Picamera2
2 import time
3
4 picam2 = Picamera2()
5 picam2.start()
6 time.sleep(2)
```

Рисунок 3.4 – Лістинг ініціалізації і прогрів камери

Для подальшої роботи з зображеннями підключається бібліотека *cv2* (OpenCV) – це бібліотека з відкритим кодом призначена для комп'ютерного зору. Завдяки цій бібліотеці можна обробляти і зберігати зображення, аналізувати відео і виявляти об'єкти.

Після запуску камера зчитує кадри у заданому форматі, зменшуються розміри зображення для пришвидшення обробки. Також в даному випадку потрібне перетворення у відтінки сірого та гаусове розмиття (Рисунок 3.5).

```
1 import cv2
2
3 frame = picam2.capture_array()
4 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
5 blurred = cv2.GaussianBlur(gray, (21, 21), 0)
```

Рисунок 3.5 – Лістинг попередньої обробки зображення

Програма зберігає попередній кадр і порівнює його з новим [12]. Використовується побудова різниці, бінаризація, пошук контурів — якщо контури великі, рух вважається наявним (Рисунок 3.6).

```

delta = cv2.absdiff(prev_frame, blurred)
thresh = cv2.threshold(delta, 25, 255, cv2.THRESH_BINARY)[1]
contours, _ = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

motion_detected = any(cv2.contourArea(c) > 1000 for c in contours)

```

Рисунок 3.6 – Лістинг порівняння кадрів для виявлення руху

У разі виявлення автомобіля формується UDP-повідомлення та надсилається на IP-адресу центрального блоку. Для цього використовується стандартна бібліотека `socket`. На рисунку 3.7, в змінній “`LANE_ID`” необхідно вказати ID, який буде відповідати за конкретну смугу на дорозі, щоб при отриманні повідомлення центральним контролером він зрозумів на якій смузі знаходиться авто.

```

import socket

UDP_IP = "192.168.1.100" # IP центрального Raspberry Pi
UDP_PORT = 5005
LANE_ID = "lane1" # ідентифікатор смуги

message = f"{LANE_ID}_vehicle_detected".encode()
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.sendto(message, (UDP_IP, UDP_PORT))

```

Рисунок 3.7 – Лістинг надсилання повідомлення на ЦК

Уся логіка реалізується у нескінченному циклі “`while True`”, який зчитує зображення з камери, виконує обробку, порівнює з попереднім кадром та у разі потреби відправляє сигнал (Рисунок 3.8). Після цього, щоб мати з чим аналізувати наступну фотографію, необхідно поточну перезаписати як попередню. Для цього присвоюємо змінній “`prev_frame`” значення обробленого зображення “`blurred`”. У циклі також реалізується затримка для зменшення навантаження на процесор, яка записується в змінній “`FRAME_DELAY`”. Такий підхід дозволяє отримувати надійні сигнали про наявність транспорту без складної обробки.


```

while True:
    # Зчитування кадру
    frame = picam2.capture_array()
    gray = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
    blurred = cv2.GaussianBlur(gray, (21, 21), 0)

    if prev_frame is not None:
        # Різниця між кадрами
        frame_diff = cv2.absdiff(prev_frame, blurred)
        _, thresh = cv2.threshold(frame_diff, 25, 255, cv2.THRESH_BINARY)

        # Контури змін
        contours, _ = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        for contour in contours:
            if cv2.contourArea(contour) > MOTION_THRESHOLD:
                message = f"{LANE_ID}_vehicle_detected".encode()
                sock.sendto(message, (UDP_IP, UDP_PORT))
                print(f"[INFO] Motion detected, message sent: {message.decode()}")
                break # надсилаємо лише одне повідомлення за цикл

    prev_frame = blurred
    time.sleep(FRAME_DELAY)

```

Рисунок 3.8 – Лістинг нескінченного циклу програми для виявлення руху

3.3 Розробка програмного забезпечення для центрального керування

Центральна частина системи керування світлофорами реалізується на окремому Raspberry Pi, який виконує роль контролера. Основним його завданням є прийом сигналів від периферійних Raspberry Pi (які відповідають за детекцію автомобілів) та керування реле, що комутують сигнали світлофорів.

Для обміну даними між модулями застосовується дві паралельні моделі зв'язку:

- через UDP передача сигналів від камер про наявність транспорту;
- через TCP передача команд на Raspberry Pi, які керують реле.

Було обрано саме такі моделі зв'язку, тому що UDP має найвищу швидкість передачі даних, але немає гарантії що всі пакети даних будуть доставлені, але враховуючи що саме ці дані будуть постійно надсилатися від Raspberry Pi з камерою, втрати не є критичними. Передача команди на переключення сигналу світлофора є найважливішим етапом в роботі системи, не можна допускати ситуації в якій, наприклад, один світлофор переключиться а інший ні. Тому для цієї задачі було обрано протокол передачі даних TCP –

швидкість передачі даних в нього дещо менша ніж в UDP, але є гарантія доставки повідомлення і команди доставляються по черзі.

Для початку задаються мережеві порти, IP-адреси Raspberry Pi з реле та часові константи, що визначають правила перемикання. Для синхронної роботи потрібно вказати дві IP адреси – для головної і для другорядної дороги, тобто в даному випадку два контролери з камерою на головній будуть мати одну адресу, а два на другорядній – іншу (Рисунок 3.9).

```
MAIN_LIGHT_IP = "192.168.1.201"  
SECONDARY_LIGHT_IP = "192.168.1.202"  
TCP_PORT = 6000
```

Рисунок 3.9 – Присвоєння змінним IP адреси контролерів з реле

Для надсилання команд на Raspberry Pi яке перемикає реле для живлення світлофорів, було реалізовано функцію “send_command()”, яка приймає параметри з інших функцій, і в залежності які параметри надійшли встановлює TCP-з'єднання з відповідним IP та портом і відсилає команди (“green_on”, “red_off”). Лістинг коду функції продемонстровано на рисунку 3.10.

```
def send_command(ip, port, command):  
    try:  
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:  
            s.connect((ip, port))  
            s.sendall(command.encode())  
            print(f"[TCP] Sent '{command}' to {ip}:{port}")  
    except Exception as e:  
        print(f"[ERROR] Failed to send '{command}' to {ip}: {e}")
```

Рисунок 3.10 – Лістинг функції відправки TCP-команд

Далі слідують три функції які відповідають безпосередньо за те як будуть переключатися світлофори.

Першою з них є “blink_green()” – це простенька функція яка відповідає за мигання зеленого сигналу світлофора перед переключенням на жовтий. Мигання відбувається завдяки циклу for і повторюється три рази, з інтервалом включення і виключення 0.5 секунди.

Друга функція “yellow_phase()” відповідає за включення жовтого сигналу. Для початку вона вимикає всі сигнали які були включені до цього, включає жовтий на 3 секунди, після закінчення таймеру виключає його.

Третя функція – це головна функція перемикавання “switch_lights()” послідовно виконує миготіння зеленого на дорозі, яка зараз має зелений, потім жовту фазу на всіх світлофорах, і врешті переключає зелений/червоний відповідно. Для початку потрібно визначити на якій дорозі зараз зелене світло, для цього в нас є параметри функції from_lane (з якої дороги переключаємо зелене світло) to_lane (на яку дорогу переключаємо зелене світло), для цього на ці параметри передаються значення “main” і “secondary” – головна і другорядна дорога відповідно.

Далі в змінні from_ip, to_ip присвоюються IP адреси головної чи другорядної дороги, шляхом порівняння from_lane / to_lane == “main”.

Лістинг функції продемонстровано на рисунку 3.11.

```
def switch_lights(from_lane, to_lane):  
    print(f"[INFO] Перемикання з {from_lane} до {to_lane}")  
  
    from_ip = MAIN_LIGHT_IP if from_lane == "main" else SECONDARY_LIGHT_IP  
    to_ip = MAIN_LIGHT_IP if to_lane == "main" else SECONDARY_LIGHT_IP  
  
    blink_green(from_ip)  
  
    yellow_phase()  
  
    send_command(from_ip, TCP_PORT, "red_on")  
    send_command(from_ip, TCP_PORT, "green_off")  
    send_command(to_ip, TCP_PORT, "green_on")  
    send_command(to_ip, TCP_PORT, "red_off")
```

Рисунок 3.11 – Лістинг функції перемикавання сигналів

Основна функція починається з запуску UDP-сервера для прийому сигналів від модуля Raspberry Pi з камерою, після чого задаються початкові значення роботи програми при старті:

- `current_state = "main"` – означає що при старті програми світлофор включить зелене світло на головній дорозі;
- `last_switch_time = time.time()` – запис поточного часу в секундах, використовується для фіксування останньої зміни сигналів світлофора, для майбутніх обчислень таймера;
- `active_main = False / active_secondary = False` – буде присвоюватися значення True коли буде виявлено авто на головній чи на другорядній дорозі відповідно;
- `secondary_start_time = None` – в цю змінну буде записуватися поточний час, щоб на другорядній дорозі зелене світло світилося не довше ніж 15 секунд.

Таким чином, центральна логіка системи керування світлофорами реалізує адаптивне регулювання дорожнього руху з урахуванням пріоритету головної дороги, гарантованої доставки критичних команд керування та часових обмежень на тривалість фаз. Завдяки розділенню на окремі функції та структурованій логіці перемикачів забезпечується передбачувана й надійна поведінка системи в реальному часі. Такий підхід дозволяє ефективно обробляти ситуації, коли автомобілі з'являються на різних напрямках, дотримуючись як правил дорожнього руху, так і заданих логічних умов.

3.4 Розробка програмного забезпечення для керування реле

Одним із ключових компонентів комп'ютерної системи керування світлофорами є вузли, що безпосередньо керують увімкненням та вимкненням світлових сигналів. Для цього програма запускається на Raspberry Pi з камерою, паралельно з програмою для виявлення руху по смугах. Зв'язок між центральною платою і Raspberry Pi, яка керує реле, відбувається через TCP-

					КС КРБ 123.198.00.00 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

протокол. Таким чином, центральна плата формує текстові команди які передаються на відповідну IP-адресу.

Першим кроком потрібно задати статичну IP-адресу для Raspberry Pi. Для цього потрібно відкрити термінал і прописати команду `sudo nano /etc/dhcpd.conf`. Після чого в самому кінці файлу дописати такі рядки:

- в контролері на головній дорозі

```
interface eth0
static ip_address=192.168.1.201/24
static routers=192.168.1.1
static domain_name_servers=8.8.8.8 8.8.4.4
```
- в контролері на другорядній дорозі

```
interface eth0
static ip_address=192.168.1.202/24
static routers=192.168.1.1
static domain_name_servers=8.8.8.8 8.8.4.4
```

Це потрібно, щоб Raspberry Pi розуміла що команда адресується саме їй, не допускається, щоб адреса змінилася в процесі, бо це призведе до некоректної роботи системи. Програма, працює як TCP-сервер, що прослуховує визначений порт (6000) і реагує на команди, змінюючи стан відповідних GPIO-виводів. На початку програми виконується налаштування портів GPIO:

- зелений сигнал, реле 1 – GPIO (BCM) 26;
- жовтий сигнал, реле 2 – GPIO (BCM) 20;
- червоний сигнал, реле 3 – GPIO (BCM) 21.

Ініціалізація GPIO реалізується через бібліотеку `RPi.GPIO`. Усі виходи налаштовуються як вихідні, а після запуску всі сигнали вимикаються, щоб уникнути випадкового вмикання.

Фрагмент коду для ініціалізації показано на рисунку 3.12.

					КС КРБ 123.198.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

```

GPIO.setmode(GPIO.BCM)
GPIO.setup(RELAY_GREEN, GPIO.OUT)
GPIO.setup(RELAY_YELLOW, GPIO.OUT)
GPIO.setup(RELAY_RED, GPIO.OUT)

GPIO.output(RELAY_GREEN, GPIO.LOW)
GPIO.output(RELAY_YELLOW, GPIO.LOW)
GPIO.output(RELAY_RED, GPIO.LOW)

```

Рисунок 3.12 – Лістинг ініціалізації виходів GPIO

Після закінчення ініціалізації налаштовується TCP-сокет і починається вічний цикл *while True*. В ньому відбувається підключення до центрального контролера, і очікування команд. Після отримання, команд декодується і шляхом порівняння визначається на який пін GPIO потрібно подати живлення (Рисунок 3.13).

```

if command == "green_on":
    GPIO.output(RELAY_GREEN, GPIO.HIGH)
elif command == "green_off":
    GPIO.output(RELAY_GREEN, GPIO.LOW)
elif command == "yellow_on":
    GPIO.output(RELAY_YELLOW, GPIO.HIGH)
elif command == "yellow_off":
    GPIO.output(RELAY_YELLOW, GPIO.LOW)
elif command == "red_on":
    GPIO.output(RELAY_RED, GPIO.HIGH)
elif command == "red_off":
    GPIO.output(RELAY_RED, GPIO.LOW)
else:
    print(f"[WARN] Невідома команда: {command}")

```

Рисунок 3.13 – Лістинг обробки команд на переключення сигналів світлофора

Реалізація цього програмного модуля забезпечує безпосереднє керування світлофорами з боку виконавчих пристроїв. Завдяки чіткому поділу функцій та використанню TCP-протоколу гарантовано надійне приймання кожної команди від центрального вузла.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська допомога при переломах

Перелом – це порушення цілісності кісткової тканини, що виникає внаслідок надмірного механічного впливу. Існують два основні типи переломів: закриті, при яких шкіра не ушкоджена, та відкриті, коли уламки кістки проникають назовні або мають сполучення з зовнішнім середовищем через рану. За будь-якої підозри на перелом, ушкодження слід розцінювати як серйозне та надавати допомогу відповідно до стандартів.

Основними ознаками перелому є сильний біль у місці ушкодження, який посилюється під час навантаження або дотику, обмеження або повна втрата рухливості кінцівки, деформація або укорочення частини тіла, набряк, крововилив, а іноді – патологічна рухливість у місці перелому. У важких випадках, зокрема при пошкодженнях стегна, таза або черепа, загальний стан потерпілого може бути тяжким, спостерігаються ознаки шоку, підвищення температури тіла або втрата свідомості. Потерпілий може скаржитися на запаморочення, утруднене дихання або різке зниження артеріального тиску, що вказує на серйозні внутрішні порушення [13].

Долікарська допомога при переломах полягає в забезпеченні нерухомості ушкодженої ділянки, щоб уникнути зміщення уламків кістки та додаткової травматизації тканин. Основним методом є іммобілізація – створення нерухомого положення травмованої частини тіла за допомогою фіксуючих пов'язок або шин. Найефективніше використовувати стандартні шини (металеві, дерев'яні, дротяні), однак за їх відсутності можна

					КС КРБ 123.198.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Вовчак Р.А.			Безпека життєдіяльності, основи охорони праці		
Перевірів		Шингера Н.Я.					
Консульт.		Гурик О.Я.					
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					
					Літ.	Арк.	Аркушів
						55	
					ТНТУ, каф. КС, гр. СІ-41		

імпровізувати: застосовувати палиці, дошки, шматки фанери, картон, щільну тканину. Головне – щоб шина була достатньо довгою, охоплювала місце перелому та фіксувала принаймні два суміжні суглоби – вище і нижче від місця ушкодження. Імобілізація дозволяє не лише зменшити ризик ускладнень, але й знизити інтенсивність болю, що значно полегшує стан потерпілого. За можливості варто уникати переміщення потерпілого до повного знерухомлення травмованої ділянки.

Шину доцільно накладати на оголену шкіру, попередньо обгорнувши її м'яким матеріалом у місцях прилягання до кісткових виступів. Якщо знімання одягу завдає додаткового болю, шину накладають поверх одягу. У випадку відкритого перелому ділянку навколо рани слід обробити антисептиком, накласти стерильну пов'язку і лише після цього зафіксувати кінцівку шиною. Транспортувати потерпілого можна лише після накладання надійної іммобілізації. У разі сильної кровотечі необхідно попередньо накласти джгут або тиснучу пов'язку вище місця пошкодження, контролюючи час її накладення.

Особливої уваги потребують переломи різних частин тіла. Так, при переломі ключиці часто спостерігається вкорочення плеча, біль у місці травми та опущення руки. Для фіксації в пахвову ямку кладуть ватно-марлевий валик, зігнутому в лікті руку прибинтовують до тулуба, а передпліччя підвішують на косинку. Додатково рекомендується перевірити пульсацію на периферичних артеріях руки, щоб переконатися у відсутності порушення кровообігу.

При переломах ребер характерними є різкий біль при диханні, кашлі або рухах грудної клітки. У складних випадках можливе пошкодження легенів, підшкірна емфізема (під шкірою накопичується повітря) та відхаркування крові. Надання допомоги полягає у тугому бинтуванні нижньої частини грудної клітки для зменшення рухливості та больових відчуттів. Потерпілого транспортують у положенні сидячи.

					КС КРБ 123.198.00.00 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

Переломи хребта – найнебезпечніші, особливо коли пошкоджується спинний мозок. Ознаками є сильний біль вздовж хребта, обмеження або втрата рухів, а також параліч чи порушення функцій тазових органів. В такому випадку найважливішим є уникати будь-якого згинання хребта. Потерпілого обережно кладуть на живіт на тверду поверхню з валиками під плечі й голову. У разі ушкодження шийного відділу голову фіксують ватною пов'язкою у формі нашійника або підкови. Тільки після цього дозволене транспортування. Неправильне транспортування може призвести до незворотних ушкоджень спинного мозку, тому будь-які маніпуляції необхідно проводити вдвох або втрьох, синхронно.

Переломи тазу часто спричинені сильним стисненням, падінням або ударом (транспортним засобом). Потерпілий не може стояти чи ходити, спостерігаються гематоми в паховій ділянці, поза “жабки” (напів зігнуті ноги). Допомога полягає у фіксації таза широким рушником, розташування ніг в зручному положенні з валиками під колінами та терміновому доправленні до медичного закладу.

Черепно-мозкові травми, як правило, супроводжуються ушкодженням м'яких тканин голови або мозку. Наявність розладу свідомості, кровотечі з носа чи вух, а також нудота, блювання чи судоми свідчать про можливі тяжкі наслідки. Потерпілого слід терміново і дуже обережно транспортувати, фіксуючи голову та шию, у положенні лежачи. При втраті свідомості важливо перевірити дихання та пульс, при потребі – провести серцево-легеневу реанімацію.

Правильне надання першої допомоги при переломах, зокрема своєчасна іммобілізація та дотримання принципів транспортування, має вирішальне значення для збереження здоров'я та життя потерпілого. Кожна хвилина зволікання або неправильних дій може спричинити ускладнення, тому важливо діяти швидко, чітко і відповідно до стандартних методик.

					КС КРБ 123.198.00.00 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

4.2 Вплив світла на людину

Світло – це форма електромагнітного випромінювання, що сприймається людським оком. Видиме світло займає частину спектру з довжиною хвилі приблизно від 380 до 780 нм. Воно є необхідною умовою для зору та орієнтації людини у просторі. Окрім того, світло має суттєвий фізіологічний, психологічний та біологічний вплив на організм людини, особливо в контексті умов праці [14].

У виробничому середовищі правильне освітлення є однією з основ безпеки та гігієни праці. Недостатнє або, навпаки, надмірне освітлення може призводити до погіршення зору, зниження працездатності, підвищення втоми та навіть до професійних захворювань.

У системі охорони праці велике значення мають норми освітлення робочих місць. В Україні норми контролюються ДБН В.2.5-28:2018 «Природне і штучне освітлення» [15].

Основні параметри освітлення:

- яскравість (лк) – інтенсивність світлового потоку;
- співвідношення мінімального і середнього значення освітленості;
- освітлення робочої зони – залежить від типу роботи: для точних робіт вимагається 500 - 1000 лк, для менш точних – 200 - 300 лк;
- температура світла – нейтрально-біле або денне світло (4000 – 6000 К) є найбільш комфортним для очей. Правильно підібрана температура світла також позитивно впливає на емоційний стан працівника, сприяючи зниженню рівня стресу та покращенню настрою.

Також важливим є співвідношення природного та штучного освітлення, що дозволяє знизити навантаження на очі. Природне освітлення поділяється на бокове, що здійснюється через вікна в зовнішніх стінах, верхнє, здійснюване через ліхтарі та отвори в дахах і перекриттях та комбіноване – поєднання верхнього та бокового освітлення. Штучне освітлення у виробничих приміщеннях поділяється на загальне та комбіноване.

					КС КРБ 123.198.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

Загальне освітлення передбачає встановлення світильників у верхній частині приміщення (не нижче 2,5 метра від підлоги). Розміщення освітлювальних приладів може бути рівномірним по всій площі (загальне рівномірне освітлення) або орієнтованим на зони з розташованими робочими місцями (загальне локалізоване освітлення).

Комбіноване освітлення поєднує загальне та місцеве освітлення. Воно є доцільним для виконання робіт, що вимагають високої точності або коли необхідно змінювати напрямок світла під час процесу.

Місьцеве освітлення забезпечується за допомогою світильників, які спрямовують світловий потік безпосередньо на конкретну робочу зону. Проте використання виключно місцевого освітлення забороняється, оскільки це може призвести до зростання ризику травмування працівників і розвитку професійних захворювань.

Залежно від функціонального призначення штучне освітлення поділяється на такі типи:

- робоче освітлення є обов'язковим для всіх виробничих приміщень;
- адаптивне освітлення це системи, які автоматично регулюють інтенсивність світла відповідно до часу доби або наявності працівників у приміщенні. Це сприяє економії енергії та покращенню умов праці;
- аварійне освітлення використовується в разі вимкнення робочого освітлення, якщо це може спричинити порушення роботи технологічного процесу, небезпечну ситуацію для персоналу або аварію. Мінімальна освітленість має складати не менше 2 лк;
- евакуаційне освітлення призначене для забезпечення евакуації робочого персоналу з приміщень під час аварійного вимкнення світла. Обов'язкове в місцях небезпечних для проходження людей, в проходах та на сходових клітках. Освітлення повинне складати не менше 0.5 лк;
- охоронне освітлення встановлюється вздовж меж території, яка охороняється спеціальним персоналом.

Для забезпечення комфортних умов зорової роботи, зменшення втомлюваності очей, профілактики професійних хвороб та запобігання нещасним випадкам, а також з метою підвищення продуктивності та якості праці, виробниче освітлення має відповідати таким критеріям:

- освітленість робочої поверхні повинна відповідати типу зорової діяльності та не бути нижчою за встановлені нормативи;
- необхідно забезпечити стабільний рівень освітлення по всій площі приміщення, щоб уникати частих змін адаптації зору;
- не допускається поява супутніх небезпечних або шкідливих факторів, таких як надмірне тепло, шум, ризик ураження електричним струмом чи пожежа;

Світло є одним із ключових факторів, що впливають на фізіологічний і психологічний стан людини в умовах дорожнього руху. Світлофори — це важливі елементи міської інфраструктури, які використовують світлові сигнали для регулювання руху транспорту і пішоходів. Колір світлового сигналу має критичне значення: він повинен бути чітко видимим у різних погодних умовах, не викликати зорової втоми і не створювати небезпечних ілюзій для водіїв. Зелений колір, що використовується для дозволу руху, з роками змінив свій відтінок у багатьох країнах, зокрема й в Україні. Перехід до смарагдово-зеленого пов'язаний із покращенням видимості цього сигналу у денний і нічний час, а також із зниженням ризику плутанини із жовтим кольором на старих типах світлофорів.

Смарагдовий зелений відтінок забезпечує кращий контраст на тлі природного освітлення, зокрема в сутінках і в умовах яскравого сонячного світла. Це позитивно впливає на швидкість реакції водіїв та пішоходів, знижуючи ризик аварій. З позиції охорони праці, особливо для працівників, які обслуговують або встановлюють світлофорне обладнання, важливим є також дотримання норм освітленості та захист зору від тривалого впливу інтенсивного світла LED-ламп.

					КС КРБ 123.198.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

ВИСНОВКИ

У межах виконання кваліфікаційної роботи реалізовано комп'ютерну систему керування світлофорами на базі мікрокомп'ютерів Raspberry Pi. Система передбачає взаємодію окремих модулів в умовах мережевої взаємодії та забезпечує автоматичне перемикання сигналів світлофорів залежно від інтенсивності транспортного потоку на кожному напрямку.

Основу апаратної конфігурації становлять п'ять плат Raspberry Pi 4 Model B, з яких чотири виконують роль локальних сенсорних вузлів із підключеними модулями камер, а п'ята — центральний контролер, що координує зміну світлофорних фаз. Виявлення транспортних засобів на смугах руху здійснюється за допомогою аналізу зміни зображення з камери у реальному часі. Обмін даними між модулями реалізовано за допомогою UDP і TCP-протоколів через Ethernet кабель, що дозволяє забезпечити як швидку передачу сигналів про виявлення автомобілів, так і гарантоване керування станами реле світлофорів.

У межах технічного завдання було виконано такі основні дії:

- сформульовано функціональні вимоги до інтелектуальної системи керування світлофорами в умовах простого перехрестя з двох доріг;
- проведено аналіз варіантів реалізації систем розподіленого виявлення транспорту та вибір апаратної платформи;
- розроблено логіку взаємодії між вузлами системи;
- реалізовано алгоритм пріоритетного обслуговування головної дороги з урахуванням обмеження часу очікування для другорядної;
- розроблено програмне забезпечення для кожного модуля системи;
- реалізовано логіку централізованого прийняття рішень щодо зміни сигналів з урахуванням стану кожної смуги;
- проведено тестування програмної частини системи в умовах модельного середовища для перевірки правильності чергування фаз і обробки вхідних подій.

					КС КРБ 123.198.00.00 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жаровський Р.О., Луцик Н.С., Осухівська Г.М., Паламар А.М., Тиш Є.В. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль: ТНТУ, 2024. 39 с.

2. Микитишин А. Г., Митник М. М., Стухляк П. Д. Телекомунікаційні системи та мережі. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017. 384 с.

3. Оконський М.В., Лупенко С.А., Паламар А.М. Комп'ютерна система для моніторингу метеорологічних параметрів на основі IoT. 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. 112.

4. DATASHEET Raspberry Pi 4 Model B. URL <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf> (дата звернення 15.05.2025р.)

5. Ларіоник Р.В., Луцик Н.С., Паламар А.М. Система для моніторингу якості атмосферного повітря на базі IoT. 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. 116.

6. Raspberry Pi Camera Module 3. URL: <https://datasheets.raspberrypi.com/camera/camera-module-3-product-brief.pdf> (дата звернення 16.05.2025)

7. RPi Expansion Board Relay Module. URL: https://www.waveshare.com/wiki/RPi_Relay_Board (дата звернення 19.05.2025)

8. 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.

					КС КРБ 123.198.00.00 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

9. Буров Є.В., Митник М.М. Комп'ютерні мережі. Підручник. Том другий. Львів: «Магнолія 2006», 2024. 204 с.

10. Python Tutorial. URL: <https://www.w3schools.com/python/default.asp> (дата звернення 18.05.2023 р.).

11. Лупенко С.А., Пасічник В.В., Тиш Є.В. Комп'ютерна логіка. Навчальний посібник. Львів: Видавництво «Магнолія 2006», 2024. 354 с.

12. Слюз І., Жаровський Р. Критерії ефективності тестування комп'ютерної інформаційної системи. Матеріали XI Міжнародна науково-технічна конференція молодих учених та студентів «Актуальні задачі сучасних технологій» (7-8 грудня 2022 року). Тернопіль: ТНТУ. 2022. С. 174

13. ДБН В.2.5-28 : 2018. „Природне і штучне освітлення” – Київ: Мінрегіон України, 2018. 133 с.

14. Жидецький В.Ц. Основи охорони праці: підручник. Львів: Афіша. 2004. 250 с.

15. Желібо Є.П., Заверуха Н.М., Зацарний В.В. Безпека життєдіяльності: навчальний посібник. Київ: Каравела. 2008. 344 с.

					КС КРБ 123.198.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС

_____ Осухівська Г.М.

“ ____ ” _____ 2025 р

Комп'ютерна система керування світлофорами

ТЕХНІЧНЕ ЗАВДАННЯ

на 9 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

_____ к.т.н., доц. Шингера Н.Я.

« ____ » _____ 2025 р.

«ВИКОНАВЕЦЬ»

Студент групи СІ-41

_____ Вовчак Р.А.

« ____ » _____ 2025 р.

Тернопіль 2025

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп'ютерна система керування світлофорами».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.198.00.00

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, Вовчак Р.А.

1.3 Підстава для виконання роботи

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

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 27.01.2025 р.

Плановий термін завершення виконання кваліфікаційної роботи – 23.06.2025 р.

1.5 Порядок оформлення та пред'явлення результатів роботи

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

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

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

2 Призначення і цілі створення системи

2.1 Призначення системи

Проектована комп'ютерна система керування світлофорами на базі Raspberry Pi призначена для автоматичного регулювання дорожнього руху на простому перехресті. Основна мета системи – забезпечення динамічного перемикавання сигналів світлофорів залежно від наявності транспортних засобів на смугах руху. Система спрямована на зниження простоїв транспорту, підвищення пропускної здатності та безпеки дорожнього руху.

Система може бути використана:

- у якості автономного рішення для регулювання руху в умовах, де недоцільно або неможливо використовувати централізоване керування;
- для впровадження в розумні міські системи.

2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка комп'ютерної системи керування світлофорами на базі мікрокомп'ютера Raspberry Pi.

Для досягнення мети необхідно реалізувати:

- аналіз архітектури сучасних систем керування дорожнім рухом;

- проєктування модульної системи з розподіленою обробкою подій;
- обґрунтування вибору апаратної платформи;
- розробку програмного забезпечення для виявлення руху, передавання сигналів і керування світлофорами;
- розгортання мережевої взаємодії між модулями з використанням UDP і TCP;

2.3 Характеристика об'єкту

Комп'ютерна система призначена для використання на дорогах загального призначення, а саме на регульованих перехрестях.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

Комп'ютерна система повинна забезпечити:

- виявлення наявності транспорту на кожній смузі руху;
- керування сигналами світлофора відповідно до алгоритму пріоритету;
- узгоджене перемикання світлофорів на головній і другорядній дорогах;
- безпечну та надійну взаємодію між модулями системи.

3.1.1 Вимоги до структури та функціонування системи

Система керування світлофорами повинна складатись з:

- Raspberry Pi 4 Model B як обчислювальна платформа;

- чотирьох вузлів відеофіксації з камерами та обробкою зображення;
- центральний вузол для прийняття рішень і передавання команд;
- використання реле-модуля для комутації світлофорів;
- підтримка протоколів TCP і UDP для передачі даних.

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

- статичні IP-адреси для всіх вузлів;
- UDP-протокол між вузлами з відеофіксацією і центральним вузлом;
- TCP-протокол для надсилання команд на вузли з реле;
- мінімальні затримки передачі, гарантія доставки критичних команд.

3.1.3 Вимоги по діагностуванню системи

До вимог по діагностуванню відноситься:

- логування отриманих повідомлень;
- логування стану світлофорів і часу перемикань;
- ручний та автоматичний моніторинг підключення вузлів.

3.1.4 Перспективи розвитку, проектування системи

- масштабування до складніших перехресть;
- використання алгоритмів машинного навчання для прогнозування потоків;
- централізоване управління декількома перехрестями через сервер;
- впровадження веб-інтерфейсу моніторингу.
-

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

- стійкість до короточасного зникнення зв'язку;
- забезпечення коректної роботи при перезапуску вузлів;
- уникнення конфліктів сигналів на різних дорогах;
- повна синхронізація фаз перемикання.

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 .

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 апаратного забезпечення

- Raspberry Pi 4 Model B (мін. 2 ГБ RAM) – 5 шт.;
- Raspberry Pi Camera Module 3 – 4 шт.;
- Ethernet-комутатор для з'єднання вузлів;
- Релейні модулі з підтримкою 3 каналів – 2 шт.;
- Блоки живлення 5 В / 3 А – 5 шт.;
- Кабелі та монтажна плата.

3.3.2 Вимоги до функцій

- фіксація руху на смузі;
- передавання події на центральний вузол;
- перемикання сигналів світлофора відповідно до алгоритму;
- облік часу останнього перемикання;
- підтримка таймерів для обмеження фази другорядної дороги.

3.3.3 Вимоги до програмного забезпечення:

- ОС Raspberry Pi OS (Bookworm);
- Python 3.9+ з бібліотеками: OpenCV (cv2), socket, threading, time;
- модулі для керування GPIO: RPi.GPIO;
- підтримка автоматичного запуску при включенні.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:
 - а) Структурна схема.
 - б) Схема електрична принципова.
 - б) Блок схема.
 - в) Результат роботи.

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

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання.	27.01.2025р. – 02.02.2025р.
2	Робота над основними розділами – Аналіз технічного завдання	28.02.2025р. – 15.03.2025р.
3	Робота над основними розділами – Проектна частина	16.03.2025р. – 07.04.2025р.
4	Робота над основними розділами – Програмна частина	08.04.2025р. – 10.05.2025р.
5	Безпека життєдіяльності, основи охорони праці	11.05.2025р. – 20.05.2025р.
6	Оформлення пояснювальної записки і графічного матеріалу	25.05.2025р. – 9.06.2025р.
7	Перевірка на академічний плагіат, перевірка керівником та консультантами	9.06.2025р. – 15.06.2025р.
8	Попередній захист кваліфікаційної роботи бакалавра	16.06.2025р. – 22.06.2025р.
9	Захист кваліфікаційної роботи бакалавра	25.06.2025р.

6 Додаткові умови виконання кваліфікаційної роботи

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

Додаток Б

Перелік елементів

Додаток В
Лістинг коду

Лістинг 1 – програма для виявлення руху на смузі

```
from picamera2 import Picamera2
import cv2
import time
import socket
import numpy as np

# ----- Налаштування -----
UDP_IP = "192.168.1.100"          # IP-адреса центрального
Raspberry Pi
UDP_PORT = 5005                  # Порт для надсилання UDP-
повідомлення
LANE_ID = "lane1"                # Ідентифікатор смуги (змінити
для кожної камери)
MOTION_THRESHOLD = 1000          # Мінімальна площа змін для
фіксації руху
FRAME_DELAY = 0.5                # Затримка між кадрами, с

# ----- Ініціалізація камери -----
picam2 = Picamera2()
picam2.preview_configuration.main.size = (640, 480)
picam2.preview_configuration.main.format = "RGB888"
picam2.configure("preview")
picam2.start()
time.sleep(2)  # прогрів камери

# ----- Ініціалізація сокету -----
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# ----- Основний цикл -----
prev_frame = None

while True:
    # Зчитування кадру
    frame = picam2.capture_array()
```

```

gray = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
blurred = cv2.GaussianBlur(gray, (21, 21), 0)

if prev_frame is not None:
    # Пізниця між кадрами
    frame_diff = cv2.absdiff(prev_frame, blurred)
    _, thresh = cv2.threshold(frame_diff, 25, 255,
cv2.THRESH_BINARY)

    # Контури змін
    contours, _ = cv2.findContours(thresh.copy(),
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    for contour in contours:
        if cv2.contourArea(contour) > MOTION_THRESHOLD:
            message = f"{LANE_ID}_vehicle_detected".encode()
            sock.sendto(message, (UDP_IP, UDP_PORT))
            print(f"[INFO] Motion detected, message sent:
{message.decode()}")
            break # надсилаємо лише одне повідомлення за
цикл

prev_frame = blurred
time.sleep(FRAME_DELAY)

```

Лістинг 2 – Центральне керування світлофорами

```
import socket
import time

# ----- Налаштування -----
UDP_PORT = 5005

MAIN_LIGHT_IP = "192.168.1.201"
SECONDARY_LIGHT_IP = "192.168.1.202"
TCP_PORT = 6000

main_priority_hold_time = 60      # Утримання зеленої головної при
о́бох авто

secondary_green_time = 15         # Макс. час зеленої другорядної
green_blink_interval = 0.5       # Інтервал миготіння зеленого
green_blink_count = 3            # Кількість миготінь

yellow_duration = 3              # Тривалість жовтого сигналу

# ----- Функції TCP-команд -----
def send_command(ip, port, command):
    try:
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as
s:
            s.connect((ip, port))
            s.sendall(command.encode())
            print(f"[TCP] Sent '{command}' to {ip}:{port}")
    except Exception as e:
        print(f"[ERROR] Failed to send '{command}' to {ip}: {e}")

def blink_green(ip):
    for _ in range(green_blink_count):
        send_command(ip, TCP_PORT, "green_off")
        time.sleep(green_blink_interval)
        send_command(ip, TCP_PORT, "green_on")
        time.sleep(green_blink_interval)
    send_command(ip, TCP_PORT, "green_off")
```

```

def yellow_phase():
    # Вимикаємо зелений і червоний на обох світлофорах перед
    # ЖОВТИМ
    send_command(MAIN_LIGHT_IP, TCP_PORT, "green_off")
    send_command(MAIN_LIGHT_IP, TCP_PORT, "red_off")
    send_command(SECONDARY_LIGHT_IP, TCP_PORT, "green_off")
    send_command(SECONDARY_LIGHT_IP, TCP_PORT, "red_off")

    # Включаємо жовтий на обох світлофорах
    send_command(MAIN_LIGHT_IP, TCP_PORT, "yellow_on")
    send_command(SECONDARY_LIGHT_IP, TCP_PORT, "yellow_on")
    print("[INFO] Жовтий на всіх світлофорах")

    time.sleep(yellow_duration)

    # Вимикаємо жовтий
    send_command(MAIN_LIGHT_IP, TCP_PORT, "yellow_off")
    send_command(SECONDARY_LIGHT_IP, TCP_PORT, "yellow_off")

def switch_lights(from_lane, to_lane):
    print(f"[INFO] Перемикання з {from_lane} до {to_lane}")

    from_ip = MAIN_LIGHT_IP if from_lane == "main" else
SECONDARY_LIGHT_IP
    to_ip = MAIN_LIGHT_IP if to_lane == "main" else
SECONDARY_LIGHT_IP

    # 1. Миготіння зеленого на напрямку from_lane
    blink_green(from_ip)

    # 2. Жовтий на обох світлофорах
    yellow_phase()

    # 3. Червоне на from_ip, зелене на to_ip
    send_command(from_ip, TCP_PORT, "red_on")
    send_command(from_ip, TCP_PORT, "green_off")

    send_command(to_ip, TCP_PORT, "green_on")
    send_command(to_ip, TCP_PORT, "red_off")

```

```

# ----- ОСНОВНИЙ ЦИКЛ -----
def run_server():
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    sock.bind(("0.0.0.0", UDP_PORT))
    sock.settimeout(1.0)
    print(f"[UDP] Сервер слухає порт {UDP_PORT}")

    current_state = "main"
    last_switch_time = time.time()
    active_main = False
    active_secondary = False
    secondary_start_time = None

    while True:
        try:
            try:
                data, addr = sock.recvfrom(1024)
                message = data.decode().strip()
                print(f"[UDP] Отримано від {addr}: {message}")

                if "lane" in message:
                    if message in ["lane1", "lane2"]:
                        active_main = True
                    elif message in ["lane3", "lane4"]:
                        active_secondary = True

            except socket.timeout:
                pass

            now = time.time()

            if current_state == "main":
                # Якщо авто є на другорядній, а на головній немає
                - негайне переключення
                if active_secondary and not active_main:
                    if now - last_switch_time >= 3: # затримка
                        для миготіння

```



```

switch_lights("main", "secondary")
current_state = "secondary"
last_switch_time = now
secondary_start_time = now
active_secondary = False
active_main = False

# Якщо авто на обох - утримуємо головну до 60 с,
потім переключаємо
elif active_secondary and active_main:
    if now - last_switch_time >=
main_priority_hold_time:
        switch_lights("main", "secondary")
        current_state = "secondary"
        last_switch_time = now
        secondary_start_time = now
        active_secondary = False
        active_main = False

# Якщо на головній вже немає авто -
переключаємо раніше
elif not active_main:
    switch_lights("main", "secondary")
    current_state = "secondary"
    last_switch_time = now
    secondary_start_time = now
    active_secondary = False
    active_main = False

# Якщо авто тільки на головній або немає авто -
нічого не робимо

elif current_state == "secondary":
    # Повернення зразу якщо на другорядній машин
    немає а на головній є
    if active_main and not active_secondary:
        switch_lights("secondary", "main")
        current_state = "main"

```

```

        last_switch_time = now
        secondary_start_time = None
        active_main = False
        active_secondary = False

        # Повернення на головну після 15 с зеленого
другого

        elif secondary_start_time and (now -
secondary_start_time >= secondary_green_time):
            switch_lights("secondary", "main")
            current_state = "main"
            last_switch_time = now
            secondary_start_time = None
            active_secondary = False
            active_main = False

    except Exception as e:
        print(f"[ERROR] {e}")

# ----- Запуск -----
if __name__ == "__main__":
    run_server()

```

Лістинг 3 – програма перемикання реле світлофорів.

```
import socket

import RPi.GPIO as GPIO

# Налаштування GPIO
RELAY_GREEN = 26    # BCM 26 = Pin 37 = Реле 1
RELAY_YELLOW = 20   # BCM 20 = Pin 38 = Реле 2
RELAY_RED = 21      # BCM 21 = Pin 40 = Реле 3

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

# Ініціалізація пінів
GPIO.setup(RELAY_GREEN, GPIO.OUT)
GPIO.setup(RELAY_YELLOW, GPIO.OUT)
GPIO.setup(RELAY_RED, GPIO.OUT)

# Вимкнути всі реле при старті
GPIO.output(RELAY_GREEN, GPIO.LOW)
GPIO.output(RELAY_YELLOW, GPIO.LOW)
GPIO.output(RELAY_RED, GPIO.LOW)

# Налаштування TCP-сокету
HOST = "0.0.0.0"
PORT = 6000

print(f"[TCP] Сервер запущено на порту {PORT}")

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen(5)

    while True:
        conn, addr = s.accept()
        with conn:
            print(f"[TCP] Підключення від {addr}")
            data = conn.recv(1024)
```

```

if not data:
    continue
command = data.decode().strip()
print(f"[TCP] Команда: {command}")

# Обробка команд
if command == "green_on":
    GPIO.output(RELAY_GREEN, GPIO.HIGH)
elif command == "green_off":
    GPIO.output(RELAY_GREEN, GPIO.LOW)
elif command == "yellow_on":
    GPIO.output(RELAY_YELLOW, GPIO.HIGH)
elif command == "yellow_off":
    GPIO.output(RELAY_YELLOW, GPIO.LOW)
elif command == "red_on":
    GPIO.output(RELAY_RED, GPIO.HIGH)
elif command == "red_off":
    GPIO.output(RELAY_RED, GPIO.LOW)
else:
    print(f"[WARN] Невідома команда: {command}")

```